Tarea UD10: Comunicándonos con el usuario. Interfaces.

Programación

Curso DAM 2024/25

Calificación obtenida: 8’50/10

Cosas a mejorar: *La funcionalidad con más operandos trataba sobre resolver operaciones como 3+8*4 = 35. Tu calculadora siempre trabaja con dos operandos porque resuelve cada vez que se inserta un dato.*

Índice

Detalles de la tarea..... 3
Anexo.....4
Código controlador..... 4
Código Principal..... 9
Código Interfaz..... 10

Detalles de la tarea

Con los conocimientos adquiridos durante la unidad se propone la implementación de una calculadora, parecida a la que se implementa en uno de los vídeos referenciados en los contenidos, pero utilizando la librería JavaFX. El nombre de la aplicación será **PROG10_Tarea**. Ten en cuenta que:

- La calculadora podrá realizar sólo operaciones sumar, restar, multiplicación y división de números enteros.
- Será obligatorio como mínimo realizar operaciones con dos operandos, valorándose positivamente la posibilidad de realizar operaciones con mas operandos.

¡IMPORTANTE! Debido a la diversidad de versiones de librerías FX instaladas por los alumnos, en esta tarea será imprescindible la entrega de un vídeo explicativo con toda la funcionalidad desarrollada. El vídeo tendrá una duración aproximada de 3 minutos. Debéis tener cuidado con el tamaño del archivo y grabarlo con baja definición para no tener problemas al subirlo.

El alumno puede añadir cualquier funcionalidad adicional que considere oportuna, siempre y cuando suponga añadir funciones a las mínimas propuestas.

Anexo

Código controlador

```
package minicalculadorafx;

import javafx.scene.control.TextField;
import java.net.URL;
import java.text.DecimalFormat;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.Initializable;
import javafx.fxml.FXML;
import javafx.scene.control.Button;

/**
 * MiniCalculadoraInterfazController es la clase controladora para la interfaz
 * de la mini calculadora. Gestiona la lógica de la calculadora, incluyendo el
 * manejo de números, operaciones, resultados y la limitación de la longitud de
 * entrada y resultado.
 */
public class MiniCalculadoraInterfazController implements Initializable {
    // Atributo para mostrar los números y resultados en la interfaz.

    @FXML
    private TextField textField;

    // Variables para almacenar los operandos y la operación actual.
    /**
     * Primer operando o resultado acumulado.
     */
    private double numero1 = 0;
    /**
     * Segundo operando.
     */
    private double numero2 = 0;
    /**
     * Operación pendiente ("+", "-", "*", "/").
     */
```

```
    private String operacion = "";
    /**
     * Indica si se está esperando que el usuario ingrese un nuevo número.
     */
    private boolean nuevaEntrada = true;

    // Constante para la longitud máxima permitida en la entrada y el
resultado.
    private final int MAX_LENGTH = 14;

    /**
     * Maneja la entrada de números. Actualiza el contenido del TextField con
el
     * dígito presionado. Si se está iniciando una nueva entrada o el TextField
     * muestra "0", reemplaza el contenido; de lo contrario, concatena el
dígito
     * siempre que la longitud actual sea menor a MAX_LENGTH.
     *
     * @param event Evento de acción que se produce al presionar un botón
     * numérico.
     */
    @FXML
    private void handleNumero(ActionEvent event) {
        Button boton = (Button) event.getSource();
        String valor = boton.getText();

        if (nuevaEntrada || textField.getText().equals("0")) {
            textField.setText(valor);
            nuevaEntrada = false;
        } else {
            if (textField.getText().length() < MAX_LENGTH) {
                textField.setText(textField.getText() + valor);
            }
        }
    }

    /**
     * Borra el contenido del TextField y reinicia las variables de la
     * calculadora. Se invoca al presionar el botón de borrar, estableciendo el
     * TextField en "0" y reiniciando los operandos, la operación pendiente y
el
```

```
     * estado que indica si se espera una nueva entrada numérica.
     *
     * @param event Evento de acción del botón borrar.
     */
    @FXML
    private void handleBorrar(ActionEvent event) {
        textField.setText("0");
        numero1 = 0;
        numero2 = 0;
        operacion = "";
        nuevaEntrada = true;
    }

    /**
     * Maneja la inserción del punto decimal. Añade un punto decimal al
     * contenido del TextField, siempre que éste no lo contenga ya y la
longitud
     * actual no exceda MAX_LENGTH.
     *
     * @param event Evento de acción que se produce al presionar el botón de
     * coma.
     */
    @FXML
    private void handleComa(ActionEvent event) {
        String currentText = textField.getText();
        if (!currentText.contains(".") && currentText.length() < MAX_LENGTH) {
            textField.setText(currentText + ".");
        }
    }

    /**
     * Realiza la operación pendiente y muestra el resultado en el TextField.
     * Toma el segundo operando del TextField, realiza la operación (suma,
     * resta, multiplicación o división) y formatea el resultado a dos
     * decimales. Si la longitud del resultado supera MAX_LENGTH, se redondea
     * para ajustarlo al límite.
     *
     * @param event Evento de acción que se produce al presionar el botón de
     * resultado.
     */
    @FXML
```

```
    private void handleResultado(ActionEvent event) {
        if (!textField.getText().isEmpty() && !operacion.isEmpty()) {
            try {
                numero2 = Double.parseDouble(textField.getText());

                switch (operacion) {
                    case "+":
                        numero1 += numero2;
                        break;
                    case "-":
                        numero1 -= numero2;
                        break;
                    case "*":
                        numero1 *= numero2;
                        break;
                    case "/":
                        if (numero2 != 0) {
                            numero1 /= numero2;
                        } else {
                            textField.setText("Error");
                            return;
                        }
                        break;
                }

                // Redondear el resultado a 2 decimales
                DecimalFormat df = new DecimalFormat("#.##");
                String resultado = df.format(numero1);

                // Si el resultado supera el límite de longitud, redondearlo
para ajustarlo.
                if (resultado.length() > MAX_LENGTH) {
                    resultado = String.format("%." + (MAX_LENGTH - 1) + "f",
numero1);
                }
                textField.setText(resultado);
                operacion = "";
                nuevaEntrada = true;
            } catch (NumberFormatException e) {
                textField.setText("Error");
            }
```

```
        }
    }

    /**
     * Maneja la selección de una operación. Guarda el número ingresado (primer
     * operando) y establece la operación a realizar, preparando la calculadora
     * para la siguiente entrada numérica. Si se presionan dos botones de
     * operación seguidos (sin ingresar un número nuevo), simplemente actualiza
     * el operador sin calcular un resultado.
     *
     * @param event Evento de acción que se produce al presionar un botón de
     * operación.
     */
    @FXML
    private void handleOperacion(ActionEvent event) {
        Button boton = (Button) event.getSource();
        String nuevaOperacion = boton.getText();

        if (!textField.getText().isEmpty()) {
            try {
                // Si ya hay una operación pendiente, se realiza la operación
actual
                if (!operacion.isEmpty()) {
                    handleResultado(new ActionEvent());
                }

                // Si se acaba de ingresar un número, se guarda en numero1
                if (!nuevaEntrada) {
                    numero1 = Double.parseDouble(textField.getText());
                }
                // Establece la nueva operación y prepara la entrada del
siguiente número
                operacion = nuevaOperacion;
                nuevaEntrada = true;

            } catch (NumberFormatException e) {
                textField.setText("Error");
            }
        }
    }
```

```
    /**
     * Inicializa la interfaz de la calculadora. Se invoca al cargar la
interfaz
     * y establece el valor inicial del TextField en "0".
     *
     * @param url URL de la ubicación del recurso.
     * @param rb Objeto ResourceBundle que contiene la localización.
     */
    @Override
    public void initialize(URL url, ResourceBundle rb) {
        textField.setText("0");
    }
}
```

Código Principal

```
package minicalculadorafx;

import java.io.IOException;
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

/**
 * Clase principal que lanza la aplicación de la mini calculadora en JavaFX.
 */
public class Principal extends Application {

    @Override
    public void start(Stage primaryStage) throws IOException {

        // Cargar el FXML
        Parent root=
FXMLLoader.load(getClass().getResource("MiniCalculadoraInterfaz.fxml"));

        // Crear la escena
        Scene scene= new Scene(root);
```

```
        // Establecer el título de la ventana
        primaryStage.setTitle("Mini Calculadora JavaFX");

        // Configurar las opciones de la ventana (minimizar y cerrar)
        primaryStage.setResizable(false); // No permitir maximizar


        // Asignar la escena al Stage
        primaryStage.setScene(scene);

        // Mostrar la ventana
        primaryStage.show();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        launch(args);
    }

}
```

Código Interfaz

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<AnchorPane id="AnchorPane" prefHeight="500.0" prefWidth="400.0"
style="-fx-background-color: black;" xmlns="http://javafx.com/javafx/23.0.1"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="minicalculadorafx.MiniCalculadoraInterfazController">
```

```
   <children>
      <VBox alignment="CENTER" layoutX="4.0" layoutY="6.0" prefHeight="508.0"
prefWidth="392.0">
         <children>
            <TextField fx:id="textField" alignment="CENTER_RIGHT"
prefHeight="77.0" prefWidth="392.0">
               <font>
                  <Font name="Arial" size="36.0" />
               </font>
               <VBox.margin>
                  <Insets left="10.0" right="10.0" top="10.0" />
               </VBox.margin>
            </TextField>
            <GridPane alignment="TOP_CENTER" hgap="5.0" prefHeight="425.0"
prefWidth="392.0" style="-fx-background-color: black;" vgap="5.0">
              <columnConstraints>
                <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0"
prefWidth="100.0" />
                <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0"
prefWidth="100.0" />
                  <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0"
prefWidth="100.0" />
                  <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0"
prefWidth="100.0" />
              </columnConstraints>
              <rowConstraints>
                  <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                  <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
              </rowConstraints>
               <children>
                  <Button fx:id="btn1" alignment="CENTER" layoutX="10.0"
layoutY="10.0" mnemonicParsing="false" onAction="#handleNumero"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: white;"
```

```
text="1" GridPane.rowIndex="1">
                     <font>
                        <Font name="Arial Bold" size="36.0" />
                     </font>
                  </Button>
                  <Button fx:id="btnC" alignment="CENTER" layoutX="10.0"
layoutY="97.0" mnemonicParsing="false" onAction="#handleBorrar"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: red;"
text="C" GridPane.columnIndex="3">
                     <font>
                        <Font name="Arial Bold" size="36.0" />
                     </font>
                  </Button>
                  <Button fx:id="btn2" alignment="CENTER" layoutX="10.0"
layoutY="97.0" mnemonicParsing="false" onAction="#handleNumero"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: white;"
text="2" GridPane.columnIndex="1" GridPane.rowIndex="1">
                     <font>
                        <Font name="Arial Bold" size="36.0" />
                     </font>
                  </Button>
                  <Button fx:id="btn3" alignment="CENTER" layoutX="106.0"
layoutY="97.0" mnemonicParsing="false" onAction="#handleNumero"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: white;"
text="3" GridPane.columnIndex="2" GridPane.rowIndex="1">
                     <font>
                        <Font name="Arial Bold" size="36.0" />
                     </font>
                  </Button>
                  <Button fx:id="btn4" alignment="CENTER" layoutX="202.0"
layoutY="97.0" mnemonicParsing="false" onAction="#handleNumero"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: white;"
text="4" GridPane.rowIndex="2">
                     <font>
                        <Font name="Arial Bold" size="36.0" />
                     </font>
                  </Button>
                  <Button fx:id="btn5" alignment="CENTER" layoutX="106.0"
layoutY="97.0" mnemonicParsing="false" onAction="#handleNumero"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: white;"
text="5" GridPane.columnIndex="1" GridPane.rowIndex="2">
```

```
                    <font>
                        <Font name="Arial Bold" size="36.0" />
                    </font>
                </Button>
                <Button fx:id="btn6" alignment="CENTER" layoutX="10.0"
layoutY="97.0" mnemonicParsing="false" onAction="#handleNumero"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: white;"
text="6" GridPane.columnIndex="2" GridPane.rowIndex="2">
                    <font>
                        <Font name="Arial Bold" size="36.0" />
                    </font>
                </Button>
                <Button fx:id="btn7" alignment="CENTER" layoutX="10.0"
layoutY="184.0" mnemonicParsing="false" onAction="#handleNumero"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: white;"
text="7" GridPane.rowIndex="3">
                    <font>
                        <Font name="Arial Bold" size="36.0" />
                    </font>
                </Button>
                <Button fx:id="btn8" alignment="CENTER" layoutX="106.0"
layoutY="184.0" mnemonicParsing="false" onAction="#handleNumero"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: white;"
text="8" GridPane.columnIndex="1" GridPane.rowIndex="3">
                    <font>
                        <Font name="Arial Bold" size="36.0" />
                    </font>
                </Button>
                <Button fx:id="btn9" alignment="CENTER" layoutX="106.0"
layoutY="271.0" mnemonicParsing="false" onAction="#handleNumero"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: white;"
text="9" GridPane.columnIndex="2" GridPane.rowIndex="3">
                    <font>
                        <Font name="Arial Bold" size="36.0" />
                    </font>
                </Button>
                <Button fx:id="btn0" alignment="CENTER" layoutX="10.0"
layoutY="271.0" mnemonicParsing="false" onAction="#handleNumero"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: white;"
text="0" GridPane.rowIndex="4">
                    <font>
```

```
                        <Font name="Arial Bold" size="36.0" />
                     </font>
                  </Button>
                  <Button fx:id="btnComa" alignment="CENTER" layoutX="106.0"
layoutY="271.0" mnemonicParsing="false" onAction="#handleComa"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: white;"
text="," GridPane.columnIndex="1" GridPane.rowIndex="4">
                     <font>
                        <Font name="Arial Bold" size="36.0" />
                     </font>
                  </Button>
                  <Button fx:id="btnIgual" alignment="CENTER" layoutX="202.0"
layoutY="271.0" mnemonicParsing="false" onAction="#handleResultado"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: white;"
text="=" GridPane.columnIndex="2" GridPane.rowIndex="4">
                     <font>
                        <Font name="Arial Bold" size="36.0" />
                     </font>
                  </Button>
                  <Button fx:id="btnSumar" alignment="CENTER" layoutX="202.0"
layoutY="97.0" mnemonicParsing="false" onAction="#handleOperacion"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: orange;"
text="+" textFill="WHITE" GridPane.columnIndex="3" GridPane.rowIndex="1">
                     <font>
                        <Font name="Arial Bold" size="36.0" />
                     </font>
                  </Button>
                  <Button fx:id="btnRestar" alignment="CENTER" layoutX="298.0"
layoutY="97.0" mnemonicParsing="false" onAction="#handleOperacion"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: orange;"
text="-" textFill="WHITE" GridPane.columnIndex="3" GridPane.rowIndex="2">
                     <font>
                        <Font name="Arial Bold" size="36.0" />
                     </font>
                  </Button>
                  <Button fx:id="btnMultiplicar" alignment="CENTER"
layoutX="298.0" layoutY="184.0" mnemonicParsing="false"
onAction="#handleOperacion" prefHeight="100.0" prefWidth="100.0"
style="-fx-background-color: orange;" text="*" textFill="WHITE"
GridPane.columnIndex="3" GridPane.rowIndex="3">
                     <font>
```

```
                        <Font name="Arial Bold" size="36.0" />
                     </font>
                  </Button>
                  <Button fx:id="btnDividir" alignment="CENTER" layoutX="298.0"
layoutY="97.0" mnemonicParsing="false" onAction="#handleOperacion"
prefHeight="100.0" prefWidth="100.0" style="-fx-background-color: orange;"
text="/" textFill="WHITE" GridPane.columnIndex="3" GridPane.rowIndex="4">
                     <font>
                        <Font name="Arial Bold" size="36.0" />
                     </font>
                  </Button>
               </children>
               <VBox.margin>
                  <Insets bottom="10.0" left="10.0" right="10.0" top="10.0" />
               </VBox.margin>
            </GridPane>
         </children>
      </VBox>
   </children>
</AnchorPane>
```