



Actividad 3 de Programación Orientada a Objetos: Aplicaciones con interfaz gráfica usando Swing)

Natalia Patiño Pérez, npatinop@unal.edu.co, CC. 1002633746

Docente: Walter Hugo Arboleda Mazo

Octubre 1, 2022

Enlace de acceso a GitHub: <https://github.com/npatinop/Actividad3POO.git>

CAPÍTULO 3: Estructura secuencial

1. Ejercicio propuesto N°18:

Método principal:

```
package EjercicioPropuesto18;

//método principal
public class EjercicioP18 {
    public static void main(String[] args) {
        InterfazP18 fr= new InterfazP18();
        fr.setVisible(true);
    }
}
```

Interfaz:

```
package EjercicioPropuesto18;

import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;
```

```

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.SwingConstants;
import java.awt.Color;
import java.awt.Font;
import javax.swing.JTextPane;

public class InterfazP18 extends JFrame {

    private JPanel contentPane;
    private JTextField textCodigo;
    private JTextField textNombre;
    private JTextField textHoras;
    private JTextField textValorHora;
    private JTextField textPorcentajeRetencion;
    private JTextField textSalarioBruto;
    private JTextField textSalarioNeto;

    /**
     * Create the frame.
     */
    public InterfazP18() {
        //siempre que sale Bounds son las medidas
        this.setTitle("Ejercicio Propuesto 18"); //poner título a la interfaz
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 725, 394);
        contentPane = new JPanel();
        contentPane.setBackground(new Color(204, 227, 179)); //color del fondo de la
interfaz
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        setContentPane(contentPane);
        contentPane.setLayout(null);

        //donde se indica setFont es el tipo de letra y el tamaño
        JLabel lblNewLabel = new JLabel("Código"); //indicador
        lblNewLabel.setFont(new Font("Rockwell", Font.PLAIN, 13));
        lblNewLabel.setBounds(35, 101, 89, 14);
        contentPane.add(lblNewLabel);

        JLabel lblNewLabel_1 = new JLabel("Nombre"); //indicador
        lblNewLabel_1.setFont(new Font("Rockwell", Font.PLAIN, 13));
        lblNewLabel_1.setBounds(35, 138, 89, 14);
        contentPane.add(lblNewLabel_1);

```

```
JLabel lblNewLabel_2 = new JLabel("Número de horas trabajadas"); //indicador
lblNewLabel_2.setFont(new Font("Rockwell", Font.PLAIN, 13));
lblNewLabel_2.setBounds(35, 180, 179, 14);
contentPane.add(lblNewLabel_2);
```

```
JLabel lblNewLabel_3 = new JLabel("Valor por hora trabajada ($)"); //indicador
lblNewLabel_3.setFont(new Font("Rockwell", Font.PLAIN, 13));
lblNewLabel_3.setBounds(35, 214, 179, 14);
contentPane.add(lblNewLabel_3);
```

```
JLabel lblNewLabel_4 = new JLabel("Retencion(%)"); //indicador
lblNewLabel_4.setFont(new Font("Rockwell", Font.PLAIN, 13));
lblNewLabel_4.setBounds(35, 256, 89, 14);
contentPane.add(lblNewLabel_4);
```

```
JLabel lblNewLabel_5 = new JLabel("Salario bruto ($)"); //indicador
lblNewLabel_5.setFont(new Font("Rockwell", Font.PLAIN, 13));
lblNewLabel_5.setBounds(366, 138, 114, 14);
contentPane.add(lblNewLabel_5);
```

```
JLabel lblNewLabel_6 = new JLabel("Salario neto ($)"); //indicador
lblNewLabel_6.setFont(new Font("Rockwell", Font.PLAIN, 13));
lblNewLabel_6.setBounds(366, 214, 114, 14);
contentPane.add(lblNewLabel_6);
```

```
textCodigo = new JTextField(); //ingresar por teclado
textCodigo.setFont(new Font("Rockwell", Font.PLAIN, 12));
textCodigo.setBounds(134, 98, 211, 20);
contentPane.add(textCodigo);
textCodigo.setColumns(10);
```

```
textNombre = new JTextField(); //ingresar por teclado
textNombre.setFont(new Font("Rockwell", Font.PLAIN, 12));
textNombre.setColumns(10);
textNombre.setBounds(134, 135, 211, 20);
contentPane.add(textNombre);
```

```
textHoras = new JTextField(); //ingresar por teclado
textHoras.setFont(new Font("Rockwell", Font.PLAIN, 12));
textHoras.setColumns(10);
textHoras.setBounds(242, 177, 103, 20);
contentPane.add(textHoras);
```

```

textValorHora = new JTextField(); //ingresar por teclado
textValorHora.setFont(new Font("Rockwell", Font.PLAIN, 12));
textValorHora.setColumns(10);
textValorHora.setBounds(242, 211, 103, 20);
contentPane.add(textValorHora);

textPorcentajeRetencion = new JTextField(); //ingresar por teclado
textPorcentajeRetencion.setFont(new Font("Rockwell", Font.PLAIN, 12));
textPorcentajeRetencion.setColumns(10);
textPorcentajeRetencion.setBounds(134, 253, 211, 20);
contentPane.add(textPorcentajeRetencion);

textSalarioBruto = new JTextField(); //calculado
textSalarioBruto.setFont(new Font("Rockwell", Font.PLAIN, 12));
textSalarioBruto.setColumns(10);
textSalarioBruto.setBounds(490, 136, 186, 20);
contentPane.add(textSalarioBruto);

textSalarioNeto = new JTextField(); //calculado
textSalarioNeto.setFont(new Font("Rockwell", Font.PLAIN, 12));
textSalarioNeto.setColumns(10);
textSalarioNeto.setBounds(490, 212, 186, 20);
contentPane.add(textSalarioNeto);

//definición del botón calcular que permite efectuar las operaciones
JButton btnCalcular = new JButton("Calcular");
btnCalcular.setBackground(new Color(165, 205, 120));
btnCalcular.setFont(new Font("Rockwell", Font.BOLD | Font.ITALIC, 12));
btnCalcular.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //para determinar el salario bruto y salario neto
        try {
            double horas, valorHora,
porcentajeRetenido, salarioRetenido, salarioBruto; //tipo
            //tomar entradas y cálculos hechos con los métodos
            horas = Double.parseDouble(textHoras.getText());
            valorHora = Double.parseDouble(textValorHora.getText());
            porcentajeRetenido =
Double.parseDouble(textPorcentajeRetencion.getText());

            textSalarioBruto.setText(String.valueOf(MetodosP18.metodoSalarioBruto(horas,
valorHora)));

            salarioBruto = MetodosP18.metodoSalarioBruto(horas, valorHora);

```

```
        salarioRetenido = MetodosP18.metodoSalarioRetenido(horas,
valorHora,porcentajeRetenido);
```

```
        textSalarioNeto.setText(String.valueOf(MetodosP18.metodoSalarioNeto(salarioRetenido,salarioBruto)));
```

```
    }
```

```
    //para excepciones
```

```
    catch(NumberFormatException ex) {
```

```
        //en caso de que se ingresen valores que no sean apropiados,
```

```
no se puede hacer el programa
```

```
        JOptionPane.showMessageDialog(null, "Los parámetros
ingresados no concuerdan con lo solicitado, intente de nuevo.");
```

```
    }
```

```
}
```

```
});
```

```
btnCalcular.setBounds(182, 304, 89, 23); //botón para presionar y determinar el
salario bruto y salario neto
```

```
contentPane.add(btnCalcular);
```

```

JButton btnBorrar = new JButton("Borrar"); //botón para eliminar el contenido de los
campos
```

```
btnBorrar.setBackground(new Color(165, 205, 120));
```

```
btnBorrar.setFont(new Font("Rockwell", Font.BOLD | Font.ITALIC, 12));
```

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

```
    public void actionPerformed(ActionEvent e) {
```

```
        //borra el contenido de todos los campos ingresados o entregados
```

```
(JField)
```

```
        textSalarioBruto.setText("");
```

```
        textSalarioNeto.setText("");
```

```
        textCodigo.setText("");
```

```
        textNombre.setText("");
```

```
        textHoras.setText("");
```

```
        textPorcentajeRetencion.setText("");
```

```
        textValorHora.setText("");
```

```
    }
```

```
});
```

```
btnBorrar.setBounds(292, 304, 89, 23);
```

```
contentPane.add(btnBorrar);
```

```
//los dos JLabel siguientes son para hacer más estética la interfaz
```

```
JLabel lblNewLabel_7 = new JLabel("Empresa Chimuelito S.A.\r\n");
```

```
lblNewLabel_7.setFont(new Font("Rockwell Extra Bold", Font.BOLD, 16));
```

```
lblNewLabel_7.setHorizontalAlignment(SwingConstants.CENTER);
```

```
lblNewLabel_7.setBounds(192, 48, 308, 20);
```

```

contentPane.add(lblNewLabel_7);

JLabel lblNewLabel_7_1 = new JLabel("Base de datos nómina");
lblNewLabel_7_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_7_1.setFont(new Font("Rockwell Extra Bold", Font.BOLD, 20));
lblNewLabel_7_1.setBounds(182, 17, 308, 20);
contentPane.add(lblNewLabel_7_1);

JTextPane txtpnUstedRecibir = new JTextPane();
txtpnUstedRecibir.setFont(new Font("Rockwell", Font.PLAIN, 13));
txtpnUstedRecibir.setText("Usted recibirá:");
txtpnUstedRecibir.setBackground(new Color(0, 128, 64));
txtpnUstedRecibir.setBounds(459, 174, 93, 20);
contentPane.add(txtpnUstedRecibir);

//para que el usuario pueda salir de la interfaz
JButton btnSalir = new JButton("Cerrar");
btnSalir.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(WIDTH); //comando para cerrar el programa
    }
});
btnSalir.setFont(new Font("Rockwell", Font.BOLD | Font.ITALIC, 12));
btnSalir.setBackground(new Color(165, 205, 120));
btnSalir.setBounds(411, 305, 89, 23);
contentPane.add(btnSalir);
//siempre que aparece setBounds son las medidas que ocupa cada aspecto en el panel
}
}

```

Métodos:

```

package EjercicioPropuesto18;

public class MetodosP18 {
    //return devuelve los resultados
    //este primer metodo determina el salario bruto, es decir, sin descontar la retencion, es lo
    trabajo
    public static double metodoSalarioBruto(double horas, double valorHora) {
        double salarioBruto;
        salarioBruto = horas*valorHora;
        return salarioBruto;
    }
}

```

//el segundo método lo que hace es determinar cuánto salario se queda retenido según el porcentaje definido

```
public static double metodoSalarioRetenido(double porcentajeRetenido, double horas, double valorHora) {  
    double salarioRetenido;  
    salarioRetenido = (horas*valorHora)*(porcentajeRetenido)/100;  
    return salarioRetenido;  
}
```

//finalmente, el tercer método se encarga de hallar el salario neto que se le da al empleado, descontando el retenido

```
public static double metodoSalarioNeto(double salarioRetenido, double salarioBruto) {  
    double salarioNeto;  
    salarioNeto = salarioBruto - salarioRetenido;  
    return salarioNeto;  
}  
  
}
```

Resultado:



2. Ejercicio propuesto N°19:

Método principal:

```
package EjercicioPropuesto19;
```

//método principal

```
public class EjercicioP19 {  
    public static void main(String[] args) {  
        InterfazP19 fr= new InterfazP19();  
        fr.setVisible(true);  
    }  
}
```

Interfaz:

```
package EjercicioPropuesto19;

import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JOptionPane;
import java.awt.Color;
import javax.swing.border.SoftBevelBorder;
import javax.swing.border.BevelBorder;
import javax.swing.SwingConstants;
import java.awt.Font;

public class InterfazP19 extends JFrame {

    private JPanel contentPane;
    private JTextField txtLado;
    private JTextField txtPerimetro;
    private JTextField txtArea;

    /**
     * Create the frame.
     */
    public InterfazP19() {
        //siempre que sale Bounds son las medidas
        this.setTitle("Ejercicio Propuesto 19"); //poner título a la interfaz
        //setBounds son medidas
        //setBackground son el color de fondo
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 495, 352);
        contentPane = new JPanel();
        contentPane.setBackground(new Color(255, 255, 128));
        contentPane.setBorder(new SoftBevelBorder(BevelBorder.RAISED, null, null, null,
null));

        setContentPane(contentPane);
```



```
contentPane.setLayout(null);
```

indicar lo que se debe ingresar

```
JLabel lblNewLabel = new JLabel("Lado del triángulo"); //espacio creado para  
lblNewLabel.setFont(new Font("Stencil", Font.PLAIN, 13)); //tipo de letra y tamaño  
lblNewLabel.setBounds(43, 74, 139, 14);  
contentPane.add(lblNewLabel);
```

indicación

```
JLabel lblNewLabel_1 = new JLabel("Perímetro"); //resultado del perímetro,  
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);  
lblNewLabel_1.setFont(new Font("Stencil", Font.PLAIN, 13));  
lblNewLabel_1.setBounds(70, 136, 83, 14);  
contentPane.add(lblNewLabel_1);
```

y tamaño

```
JLabel lblNewLabel_2 = new JLabel("Altura"); //resultado de la altura, indicación  
lblNewLabel_2.setHorizontalAlignment(SwingConstants.CENTER); //tipo de letra  
lblNewLabel_2.setFont(new Font("Stencil", Font.PLAIN, 13));  
lblNewLabel_2.setBounds(70, 192, 83, 14);  
contentPane.add(lblNewLabel_2);
```

tamaño

```
JLabel lblNewLabel_3 = new JLabel("Área"); //resultado del area, indicación  
lblNewLabel_3.setFont(new Font("Stencil", Font.PLAIN, 13)); //tipo de letra y  
lblNewLabel_3.setHorizontalAlignment(SwingConstants.CENTER);  
lblNewLabel_3.setBounds(87, 248, 46, 14);  
contentPane.add(lblNewLabel_3);
```

```
//los valores siguientes son para ingresar los datos por teclado  
txtLado = new JTextField();  
txtLado.setHorizontalAlignment(SwingConstants.LEFT);  
txtLado.setFont(new Font("Stencil", Font.PLAIN, 13));  
txtLado.setBounds(192, 70, 168, 20);  
contentPane.add(txtLado);  
txtLado.setColumns(10);
```

```
txtPerimetro = new JTextField();  
txtPerimetro.setHorizontalAlignment(SwingConstants.CENTER);  
txtPerimetro.setFont(new Font("Stencil", Font.PLAIN, 13));  
txtPerimetro.setColumns(10);  
txtPerimetro.setBounds(24, 161, 168, 20);  
contentPane.add(txtPerimetro);
```

```

    JTextField txtAltura = new JTextField();
    txtAltura.setHorizontalAlignment(SwingConstants.CENTER);
    txtAltura.setFont(new Font("Stencil", Font.PLAIN, 13));
    txtAltura.setColumns(10);
    txtAltura.setBounds(24, 217, 168, 20);
    contentPane.add(txtAltura);

    txtArea = new JTextField();
    txtArea.setHorizontalAlignment(SwingConstants.CENTER);
    txtArea.setFont(new Font("Stencil", Font.PLAIN, 13));
    txtArea.setColumns(10);
    txtArea.setBounds(24, 273, 168, 20);
    contentPane.add(txtArea);

    //para el boton de calcular
    JButton btnCalcular = new JButton("Calcular");
    btnCalcular.setBackground(new Color(255, 128, 0));
    btnCalcular.setFont(new Font("Stencil", Font.ITALIC, 13));
    btnCalcular.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            double lado;
            try {
                //obtener el valor que entró el usuario
                lado = Double.parseDouble(txtLado.getText());
                //mostrar los resultados

                txtPerimetro.setText(String.valueOf(MetodosP19.calcularPerimetro(lado)));

                txtAltura.setText(String.valueOf(MetodosP19.calcularAltura(lado)));
                txtArea.setText(String.valueOf(MetodosP19.calcularArea(lado)));
            }
            //excepciones cuando ingresan un lado que NO es numérico
            catch(NumberFormatException ex) {
                JOptionPane.showMessageDialog(null, "El parámetro
ingresado no es un valor numérico, intente de nuevo.");
            }
        }
    });

    btnCalcular.setBounds(290, 157, 112, 30);
    contentPane.add(btnCalcular);

    //para el botón de borrar que elimina el contenido de todos los campos de la interfaz
    JButton btnBorrar = new JButton("Borrar");
    btnBorrar.setBackground(new Color(255, 128, 0));

```

```
btnBorrar.setFont(new Font("Stencil", Font.ITALIC, 13));
btnBorrar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //dejar los espacios en blanco
        txtPerimetro.setText("");
        txtAltura.setText("");
        txtArea.setText("");
        txtLado.setText("");
    }
});
btnBorrar.setBounds(290, 213, 112, 30);
contentPane.add(btnBorrar);


//un separador pero creado de manera manual
JLabel lblNewLabel_4 = new JLabel("\u0394 \u0394 \u0394 \u0394 \u0394 \u0394 \u0394 \u0394 \u0394 \u0394 \u0394 \u0394 \u0394 \u0394 \u0394 \u0394 \u0394 \u0394");
lblNewLabel_4.setForeground(new Color(255, 128, 0));
lblNewLabel_4.setBackground(new Color(240, 240, 240));
lblNewLabel_4.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_4.setBounds(24, 95, 432, 30);
contentPane.add(lblNewLabel_4);


//titulo dentro del panel de la interfaz
JLabel lblNewLabel_5 = new JLabel("C\u00e1lculos para un tri\u00e1ngulo equil\u00e1tero");
lblNewLabel_5.setFont(new Font("Stencil", Font.PLAIN, 19));
lblNewLabel_5.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_5.setBounds(37, 25, 419, 35);
contentPane.add(lblNewLabel_5);


//para cerrar la calculadora mediante un bot\u00f3n
JButton btnSalir = new JButton("Salir");
btnSalir.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(WIDTH); //comando para cerrar el programa
    }
});
btnSalir.setFont(new Font("Stencil", Font.ITALIC, 13));
btnSalir.setBackground(new Color(255, 128, 0));
btnSalir.setBounds(290, 269, 112, 30);
contentPane.add(btnSalir);
}
```

Métodos:

package EjercicioPropuesto19;

```
public class MetodosP19 {
    //return devuelve los resultados
    //primer método para calcular el perímetro del triángulo
    public static double calcularPerimetro(double lado) {
        double perimetro; //para guardar el resultado
        perimetro = lado * 3; //fórmula
        return perimetro;
    }
    //segundo método para calcular la altura del triángulo
    public static double calcularAltura(double lado) {
        double altura; //para guardar el resultado
        altura = (lado*Math.sqrt(3))/2; //fórmula
        return altura;
    }
    //tercer método para calcular el área del triángulo
    public static double calcularArea(double lado) {
        double area; //para guardar el resultado
        area = (Math.pow(lado,2)*Math.sqrt(3))/4; //fórmula
        return area;
    }
}
```

Resultado:

[illegible]

CAPÍTULO 4: Estructura decisión lógica

3. Ejercicio resuelto N°7:

Método principal:

```
package EjercicioResuelto7;

//método principal (main)
public class EjercicioR7 {
    public static void main(String[] args) {
        InterfazR7 fr= new InterfazR7();
        fr.setVisible(true);
    }
}
```

Interfaz:

```
package EjercicioResuelto7;

import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.SwingConstants;
import java.awt.Color;

public class InterfazR7 extends JFrame {

    private JPanel contentPane;
    private JTextField textA;
    private JTextField textB;
    private JTextField textResultado;
```

```

/**
 * Create the frame.
 */
public InterfazR7() {
    this.setTitle("Ejercicio Resuelto 7"); //para título de la interfaz
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 300); //dimensiones
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

    setContentPane(contentPane);
    contentPane.setLayout(null);

    //titulo dentro del panel
    JLabel lblNewLabel = new JLabel("Comparar dos números");
    lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel.setFont(new Font("Kristen ITC", Font.BOLD, 18));
    lblNewLabel.setBounds(40, 33, 352, 32);
    contentPane.add(lblNewLabel);

    //indicador A
    JLabel lblNewLabel_1 = new JLabel("A");
    lblNewLabel_1.setForeground(new Color(0, 128, 0));
    lblNewLabel_1.setFont(new Font("Kristen ITC", Font.PLAIN, 13));
    lblNewLabel_1.setBounds(40, 98, 46, 14);
    contentPane.add(lblNewLabel_1);

    //indicador B
    JLabel lblNewLabel_2 = new JLabel("B");
    lblNewLabel_2.setForeground(new Color(0, 128, 0));
    lblNewLabel_2.setFont(new Font("Kristen ITC", Font.PLAIN, 13));
    lblNewLabel_2.setBounds(247, 98, 46, 14);
    contentPane.add(lblNewLabel_2);

    //donde se ingresa A
    textA = new JTextField();
    textA.setBackground(new Color(128, 255, 128));
    textA.setForeground(new Color(0, 0, 0));
    textA.setHorizontalAlignment(SwingConstants.CENTER);
    textA.setFont(new Font("Kristen ITC", Font.PLAIN, 13));
    textA.setBounds(85, 95, 86, 20);
    contentPane.add(textA);
    textA.setColumns(10);

```

```

//donde se ingresa B
textB = new JTextField();
textB.setBackground(new Color(128, 255, 128));
textB.setForeground(new Color(0, 0, 0));
textB.setHorizontalAlignment(SwingConstants.CENTER);
textB.setFont(new Font("Kristen ITC", Font.PLAIN, 13));
textB.setText("");
textB.setBounds(286, 95, 86, 20);
contentPane.add(textB);
textB.setColumns(10);

//crear botón comparación
JButton btnComparar = new JButton("Comparar");
btnComparar.setFont(new Font("Kristen ITC", Font.BOLD, 13));
btnComparar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            //definir variables
            double A;
            double B;

            //leerlas de los valores ingresados por el usuario
            A = Double.parseDouble(textA.getText());
            B = Double.parseDouble(textB.getText());

            //condicional
            if (MetodosR7.comparar(A,B) == A) {
                textResultado.setText("A es mayor que B");
            }
            else if (MetodosR7.comparar(A,B) == B) {
                textResultado.setText("B es mayor que A");
            }
            else {
                textResultado.setText("A es igual a B");
            }
        }
        //excepciones
        catch(NumberFormatException ex) {
            JOptionPane.showMessageDialog(null, "Los
valores de A y B ingresados, no son numéricos");
        }
    }
});

```

```

btnComparar.setBounds(62, 215, 109, 23);
contentPane.add(btnComparar);

//lugar para que aparezca el resultado
textResultado = new JTextField();
textResultado.setBackground(new Color(128, 255, 128));
textResultado.setForeground(new Color(0, 0, 0));
textResultado.setHorizontalAlignment(SwingConstants.CENTER);
textResultado.setFont(new Font("Kristen ITC", Font.PLAIN, 13));
textResultado.setBounds(85, 158, 256, 32);
contentPane.add(textResultado);
textResultado.setColumns(10);

//indicador resultado
JLabel lblNewLabel_3 = new JLabel("Resultado");
lblNewLabel_3.setForeground(new Color(0, 128, 0));
lblNewLabel_3.setFont(new Font("Kristen ITC", Font.PLAIN, 13));
lblNewLabel_3.setBounds(179, 132, 69, 14);
contentPane.add(lblNewLabel_3);

//eliminar los valores ingresados y obtenidos
JButton btnBorrar = new JButton("Borrar");
btnBorrar.setFont(new Font("Kristen ITC", Font.BOLD, 13));
btnBorrar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //dejar espacios en blanco
        textResultado.setText("");
        textA.setText("");
        textB.setText("");
    }
});
btnBorrar.setBounds(181, 215, 89, 23);
contentPane.add(btnBorrar);

//botón para cerrar interfaz
JButton btnSalir = new JButton("Salir");
btnSalir.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(WIDTH); //comando para cerrar el programa
    }
});
btnSalir.setFont(new Font("Kristen ITC", Font.BOLD, 13));
btnSalir.setBounds(280, 216, 89, 23);
contentPane.add(btnSalir);

```



```
    }  
}
```

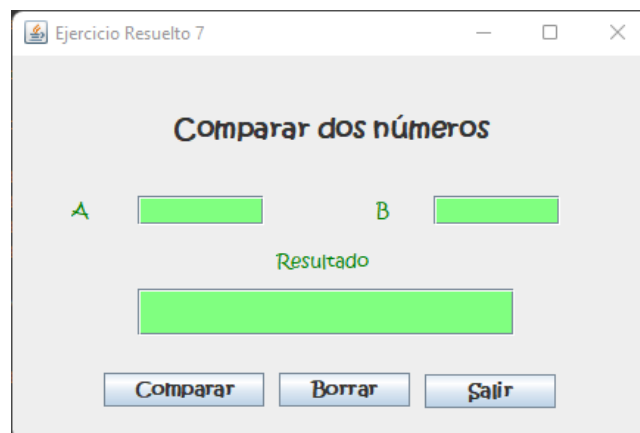
Métodos:

```
package EjercicioResuelto7;
```

```
import javax.swing.JOptionPane;
```

```
public class MetodosR7 {  
    public static double comparar(double A,double B) {  
  
        if (A>B) {  
            double r;  
            r = A;  
            return r;  
        }  
        else if (A<B) {  
            double r;  
            r = B;  
            return r;  
        }  
        else {  
            double r;  
            r = 0;  
            return r;  
        }  
    }  
}
```

Resultado:



4. Ejercicio resuelto N°10:

Método principal:

```
package EjercicioResuelto10;

//método principal
public class EjerciorR10 {
    public static void main(String[] args) {
        InterfazR10 fr= new InterfazR10();
        fr.setVisible(true);
    }
}
```

Interfaz:

```
package EjercicioResuelto10;

import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.Font;
import javax.swing.SwingConstants;
import java.awt.Color;

public class InterfazR10 extends JFrame {

    private JPanel contentPane;
    private JTextField textResultado;
    private JTextField textNumero;
    private JTextField textNombre;
    private JTextField textPatrimonio;
    private JTextField textEstrato;
    private JButton btnBorrar;
    private JButton btnCerrar;
    private JLabel lblNewLabel_4;
    private JLabel lblNewLabel_5;
```

```

private JLabel lblNewLabel_6;

/**
 * Create the frame.
 */
public InterfazR10() {
    //setBounds -> dimensiones, setFound -> fuentes
    this.setTitle("Ejercicio Resuelto 10"); //para título de la interfaz
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 877, 352);
    contentPane = new JPanel();
    contentPane.setBackground(new Color(255, 255, 255));
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

    setContentPane(contentPane);
    contentPane.setLayout(null);

    //indicadores
    JLabel lblNewLabel = new JLabel("Número de matrícula");
    lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel.setFont(new Font("Verdana", Font.PLAIN, 12));
    lblNewLabel.setBounds(38, 129, 150, 14);
    contentPane.add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("Nombre");
    lblNewLabel_1.setFont(new Font("Verdana", Font.PLAIN, 12));
    lblNewLabel_1.setBounds(38, 104, 184, 14);
    contentPane.add(lblNewLabel_1);

    JLabel lblNewLabel_2 = new JLabel("Patrimonio ($)");
    lblNewLabel_2.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel_2.setFont(new Font("Verdana", Font.PLAIN, 12));
    lblNewLabel_2.setBounds(38, 185, 150, 14);
    contentPane.add(lblNewLabel_2);

    JLabel lblNewLabel_3 = new JLabel("Estrato social");
    lblNewLabel_3.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel_3.setFont(new Font("Verdana", Font.PLAIN, 12));
    lblNewLabel_3.setBounds(38, 241, 150, 14);
    contentPane.add(lblNewLabel_3);

    textResultado = new JTextField();
    textResultado.setForeground(new Color(255, 255, 255));
    textResultado.setBackground(new Color(128, 0, 0));

```

```

textResultado.setFont(new Font("Verdana", Font.ITALIC, 12));
textResultado.setHorizontalAlignment(SwingConstants.CENTER);
textResultado.setBounds(213, 170, 602, 44);
contentPane.add(textResultado);
textResultado.setColumns(10);

//botón para encontrar el patrimonio
JButton btnNewButton = new JButton("Determinar");
btnNewButton.setForeground(new Color(128, 0, 0));
btnNewButton.setFont(new Font("Verdana", Font.BOLD, 13));
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            //variables pedidas por teclado
            double numero;
            String nombre;
            double patrimonio;
            double estrato;

            //leer los datos ingresados
            nombre = textNombre.getText();
            numero = Double.parseDouble(textNumero.getText());
            estrato = Double.parseDouble(textEstrato.getText());
            patrimonio = Double.parseDouble(textPatrimonio.getText());

            //mostrar resultados
            textResultado.setText("El/la estudiante " + nombre + ",con
número de matricula " + (int)numero + ",debe pagar este semestre $" +
MetodosR10.pagoMatricula(patrimonio,estrato));
        }

        //excepciones
        catch(NumberFormatException ex) {
            JOptionPane.showMessageDialog(null, "Los
valores ingresados no concuerdan con el formato. Intente nuevamente.");
        }
    }
});

btnNewButton.setBounds(301, 232, 115, 23);
contentPane.add(btnNewButton);

//ingresar por teclado

```

```
textNumero = new JTextField();
textNumero.setForeground(new Color(255, 255, 255));
textNumero.setBackground(new Color(128, 0, 0));
textNumero.setFont(new Font("Verdana", Font.PLAIN, 12));
textNumero.setHorizontalAlignment(SwingConstants.CENTER);
textNumero.setBounds(38, 154, 150, 20);
contentPane.add(textNumero);
textNumero.setColumns(10);
```

```
textNombre = new JTextField();
textNombre.setForeground(new Color(255, 255, 255));
textNombre.setBackground(new Color(128, 0, 0));
textNombre.setFont(new Font("Verdana", Font.PLAIN, 12));
textNombre.setHorizontalAlignment(SwingConstants.CENTER);
textNombre.setBounds(126, 101, 189, 20);
contentPane.add(textNombre);
textNombre.setColumns(10);
```

```
textPatrimonio = new JTextField();
textPatrimonio.setForeground(new Color(255, 255, 255));
textPatrimonio.setBackground(new Color(128, 0, 0));
textPatrimonio.setFont(new Font("Verdana", Font.PLAIN, 12));
textPatrimonio.setHorizontalAlignment(SwingConstants.CENTER);
textPatrimonio.setBounds(38, 210, 150, 20);
contentPane.add(textPatrimonio);
textPatrimonio.setColumns(10);
```

```
textEstrato = new JTextField();
textEstrato.setForeground(new Color(255, 255, 255));
textEstrato.setBackground(new Color(128, 0, 0));
textEstrato.setFont(new Font("Verdana", Font.PLAIN, 12));
textEstrato.setHorizontalAlignment(SwingConstants.CENTER);
textEstrato.setBounds(38, 266, 150, 20);
contentPane.add(textEstrato);
textEstrato.setColumns(10);
```

```
//botón para eliminar los contenidos
btnBorrar = new JButton("Borrar");
btnBorrar.setForeground(new Color(128, 0, 0));
btnBorrar.setFont(new Font("Verdana", Font.BOLD, 13));
btnBorrar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //dejar espacios en blanco
        textEstrato.setText("");
    }
});
```

```

        textPatrimonio.setText("");
        textNombre.setText("");
        textNumero.setText("");
        textResultado.setText("");
    }
});
btnBorrar.setBounds(468, 232, 89, 23);
contentPane.add(btnBorrar);

//para salir del programa con un botón
btnCerrar = new JButton("Cerrar");
btnCerrar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(WIDTH); //comando para cerrar el programa
    }
});
btnCerrar.setForeground(new Color(128, 0, 0));
btnCerrar.setFont(new Font("Verdana", Font.BOLD, 13));
btnCerrar.setBounds(614, 232, 89, 23);
contentPane.add(btnCerrar);

//encabezados de la interfaz
lblNewLabel_4 = new JLabel("Pago de matrícula");
lblNewLabel_4.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_4.setFont(new Font("Verdana", Font.BOLD, 17));
lblNewLabel_4.setBounds(271, 11, 197, 23);
contentPane.add(lblNewLabel_4);

Medellín");
lblNewLabel_5 = new JLabel("Universidad Nacional de Colombia - Sede
Medellín");
lblNewLabel_5.setForeground(new Color(128, 0, 0));
lblNewLabel_5.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_5.setFont(new Font("Verdana", Font.BOLD, 14));
lblNewLabel_5.setBounds(186, 36, 390, 23);
contentPane.add(lblNewLabel_5);

lblNewLabel_6 = new JLabel(";Bienvenido!");
lblNewLabel_6.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_6.setFont(new Font("Verdana", Font.BOLD | Font.ITALIC, 12));
lblNewLabel_6.setBounds(186, 70, 390, 23);
contentPane.add(lblNewLabel_6);
}

}

```

Métodos:

```
package EjercicioResuelto10;
```

```
import javax.swing.JOptionPane;
```

```
//solo hay un método que permite determinar el valor a cancelar
```

```
public class MetodosR10 {  
    public static double pagoMatricula(double plata,double estrato) {  
        double pago; //crear variable para almacenar el resultado  
        double general = 50000; //cobrado a cualquier estudiante  
        pago = 0; //iniciar variable  
        if ((plata > 2000000) && (estrato > 3)) { //condicional  
            pago = (0.03*plata) + general; //se aplica la indicación dada  
            return pago; //resultado  
        }else {  
            pago = general; //se paga solo lo general  
            return pago; //resultado  
        }  
    }  
}
```

Resultado:

The screenshot shows a Java Swing window titled "Ejercicio Resuelto 10". The window has a title bar with standard minimize, maximize, and close buttons. The main content area has a title "Pago de matrícula" in bold, followed by "Universidad Nacional de Colombia - Sede Medellín" in red. Below this is a welcome message "¡Bienvenido!". There are four input fields with labels: "Nombre", "Número de matrícula", "Patrimonio (\$)", and "Estrato social". The "Número de matrícula" field is highlighted in red. At the bottom, there are three buttons: "Determinar", "Borrar", and "Cerrar".

5. Ejercicio propuesto N°22:**Método principal:**

```
package EjercicioPropuesto22;
```

```
//principal
```

```
public class EjercicioP22 {  
    public static void main(String[] args) {
```

```

        InterfazP22 fr= new InterfazP22();
        fr.setVisible(true);
    }

```

```

}

```

Interfaz:

```

package EjercicioPropuesto22;

```

```

import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

```

```

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

```

```

import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.Font;
import javax.swing.SwingConstants;
import java.awt.Color;
import javax.swing.JTextPane;

```

```

public class InterfazP22 extends JFrame {

```

```

    private JPanel contentPane;
    private JTextField textNombre;
    private JTextField textSalario;
    private JTextField textHoras;
    private JTextField textResultado;

```

```

    /**

```

```

     * Create the frame.

```

```

     */

```

```

    public InterfazP22() { //muestra en ventana
        //siempre que sale Bounds son las medidas
        this.setTitle("Ejercicio Propuesto 22"); //poner título a la interfaz
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 457, 331);
        contentPane = new JPanel();
        contentPane.setBackground(new Color(255, 193, 255));
    }
}

```



```
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);
contentPane.setLayout(null);

JLabel lblNewLabel = new JLabel("Nombre"); //indicador de nombre
lblNewLabel.setFont(new Font("SimSun-ExtB", Font.BOLD, 13));
lblNewLabel.setBounds(33, 39, 46, 14);
contentPane.add(lblNewLabel);

JLabel lblSalarioPorHora = new JLabel("Salario por hora ($)"); //indicador de salario
lblSalarioPorHora.setFont(new Font("SimSun-ExtB", Font.BOLD, 13));
lblSalarioPorHora.setBounds(33, 64, 187, 14);
contentPane.add(lblSalarioPorHora);

JLabel lblCantidadHorasTrabajadas = new JLabel("Cantidad de horas trabajadas  
(mes)"); //indicador horas trabajadas
lblCantidadHorasTrabajadas.setHorizontalAlignment(SwingConstants.CENTER);
lblCantidadHorasTrabajadas.setFont(new Font("SimSun-ExtB", Font.BOLD, 13));
lblCantidadHorasTrabajadas.setBounds(33, 89, 339, 14);
contentPane.add(lblCantidadHorasTrabajadas);

textNombre = new JTextField(); //ingreso valor por teclado
textNombre.setBounds(140, 36, 258, 20);
contentPane.add(textNombre);
textNombre.setColumns(10);

textSalario = new JTextField(); //ingreso valor por teclado
textSalario.setColumns(10);
textSalario.setBounds(236, 61, 162, 20);
contentPane.add(textSalario);

textHoras = new JTextField(); //ingreso valor por teclado
textHoras.setColumns(10);
textHoras.setBounds(140, 114, 162, 20);
contentPane.add(textHoras);

textResultado = new JTextField(); //salida del resultado en espacio de texto
textResultado.setHorizontalAlignment(SwingConstants.CENTER);
textResultado.setFont(new Font("SimSun", Font.BOLD, 13));
textResultado.setForeground(new Color(255, 255, 255));
textResultado.setBackground(new Color(64, 0, 64));
textResultado.setBounds(33, 187, 365, 28);
contentPane.add(textResultado);
```

```

textResultado.setColumns(10);

//para el botón de calcular que determina los valores requeridos
JButton btnCalcular = new JButton("Calcular");
btnCalcular.setFont(new Font("SimSun", Font.BOLD, 14));
btnCalcular.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            double salarioHora; //por hora
            String nombre; //del empleado
            double horasMes; //trabajadas mensuales

            //capturar valores ingresados
            nombre = textNombre.getText();
            salarioHora = Double.parseDouble(textSalario.getText());
            horasMes = Double.parseDouble(textHoras.getText());

            //condicional para saber lo que se debe imprimir
            if (MetodosP22.determinarPago(salarioHora, horasMes) ==
0) {
                textResultado.setText(nombre); //solo entrega el
nombre
            }
            else {
                textResultado.setText(nombre + " su salario es " +
MetodosP22.determinarPago(salarioHora, horasMes)); //si el pago es mayor a 450 mil
            }
        }
        catch(NumberFormatException ex) {
            //en caso de que se ingresen valores que no sean
numéricos, no se puede hacer el programa
JOptionPane.showMessageDialog(null, "Los
parámetros ingresados no son valores numéricos, intente de nuevo.");
        }
    }
});
btnCalcular.setBounds(33, 226, 117, 35);
contentPane.add(btnCalcular);

JButton btnLimpiar = new JButton("Limpiar"); //botón para volver a rellenar campos
btnLimpiar.setFont(new Font("SimSun", Font.BOLD, 14));
btnLimpiar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```

```

//eliminar el contenido de los campos ingresados y
entregados(resultado)

        textResultado.setText("");
        textSalario.setText("");
        textNombre.setText("");
        textHoras.setText("");
    }
});
btnLimpiar.setBounds(160, 226, 117, 35);
contentPane.add(btnLimpiar);

//de acá en adelante, son partes que se añaden para hacer más amigable la interfaz
con el usuario
JLabel lblNewLabel_1 = new JLabel("Información para empleados");
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1.setFont(new Font("SimSun", Font.BOLD, 18));
lblNewLabel_1.setBounds(73, 11, 263, 23);
contentPane.add(lblNewLabel_1);

JTextPane txtpnAContinuacinPuede = new JTextPane(); //explicación del ejercicio
txtpnAContinuacinPuede.setFont(new Font("SimSun", Font.BOLD, 11));
txtpnAContinuacinPuede.setText("A continuación puede ver el salario que recibirá
siempre y cuando el resultado sea mayor a $450.000.");
txtpnAContinuacinPuede.setBounds(33, 146, 365, 35);
contentPane.add(txtpnAContinuacinPuede);

//para que el empleado pueda cerrar la interfaz mediante un botón
JButton btnCerrar = new JButton("Cerrar");
btnCerrar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(WIDTH); //comando para cerrar el programa
    }
});
btnCerrar.setFont(new Font("SimSun", Font.BOLD, 14));
btnCerrar.setBounds(287, 226, 117, 35);
contentPane.add(btnCerrar);
}
}

```

Métodos:

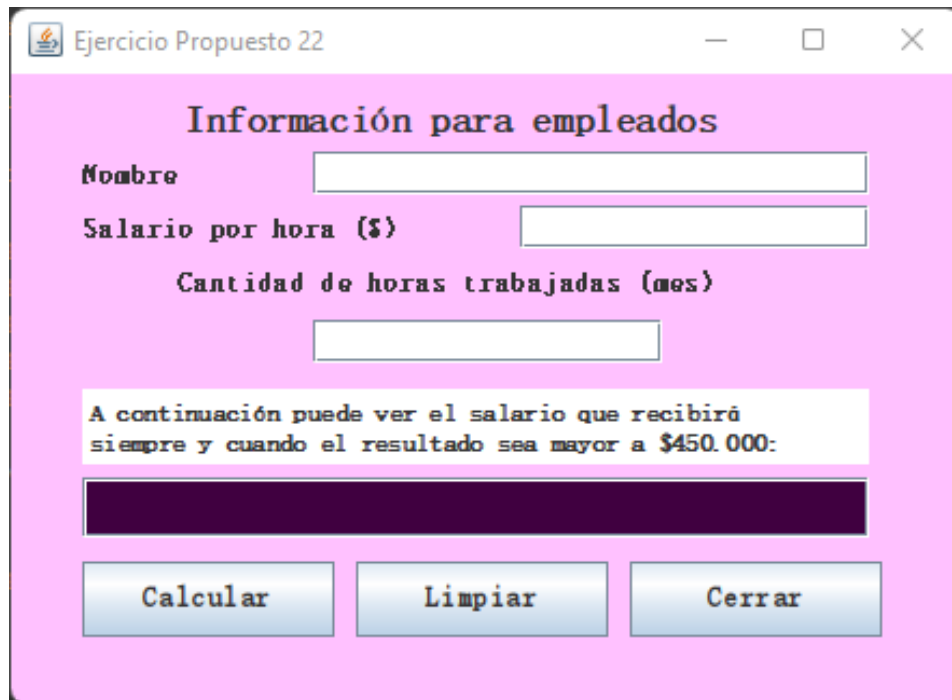
```
package EjercicioPropuesto22;
```

```
import javax.swing.JOptionPane;
```

//solo hay un metodo en este ejercicio, donde se calcula el pago que se le da al trabajador dependiendo de las horas y el valor de cada una, con esto se determina lo que se imprime

```
public class MetodosP22 {  
    public static double determinarPago(double salario, double horas) {  
        double pagoMensual; //donde se va a almacenar el calculo  
        pagoMensual = salario*horas; //multiplicación  
        if (pagoMensual > 450000) { //condicional  
            return pagoMensual; //retorna el valor  
        }else {  
            pagoMensual = 0; //el pago toma el valor de cero (de manera simbólica para el ejercicio)  
            return pagoMensual;  
        }  
    }  
}
```

Resultado:



Ejercicio Propuesto 22

Información para empleados

Nombre

Salario por hora (\$)

Cantidad de horas trabajadas (mes)

A continuación puede ver el salario que recibirá siempre y cuando el resultado sea mayor a \$450.000:

Calcular Limpiar Cerrar

6. Ejercicio propuesto N°23:

Método principal:

```
package EjercicioPropuesto23;
```

//código principal (main)

```
public class EjercicioP23 {  
    public static void main(String[] args) {  
        InterfazP23 fr= new InterfazP23();
```

```

        fr.setVisible(true);
    }

}

```

Interfaz:

```

package EjercicioPropuesto23;

import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Font;
import javax.swing.JEditorPane;
import javax.swing.JSeparator;
import java.awt.Color;
import javax.swing.SwingConstants;

public class InterfazP23 extends JFrame {

    private JPanel contentPane;
    private JTextField txtA;
    private JTextField txtB;
    private JTextField txtC;
    private JTextField txtResultado;

    /**
     * Create the frame.
     */
    public InterfazP23() {
        //siempre que sale Bounds son las medidas
        this.setTitle("Ejercicio Propuesto 23"); //poner título a la interfaz
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 519, 399);
        contentPane = new JPanel();
        contentPane.setBackground(new Color(251, 170, 208));
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    }
}

```

```
setContentPane(contentPane);
contentPane.setLayout(null);
```

```
JLabel lblNewLabel = new JLabel("a"); //ingresar por teclado
lblNewLabel.setFont(new Font("Yu Gothic Medium", Font.BOLD, 12));
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel.setBounds(40, 110, 26, 20);
contentPane.add(lblNewLabel);
```

```
JLabel lblIngreseValorDe = new JLabel("b"); //ingresar por teclado
lblIngreseValorDe.setHorizontalAlignment(SwingConstants.CENTER);
lblIngreseValorDe.setFont(new Font("Yu Gothic Medium", Font.BOLD, 12));
lblIngreseValorDe.setBounds(40, 141, 26, 20);
contentPane.add(lblIngreseValorDe);
```

```
JLabel lblIngreseValorDe_1 = new JLabel("c"); //ingresar por teclado
lblIngreseValorDe_1.setHorizontalAlignment(SwingConstants.CENTER);
lblIngreseValorDe_1.setFont(new Font("Yu Gothic Medium", Font.BOLD, 12));
lblIngreseValorDe_1.setBounds(40, 172, 26, 20);
contentPane.add(lblIngreseValorDe_1);
```

```
txtA = new JTextField();
txtA.setBounds(76, 107, 86, 20);
contentPane.add(txtA);
txtA.setColumns(10);
```

```
txtB = new JTextField();
txtB.setBounds(76, 138, 86, 20);
contentPane.add(txtB);
txtB.setColumns(10);
```

```
txtC = new JTextField();
txtC.setBounds(76, 172, 86, 20);
contentPane.add(txtC);
txtC.setColumns(10);
```

```
//botón para encontrar el algoritmo
JButton btnDeterminar = new JButton("Calcular");
btnDeterminar.setForeground(new Color(255, 0, 128));
btnDeterminar.setFont(new Font("Yu Gothic Medium", Font.BOLD | Font.ITALIC,
```

13));

```
btnDeterminar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
```

```

double A, B, C, raizpositiva, raiznegativa; //parámetros de la
ecuación y respuestas

//obtener valores ingresados por teclado en la interfaz
A = Double.parseDouble(txtA.getText());
B = Double.parseDouble(txtB.getText());
C = Double.parseDouble(txtC.getText());
double D = Math.pow(B, 2) - (4 * A * C); //con la fórmula del
estudiante

System.out.println(D); //imprimir el determinante (D)

//condicional según el valor que tomó el determinante
if(D < 0) {
    txtResultado.setText("Las soluciones encontradas para la
ecuación cuadrática son soluciones complejas.");
}
else if(D == 0) {
    txtResultado.setText(String.valueOf("x      =      "      +
MetodosP23.raizCuadrada1(A, B, D)));
}
else {
    raizpositiva = MetodosP23.raizCuadrada2(A, B, D);
    raiznegativa = MetodosP23.raizCuadrada3(A, B, D);
    txtResultado.setText("x1 = " + String.valueOf(raizpositiva)
+ ", x2 = " + String.valueOf(raiznegativa));
}
}

});
btnDeterminar.setBounds(73, 303, 89, 30);
contentPane.add(btnDeterminar);

//para eliminar el contenido de los campos, se inserta un botón de borrar
JButton btnBorrar = new JButton("Borrar");
btnBorrar.setForeground(new Color(255, 0, 128));
btnBorrar.setFont(new Font("Yu Gothic Medium", Font.BOLD | Font.ITALIC, 13));
btnBorrar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //dejar en blanco los textos ingresados y calculados
        txtA.setText("");
        txtB.setText("");
        txtC.setText("");
        txtResultado.setText("");
    }
});

```

```

btnBorrar.setBounds(210, 303, 89, 30);
contentPane.add(btnBorrar);

//indicador de que abajo aparecen las raices
JLabel lblNewLabel_10 = new JLabel("Posibles raíces:");
lblNewLabel_10.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_10.setFont(new Font("Yu Gothic Medium", Font.BOLD, 12));
lblNewLabel_10.setBounds(30, 227, 440, 20);
contentPane.add(lblNewLabel_10);

//raices encontradas
txtResultado = new JTextField();
txtResultado.setHorizontalAlignment(SwingConstants.CENTER);
txtResultado.setFont(new Font("Yu Gothic UI Semibold", Font.PLAIN, 12));
txtResultado.setBackground(new Color(251, 170, 208));
txtResultado.setBounds(52, 258, 407, 31);
contentPane.add(txtResultado);
txtResultado.setColumns(10);

//separador
JSeparator separator = new JSeparator();
separator.setBackground(new Color(244, 17, 125));
separator.setBounds(30, 68, 440, 13);
contentPane.add(separator);

//nombre del programa y lo que permite hacer
JLabel lblNewLabel_1 = new JLabel("RaizNath");
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1.setFont(new Font("Yu Gothic UI Semibold", Font.BOLD, 18));
lblNewLabel_1.setBounds(195, 11, 106, 23);
contentPane.add(lblNewLabel_1);

JLabel lblNewLabel_1_1 = new JLabel("Su calculadora para encontrar las raíces de
una ecuación cuadrática");
lblNewLabel_1_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1_1.setFont(new Font("Yu Gothic UI Semibold", Font.BOLD, 11));
lblNewLabel_1_1.setBounds(76, 34, 353, 23);
contentPane.add(lblNewLabel_1_1);

//texto para usuario
JLabel lblNewLabel_1_1_1 = new JLabel("Por favor, ingrese:");
lblNewLabel_1_1_1.setHorizontalAlignment(SwingConstants.LEFT);
lblNewLabel_1_1_1.setFont(new Font("Yu Gothic UI Semibold", Font.BOLD, 11));
lblNewLabel_1_1_1.setBounds(30, 82, 106, 23);

```



```

contentPane.add(lblNewLabel_1_1_1);

//modelo de ecuación cuadrática
JLabel lblNewLabel_1_2 = new JLabel("Ecuación cuadrática:");
lblNewLabel_1_2.setBackground(new Color(248, 114, 177));
lblNewLabel_1_2.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1_2.setFont(new Font("Yu Gothic UI Semibold", Font.BOLD, 16));
lblNewLabel_1_2.setBounds(248, 107, 195, 23);
contentPane.add(lblNewLabel_1_2);

JLabel lblNewLabel_1_2_1 = new JLabel("ax^2+bx+c");
lblNewLabel_1_2_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1_2_1.setFont(new Font("Yu Gothic UI Semibold", Font.BOLD, 16));
lblNewLabel_1_2_1.setBounds(248, 141, 195, 23);
contentPane.add(lblNewLabel_1_2_1);

//separador
JSeparator separator_1 = new JSeparator();
separator_1.setBackground(new Color(244, 17, 125));
separator_1.setBounds(30, 203, 440, 13);
contentPane.add(separator_1);

//para salir de la interfaz con un botón
JButton btnCerrar = new JButton("Cerrar");
btnCerrar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(WIDTH); //comando para cerrar el programa
    }
});
btnCerrar.setForeground(new Color(255, 0, 128));
btnCerrar.setFont(new Font("Yu Gothic Medium", Font.BOLD | Font.ITALIC, 13));
btnCerrar.setBounds(340, 304, 89, 30);
contentPane.add(btnCerrar);
}
}

```

Métodos:

```
package EjercicioPropuesto23;
```

```

public class MetodosP23 {
    //cada método es un caso según el valor del determinante
    //si el determinante es 0
    public static double raizCuadrada1(double A,double B,double D) {

```

```

        return ((-B)/(2 * A));
    }
    //si el determiante es positivo
    public static double raizCuadrada2(double A,double B,double D) {
        return ((-B) + Math.sqrt(D))/(2 * A);
    }
    //si el determinante es negativo
    public static double raizCuadrada3(double A,double B,double D) {
        return ((-B) - Math.sqrt(D)) / (2 * A);
    }
}

```

Resultado:

The screenshot shows a Java Swing window titled "Ejercicio Propuesto 23". Inside the window, there is a pink background with the title "RaizNath" and the subtitle "Su calculadora para encontrar las raíces de una ecuación cuadrática". Below this, there is a section labeled "Por favor, ingrese:" with three input fields for coefficients "a", "b", and "c". To the right of these fields, the text "Ecuación cuadrática:" is followed by the formula ax^2+bx+c . Below the input fields, there is a section labeled "Posibles raíces:" with a large empty text area for the results. At the bottom of the window, there are three buttons: "Calcular", "Borrar", and "Cerrar".

7. Ejercicio propuesto N°40:

Método principal:

```
package EjercicioPropuesto40;
```

```
//código principal que contiene el main
```

```

public class EjercicioP40 {
    public static void main(String[] args) {
        InterfazP40 fr= new InterfazP40();
        fr.setVisible(true);
    }
}

```

Interfaz:

```

package EjercicioPropuesto40;

import java.awt.EventQueue;
import javax.swing.JOptionPane;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JList;
import javax.swing.JToggleButton;
import javax.swing.JSpinner;
import javax.swing.JLabel;
import javax.swing.JTextPane;
import javax.swing.JTextField;
import javax.swing.JSeparator;
import java.awt.Color;
import javax.swing.JSlider;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.util.FormatterClosedException;
import java.awt.event.ActionEvent;
import java.awt.Font;
import javax.swing.SwingConstants;

public class InterfazP40 extends JFrame {
    private JPanel contentPane;
    private JTextField text3;
    private JTextField text1;
    private JTextField text2;
    private JTextField text4;
    private JTextField text5;
    private JTextField text6;
    private JTextField text7;
    private JLabel lblNewLabel_7;
    private JLabel lblNewLabel_8;
    private JLabel lblNewLabel_9;
}

```

```

private JTextField textR1;
private JTextField textP21;
private JTextField textP31;
private JTextField textR2;
private JTextField textP22;
private JTextField textP32;
private JTextField textR3;
private JTextField textP23;
private JTextField textP33;
private JTextField textR4;
private JTextField textP24;
private JTextField textP34;
private JTextField textR5;
private JTextField textP25;
private JTextField textP35;
private JTextField textR6;
private JTextField textP26;
private JTextField textP36;
private JTextField textR7;
private JTextField textP27;
private JTextField textP37;
private JLabel lblNewLabel_1;
private JLabel lblNewLabel_2;
private JLabel lblNewLabel_3;
private JLabel lblNewLabel_4;
private JLabel lblNewLabel_5;
private JLabel lblNewLabel_11;
private JButton btnCerrar;
private JSeparator separator;
private JLabel lblNewLabel_10;

/**
 * Create the frame.
 */
public InterfazP40() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 681, 432);
    contentPane = new JPanel();
    contentPane.setBackground(new Color(174, 174, 255));
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

    setContentPane(contentPane);
    contentPane.setLayout(null);

```

```
JLabel lblNewLabel_6 = new JLabel("7");
lblNewLabel_6.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 13));
lblNewLabel_6.setHorizontalAlignment(SwingConstants.RIGHT);
lblNewLabel_6.setBounds(33, 277, 83, 14);
contentPane.add(lblNewLabel_6);
```

```
text3 = new JTextField();
text3.setHorizontalAlignment(SwingConstants.CENTER);
text3.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
text3.setColumns(10);
text3.setBounds(126, 152, 86, 20);
contentPane.add(text3);
```

```
JLabel lblNewLabel = new JLabel("1");
lblNewLabel.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 13));
lblNewLabel.setHorizontalAlignment(SwingConstants.RIGHT);
lblNewLabel.setBounds(43, 92, 73, 14);
contentPane.add(lblNewLabel);
```

```
text1 = new JTextField();
text1.setHorizontalAlignment(SwingConstants.CENTER);
text1.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
text1.setColumns(10);
text1.setBounds(126, 90, 86, 20);
contentPane.add(text1);
```

```
text2 = new JTextField();
text2.setHorizontalAlignment(SwingConstants.CENTER);
text2.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
text2.setColumns(10);
text2.setBounds(126, 121, 86, 20);
contentPane.add(text2);
```

```
text4 = new JTextField();
text4.setHorizontalAlignment(SwingConstants.CENTER);
text4.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
text4.setColumns(10);
text4.setBounds(126, 183, 86, 20);
contentPane.add(text4);
```

```
text5 = new JTextField();
text5.setHorizontalAlignment(SwingConstants.CENTER);
text5.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
text5.setColumns(10);
```

```
text5.setBounds(126, 214, 86, 20);
contentPane.add(text5);
```

```
text6 = new JTextField();
text6.setHorizontalAlignment(SwingConstants.CENTER);
text6.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
text6.setColumns(10);
text6.setBounds(126, 244, 86, 20);
contentPane.add(text6);
```

```
text7 = new JTextField();
text7.setHorizontalAlignment(SwingConstants.CENTER);
text7.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
text7.setColumns(10);
text7.setBounds(126, 275, 86, 20);
contentPane.add(text7);
```

```
lblNewLabel_7 = new JLabel("RAÍZ CUADRADA");
lblNewLabel_7.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_7.setFont(new Font("Berlin Sans FB Demi", Font.BOLD, 13));
lblNewLabel_7.setBounds(222, 64, 125, 14);
contentPane.add(lblNewLabel_7);
```

```
lblNewLabel_8 = new JLabel("POTENCIA CUADRADA");
lblNewLabel_8.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_8.setFont(new Font("Berlin Sans FB Demi", Font.BOLD, 13));
lblNewLabel_8.setBounds(357, 64, 145, 14);
contentPane.add(lblNewLabel_8);
```

```
lblNewLabel_9 = new JLabel("POTENCIA CÚBICA");
lblNewLabel_9.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_9.setFont(new Font("Berlin Sans FB Demi", Font.BOLD, 13));
lblNewLabel_9.setBounds(516, 64, 117, 14);
contentPane.add(lblNewLabel_9);
```

```
textR1 = new JTextField();
textR1.setHorizontalAlignment(SwingConstants.CENTER);
textR1.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textR1.setColumns(10);
textR1.setBounds(241, 90, 86, 20);
contentPane.add(textR1);
```

```
textP21 = new JTextField();
textP21.setHorizontalAlignment(SwingConstants.CENTER);
```

```
textP21.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP21.setColumns(10);
textP21.setBounds(388, 90, 86, 20);
contentPane.add(textP21);
```

```
textP31 = new JTextField();
textP31.setHorizontalAlignment(SwingConstants.CENTER);
textP31.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP31.setColumns(10);
textP31.setBounds(526, 90, 86, 20);
contentPane.add(textP31);
```

```
textR2 = new JTextField();
textR2.setHorizontalAlignment(SwingConstants.CENTER);
textR2.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textR2.setColumns(10);
textR2.setBounds(241, 121, 86, 20);
contentPane.add(textR2);
```

```
textP22 = new JTextField();
textP22.setHorizontalAlignment(SwingConstants.CENTER);
textP22.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP22.setColumns(10);
textP22.setBounds(388, 121, 86, 20);
contentPane.add(textP22);
```

```
textP32 = new JTextField();
textP32.setHorizontalAlignment(SwingConstants.CENTER);
textP32.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP32.setColumns(10);
textP32.setBounds(526, 121, 86, 20);
contentPane.add(textP32);
```

```
textR3 = new JTextField();
textR3.setHorizontalAlignment(SwingConstants.CENTER);
textR3.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textR3.setColumns(10);
textR3.setBounds(241, 152, 86, 20);
contentPane.add(textR3);
```

```
textP23 = new JTextField();
textP23.setHorizontalAlignment(SwingConstants.CENTER);
textP23.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP23.setColumns(10);
```

```
textP23.setBounds(388, 152, 86, 20);
contentPane.add(textP23);
```

```
textP33 = new JTextField();
textP33.setHorizontalAlignment(SwingConstants.CENTER);
textP33.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP33.setColumns(10);
textP33.setBounds(526, 152, 86, 20);
contentPane.add(textP33);
```

```
textR4 = new JTextField();
textR4.setHorizontalAlignment(SwingConstants.CENTER);
textR4.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textR4.setColumns(10);
textR4.setBounds(241, 183, 86, 20);
contentPane.add(textR4);
```

```
textP24 = new JTextField();
textP24.setHorizontalAlignment(SwingConstants.CENTER);
textP24.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP24.setColumns(10);
textP24.setBounds(388, 183, 86, 20);
contentPane.add(textP24);
```

```
textP34 = new JTextField();
textP34.setHorizontalAlignment(SwingConstants.CENTER);
textP34.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP34.setColumns(10);
textP34.setBounds(526, 183, 86, 20);
contentPane.add(textP34);
```

```
textR5 = new JTextField();
textR5.setHorizontalAlignment(SwingConstants.CENTER);
textR5.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textR5.setColumns(10);
textR5.setBounds(241, 214, 86, 20);
contentPane.add(textR5);
```

```
textP25 = new JTextField();
textP25.setHorizontalAlignment(SwingConstants.CENTER);
textP25.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP25.setColumns(10);
textP25.setBounds(388, 214, 86, 20);
contentPane.add(textP25);
```



```
textP35 = new JTextField();
textP35.setHorizontalAlignment(SwingConstants.CENTER);
textP35.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP35.setColumns(10);
textP35.setBounds(526, 214, 86, 20);
contentPane.add(textP35);
```

```
textR6 = new JTextField();
textR6.setHorizontalAlignment(SwingConstants.CENTER);
textR6.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textR6.setColumns(10);
textR6.setBounds(241, 244, 86, 20);
contentPane.add(textR6);
```

```
textP26 = new JTextField();
textP26.setHorizontalAlignment(SwingConstants.CENTER);
textP26.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP26.setColumns(10);
textP26.setBounds(388, 244, 86, 20);
contentPane.add(textP26);
```

```
textP36 = new JTextField();
textP36.setHorizontalAlignment(SwingConstants.CENTER);
textP36.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP36.setColumns(10);
textP36.setBounds(526, 244, 86, 20);
contentPane.add(textP36);
```

```
textR7 = new JTextField();
textR7.setHorizontalAlignment(SwingConstants.CENTER);
textR7.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textR7.setColumns(10);
textR7.setBounds(241, 275, 86, 20);
contentPane.add(textR7);
```

```
textP27 = new JTextField();
textP27.setHorizontalAlignment(SwingConstants.CENTER);
textP27.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP27.setColumns(10);
textP27.setBounds(388, 275, 86, 20);
contentPane.add(textP27);
```

```
textP37 = new JTextField();
```

```
textP37.setHorizontalAlignment(SwingConstants.CENTER);
textP37.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 12));
textP37.setColumns(10);
textP37.setBounds(526, 275, 86, 20);
contentPane.add(textP37);
```

```
JButton btnNewButton = new JButton("CALCULAR");
btnNewButton.setBackground(new Color(0, 0, 255));
btnNewButton.setFont(new Font("Berlin Sans FB Demi", Font.BOLD, 14));
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //determinar los valores necesarios
        try {
            double numero;
            if(text1.getText()!="") {

                numero = Double.parseDouble(text1.getText());
                textR1.setText(String.valueOf(MetodosP40.raiz(numero)));
```

```
textP21.setText(String.valueOf(MetodosP40.potencia2(numero)));
```

```
textP31.setText(String.valueOf(MetodosP40.potencia3(numero)));
    }
```

```
    if(text2.getText()!="") {
        //leer lo ingresado por teclado en el número 2
        numero = Double.parseDouble(text2.getText());
        ///mostrar resultados en los campos del número 2
        textR2.setText(String.valueOf(MetodosP40.raiz(numero)));
```

```
textP22.setText(String.valueOf(MetodosP40.potencia2(numero)));
```

```
textP32.setText(String.valueOf(MetodosP40.potencia3(numero)));
    }
```

```
    if(text3.getText()!="") {
        //leer lo ingresado por teclado en el número 3
        numero = Double.parseDouble(text3.getText());
        //mostrar resultados en los campos del número 3
        textR3.setText(String.valueOf(MetodosP40.raiz(numero)));
```

```
textP23.setText(String.valueOf(MetodosP40.potencia2(numero)));
```

```
textP33.setText(String.valueOf(MetodosP40.potencia3(numero)));
```

```

    }

    if(text4.getText()!="") {
        //leer lo ingresado por teclado en el número 4
        numero = Double.parseDouble(text4.getText());
        ///mostrar resultados en los campos del número 4
        textR4.setText(String.valueOf(MetodosP40.raiz(numero)));

textP24.setText(String.valueOf(MetodosP40.potencia2(numero)));

textP34.setText(String.valueOf(MetodosP40.potencia3(numero)));
    }

    if(text5.getText()!="") {
        //leer lo ingresado por teclado en el número 5
        numero = Double.parseDouble(text5.getText());
        //mostrar resultados en los campos del número 5
        textR5.setText(String.valueOf(MetodosP40.raiz(numero)));

textP25.setText(String.valueOf(MetodosP40.potencia2(numero)));

textP35.setText(String.valueOf(MetodosP40.potencia3(numero)));
    }

    if(text6.getText()!="") {
        //leer lo ingresado por teclado en el número 6
        numero = Double.parseDouble(text6.getText());
        //mostrar resultados en los campos del número 6
        textR6.setText(String.valueOf(MetodosP40.raiz(numero)));

textP26.setText(String.valueOf(MetodosP40.potencia2(numero)));

textP36.setText(String.valueOf(MetodosP40.potencia3(numero)));
    }

    if(text7.getText()!="") {
        //leer lo ingresado por teclado en el número 7
        numero = Double.parseDouble(text7.getText());
        //mostrar resultados en los campos del número 7
        textR7.setText(String.valueOf(MetodosP40.raiz(numero)));

textP27.setText(String.valueOf(MetodosP40.potencia2(numero)));

textP37.setText(String.valueOf(MetodosP40.potencia3(numero)));

```

```

    }
    }catch(NumberFormatException ex) {
        //en caso de excepciones, no se pueden hacer las operaciones
        JOptionPane.showMessageDialog(null, "Complete todos los campos
o verifique que los valores ingresados sí sean números.");

    }
    }

});
btnNewButton.setBounds(328, 324, 110, 23);
contentPane.add(btnNewButton);

//volver a escribir los 7 números eliminando lo ingresado con un botón
JButton btnBorrar = new JButton("BORRAR");
btnBorrar.setBackground(new Color(0, 0, 255));
btnBorrar.setFont(new Font("Berlin Sans FB Demi", Font.BOLD, 14));
btnBorrar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //eliminar el contenido de los campos
        textR1.setText("");
        textP21.setText("");
        textP31.setText("");

        textR2.setText("");
        textP22.setText("");
        textP32.setText("");

        textR3.setText("");
        textP23.setText("");
        textP33.setText("");

        textR4.setText("");
        textP24.setText("");
        textP34.setText("");

        textR5.setText("");
        textP25.setText("");
        textP35.setText("");

        textR6.setText("");
        textP26.setText("");
        textP36.setText("");

        textR7.setText("");

```

```

        textP27.setText("");
        textP37.setText("");

        text1.setText("");
        text2.setText("");
        text3.setText("");
        text4.setText("");
        text5.setText("");
        text6.setText("");
        text7.setText("");
    }
});
btnBorrar.setBounds(219, 324, 89, 23);
contentPane.add(btnBorrar);

//el indicativo de cada número
lblNewLabel_1 = new JLabel("2");
lblNewLabel_1.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 13));
lblNewLabel_1.setHorizontalAlignment(SwingConstants.RIGHT);
lblNewLabel_1.setBounds(33, 123, 83, 14);
contentPane.add(lblNewLabel_1);

lblNewLabel_2 = new JLabel("3");
lblNewLabel_2.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 13));
lblNewLabel_2.setHorizontalAlignment(SwingConstants.RIGHT);
lblNewLabel_2.setBounds(10, 154, 104, 14);
contentPane.add(lblNewLabel_2);

lblNewLabel_3 = new JLabel("4");
lblNewLabel_3.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 13));
lblNewLabel_3.setHorizontalAlignment(SwingConstants.RIGHT);
lblNewLabel_3.setBounds(33, 185, 83, 14);
contentPane.add(lblNewLabel_3);

lblNewLabel_4 = new JLabel("5");
lblNewLabel_4.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 13));
lblNewLabel_4.setHorizontalAlignment(SwingConstants.RIGHT);
lblNewLabel_4.setBounds(33, 216, 83, 14);
contentPane.add(lblNewLabel_4);

lblNewLabel_5 = new JLabel("6");
lblNewLabel_5.setFont(new Font("Berlin Sans FB Demi", Font.PLAIN, 13));
lblNewLabel_5.setHorizontalAlignment(SwingConstants.RIGHT);
lblNewLabel_5.setBounds(33, 246, 83, 14);

```

```

contentPane.add(lblNewLabel_5);

//encabezado
lblNewLabel_11 = new JLabel("NÚMERO");
lblNewLabel_11.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_11.setFont(new Font("Berlin Sans FB Demi", Font.BOLD, 13));
lblNewLabel_11.setBounds(126, 65, 86, 14);
contentPane.add(lblNewLabel_11);

//para cerrar la calculadora con un botón
btnCerrar = new JButton("CERRAR");
btnCerrar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(WIDTH); //comando para cerrar el programa
    }
});
btnCerrar.setBackground(new Color(0, 0, 255));
btnCerrar.setFont(new Font("Berlin Sans FB Demi", Font.BOLD, 14));
btnCerrar.setBounds(457, 324, 89, 23);
contentPane.add(btnCerrar);

//para poner estética la interfaz
separator = new JSeparator();
separator.setForeground(new Color(0, 0, 255));
separator.setOrientation(SwingConstants.VERTICAL);
separator.setBounds(53, 64, 2, 283);
contentPane.add(separator);

lblNewLabel_10 = new JLabel("RAÍZ CUADRADA, POTENCIA CUADRADA Y
POTENCIA CÚBICA DE UN GRUPO DE 7 NÚMEROS");
lblNewLabel_10.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_10.setFont(new Font("Berlin Sans FB Demi", Font.BOLD, 13));
lblNewLabel_10.setBounds(53, 21, 580, 14);
contentPane.add(lblNewLabel_10);

    }
}

```

Métodos:

```

package EjercicioPropuesto40;

import java.text.DecimalFormat;

```

```

public class MetodosP40 {
    public static double raiz(double numero){
        double r;
        r = Math.sqrt(numero);
        return r;
    }
    public static double potencia2(double numero){
        double p2;
        p2 = Math.pow(numero,2);
        return p2;
    }
    public static double potencia3(double numero){
        double p3;
        p3 = Math.pow(numero,3);
        return p3;
    }
}

```

Resultado:

	NÚMERO	RAÍZ CUADRADA	POTENCIA CUADRADA	POTENCIA CÚBICA
1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
3	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
4	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
5	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
6	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
7	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

BORRAR CALCULAR CERRAR

8. Ejercicio propuesto N°41:

Método principal:

```
package EjercicioPropuesto41;
```

```
//método principal
```

```

public class EjercicioP41 {
    public static void main(String[] args) {
        InterfazP41 fr= new InterfazP41();
        fr.setVisible(true);
    }
}

```

```
}
```

Interfaz:

```
package EjercicioPropuesto41;
```

```
import java.awt.EventQueue;
```

```
import javax.swing.JFrame;
```

```
import javax.swing.JPanel;
```

```
import javax.swing.border.EmptyBorder;
```

```
import javax.swing.JLabel;
```

```
import javax.swing.JOptionPane;
```

```
import javax.swing.JTextField;
```

```
import javax.swing.JButton;
```

```
import java.awt.event.ActionListener;
```

```
import java.awt.event.ActionEvent;
```

```
import javax.swing.JTextPane;
```

```
import javax.swing.JSeparator;
```

```
import java.awt.Color;
```

```
import java.awt.Font;
```

```
import javax.swing.SwingConstants;
```

```
public class InterfazP41 extends JFrame {
```

```
    private JPanel contentPane;
```

```
    private JTextField textValores;
```

```
    protected double cosa;
```

```
    private JTextField textResultado;
```

```
    /**
```

```
     * Create the frame.
```

```
    */
```

```
    public InterfazP41() {
```

```
        this.setTitle("Ejercicio Propuesto 41"); //para título de la interfaz
```

```
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        setBounds(100, 100, 468, 301); //dimensiones
```

```
        contentPane = new JPanel();
```

```
        contentPane.setBackground(new Color(255, 221, 221));
```

```
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
```

```
        setContentPane(contentPane);
```

```
        contentPane.setLayout(null);
```



```

//ingresar números por teclado
textValores = new JTextField();
textValores.setHorizontalAlignment(SwingConstants.CENTER);
textValores.setFont(new Font("Sylfaen", Font.PLAIN, 13));//setFond hace
referencia a todo lo de las fuentes
textValores.setBounds(21, 90, 415, 20);
contentPane.add(textValores);
textValores.setColumns(10);

//botón para comparar los números
JButton btnComparar = new JButton("Comparar");
btnComparar.setBackground(new Color(255, 255, 255));
btnComparar.setForeground(new Color(255, 0, 0));
btnComparar.setFont(new Font("Sylfaen", Font.BOLD, 13));
btnComparar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            String numeros; //entran como una cadena de caracteres
            //se leen del campo
            numeros = textValores.getText();
            //se imprime el valor
            textResultado.setText("El mayor valor entre los ingresados " +
MetodosP41.mayorN(numeros));
        } catch (NumberFormatException ex) {
            //en caso de excepciones
            JOptionPane.showMessageDialog(null, "Los valores
ingresados no son válidos, intente nuevamente. Recuerde escribir los números de la siguiente manera:
1,2,3,4");
        }
    }
});

btnComparar.setBounds(163, 121, 110, 23);
contentPane.add(btnComparar);

//para eliminar lo ingresado en caso de ser necesario
JButton btnBorrar = new JButton("Borrar");
btnBorrar.setBackground(new Color(255, 255, 255));
btnBorrar.setForeground(new Color(255, 0, 0));
btnBorrar.setFont(new Font("Sylfaen", Font.BOLD, 13));
btnBorrar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //dejar los campos completados en blanco

```

```

        textValores.setText("");
        textResultado.setText("");
    }
});
btnBorrar.setBounds(58, 215, 110, 23);
contentPane.add(btnBorrar);

JTextPane txtpntexto = new JTextPane();
txtpntexto.setForeground(new Color(255, 255, 255));
txtpntexto.setFont(new Font("Sylfaen", Font.PLAIN, 13));
txtpntexto.setBackground(new Color(238, 83, 83));
txtpntexto.setText("Ingrese los números (separados por comas):");
txtpntexto.setBounds(21, 58, 415, 20);
contentPane.add(txtpntexto);

//para indicar que sigue el resultado
textResultado = new JTextField();
textResultado.setHorizontalAlignment(SwingConstants.CENTER);
textResultado.setFont(new Font("Sylfaen", Font.PLAIN, 13));
textResultado.setBounds(21, 184, 415, 20);
contentPane.add(textResultado);
textResultado.setColumns(10);

//titulo de la interfaz
JLabel lblNewLabel = new JLabel("Número mayor");
lblNewLabel.setForeground(new Color(255, 0, 0));
lblNewLabel.setBackground(new Color(255, 0, 0));
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel.setFont(new Font("Sylfaen", Font.BOLD, 15));
lblNewLabel.setBounds(145, 11, 147, 20);
contentPane.add(lblNewLabel);

//separador hecho a mano
JLabel lblNewLabel_1 = new JLabel("~~~~~");
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1.setFont(new Font("Sylfaen", Font.BOLD, 15));
lblNewLabel_1.setBounds(114, 22, 218, 40);
contentPane.add(lblNewLabel_1);

//mostrar resultado
JTextPane txtpnResultado = new JTextPane();
txtpnResultado.setText("Resultado:");
txtpnResultado.setForeground(Color.WHITE);
txtpnResultado.setFont(new Font("Sylfaen", Font.PLAIN, 13));

```

```

txtpnResultado.setBackground(new Color(238, 83, 83));
txtpnResultado.setBounds(21, 155, 415, 20);
contentPane.add(txtpnResultado);

//cerrar la interfaz con un botón
JButton btnSalir = new JButton("Salir");
btnSalir.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(WIDTH); //comando para cerrar el programa
    }
});
btnSalir.setBackground(new Color(255, 255, 255));
btnSalir.setForeground(new Color(255, 0, 0));
btnSalir.setFont(new Font("Sylfaen", Font.BOLD, 13));
btnSalir.setBounds(289, 215, 110, 23);
contentPane.add(btnSalir);
}
}

```

Métodos:

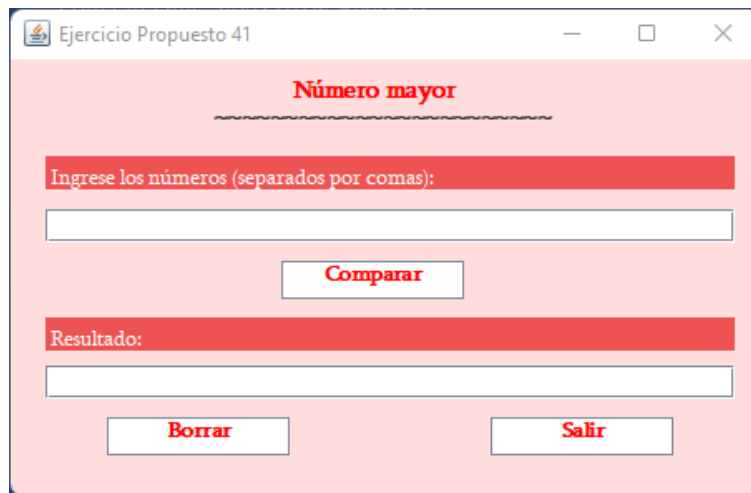
```

package EjercicioPropuesto41;

public class MetodosP41 {
    //método para determinar el número mayor en la cadena de datos
    public static double mayorN(String numeros) {
        double datoArray, cont=0;
        //para separar los números entre sí, ingresados con ,
        String split[]=numeros.split(",");
        //para recorrer el vector. comienza en 0 hasta el tamaño del vector
        //se va comparando con el contador
        for (int i=0; i<split.length; i++) {
            datoArray = Double.parseDouble(split[i]);
            if(datoArray>cont) {
                cont=datoArray;
            }
        }
        return cont;
    }
}

```

Resultado:



Aplicación con interfaz gráfica y diagrama de clases

9. Ejercicio “clases sobre figuras geométricas:

Enlace de acceso a GitHub: <https://github.com/npatinop/Actividad3.git>

Método principal:

```
package FigurasParte2;
```

```
//código principal
```

```
public class FigurasP2 {  
    public static void main(String[] args) {  
        InterfazFiguras fr = new InterfazFiguras();  
        fr.setVisible(true);  
    }  
}
```

Interfaz:

```
import java.awt.EventQueue;
```

```
import javax.swing.JFrame;
```

```
import javax.swing.JPanel;
```

```
import javax.swing.border.EmptyBorder;
```

```
import javax.swing.JSeparator;
```

```
import javax.swing.JLabel;
```

```
import javax.swing.JOptionPane;
import javax.swing.JEditorPane;
import java.awt.Color;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
```

```
public class FigurasInterfaz extends JFrame {
```

```
    private JPanel contentPane;
    private JTextField textRcirculo;
    private JTextField textAcirculo;
    private JTextField textPcirculo;
    private JTextField textBrectangulo;
    private JTextField textHrectangulo;
    private JTextField textArectangulo;
    private JTextField textPrectangulo;
    private JTextField textLcuadrado;
    private JTextField textAcuadrado;
    private JTextField textPcuadrado;
    private JTextField textHtriangulo;
    private JTextField textBtriangulo;
    private JTextField textHiptriangulo;
    private JTextField textPtriangulo;
    private JTextField textAtriangulo;
    private JTextField textDMenor_rombo;
    private JTextField textDMayor_rombo;
    private JTextField textArombo;
    private JTextField textPrombo;
    private JTextField textBMayor_trapezio;
    private JTextField textBMenor_trapezio;
    private JTextField textHtrapezio;
    private JTextField textAtrapezio;
    private JTextField textPtrapezio;
    private JTextField textTipotriangulo;
    private JTextField textLrombo;
```

```
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
```

```

        public void run() {

            try {
                FigurasInterfaz frame = new FigurasInterfaz();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

public FigurasInterfaz() {
    this.setTitle("Figuras geométricas");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 989, 957);
    contentPane = new JPanel();
    contentPane.setBackground(new Color(204, 204, 204));
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

    setContentPane(contentPane);
    contentPane.setLayout(null);

    JPanel contentPane_1 = new JPanel();
    contentPane_1.setLayout(null);
    contentPane_1.setBorder(new EmptyBorder(5, 5, 5, 5));
    contentPane_1.setBackground(new Color(51, 153, 255));
    contentPane_1.setBounds(0, 0, 1258, 918);
    contentPane.add(contentPane_1);

    JLabel lblNewLabel = new JLabel("Círculo");
    lblNewLabel.setFont(new Font("Times New Roman", Font.PLAIN, 16));
    lblNewLabel.setBounds(20, 117, 111, 14);
    contentPane_1.add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("Rectángulo");
    lblNewLabel_1.setFont(new Font("Times New Roman", Font.PLAIN, 16));
    lblNewLabel_1.setBounds(21, 251, 211, 14);
    contentPane_1.add(lblNewLabel_1);

    JLabel lblNewLabel_2 = new JLabel("Cuadrado");
    lblNewLabel_2.setFont(new Font("Times New Roman", Font.PLAIN, 16));
    lblNewLabel_2.setBounds(22, 355, 132, 45);
    contentPane_1.add(lblNewLabel_2);

```

```

JLabel lblNewLabel_2_1 = new JLabel("Triángulo rectángulo");
lblNewLabel_2_1.setFont(new Font("Times New Roman", Font.PLAIN, 16));
lblNewLabel_2_1.setBounds(10, 514, 226, 14);
contentPane_1.add(lblNewLabel_2_1);

JLabel lblNewLabel_2_1_1 = new JLabel("Rombo");
lblNewLabel_2_1_1.setFont(new Font("Times New Roman", Font.PLAIN, 16));
lblNewLabel_2_1_1.setBounds(20, 664, 111, 14);
contentPane_1.add(lblNewLabel_2_1_1);

JLabel lblNewLabel_3 = new JLabel("Trapezio");
lblNewLabel_3.setFont(new Font("Times New Roman", Font.PLAIN, 16));
lblNewLabel_3.setBounds(10, 828, 182, 14);
contentPane_1.add(lblNewLabel_3);

JLabel lblNewLabel_4 = new JLabel("Radio círculo (cm): ");
lblNewLabel_4.setFont(new Font("Times New Roman", Font.PLAIN, 11));
lblNewLabel_4.setBounds(118, 92, 323, 14);
contentPane_1.add(lblNewLabel_4);

textRcirculo = new JTextField();
textRcirculo.setColumns(10);
textRcirculo.setBounds(104, 115, 139, 20);
contentPane_1.add(textRcirculo);

JButton btnCcirculo = new JButton("Determinar");
btnCcirculo.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // para un circulo
        try {
            double radio; // variable
            // ingreso por teclado (captar)
            radio = Double.parseDouble(textRcirculo.getText());
            // definir
            Circulo miCir = new Circulo(radio);
            // salidas
            textAcirculo.setText(String.valueOf(miCir.calcularArea()));

            textPcirculo.setText(String.valueOf(miCir.calcularPerimetro()));
            // en caso de que el radio no se ingrese correctamente
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(null, "El radio ingresado
no es válido. Intente nuevamente.");

```

```
        }  
    }  
});
```

```
btnCcirculo.setBounds(118, 142, 107, 23);  
contentPane_1.add(btnCcirculo);
```

```
JLabel lblNewLabel_5 = new JLabel("Área círculo (cm^2): ");  
lblNewLabel_5.setFont(new Font("Times New Roman", Font.PLAIN, 11));  
lblNewLabel_5.setBounds(369, 92, 123, 14);  
contentPane_1.add(lblNewLabel_5);
```

```
textAcirculo = new JTextField();  
textAcirculo.setColumns(10);  
textAcirculo.setBounds(348, 115, 144, 20);  
contentPane_1.add(textAcirculo);
```

```
JLabel lblNewLabel_5_1 = new JLabel("Perimetro círculo (cm):");  
lblNewLabel_5_1.setFont(new Font("Times New Roman", Font.PLAIN, 11));  
lblNewLabel_5_1.setBounds(522, 92, 166, 14);  
contentPane_1.add(lblNewLabel_5_1);
```

```
textPcirculo = new JTextField();  
textPcirculo.setColumns(10);  
textPcirculo.setBounds(502, 115, 132, 20);  
contentPane_1.add(textPcirculo);
```

```
JLabel lblNewLabel_4_1 = new JLabel("Base rectángulo (cm): ");  
lblNewLabel_4_1.setFont(new Font("Times New Roman", Font.PLAIN, 11));  
lblNewLabel_4_1.setBounds(260, 196, 323, 14);  
contentPane_1.add(lblNewLabel_4_1);
```

```
textBrectangulo = new JTextField();  
textBrectangulo.setColumns(10);  
textBrectangulo.setBounds(260, 221, 139, 20);  
contentPane_1.add(textBrectangulo);
```

```
JLabel lblNewLabel_4_1_1 = new JLabel("Altura rectángulo (cm):");  
lblNewLabel_4_1_1.setFont(new Font("Times New Roman", Font.PLAIN, 11));  
lblNewLabel_4_1_1.setBounds(260, 252, 323, 14);  
contentPane_1.add(lblNewLabel_4_1_1);
```



```

textHrectangulo = new JTextField();
textHrectangulo.setColumns(10);
textHrectangulo.setBounds(260, 277, 139, 20);
contentPane_1.add(textHrectangulo);

JButton btnNewCrectangulo = new JButton("Determinar");
btnNewCrectangulo.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            double Brectangulo, Hrectangulo; // variables
            Brectangulo =
Double.parseDouble(textBrectangulo.getText()); // base
            Hrectangulo = Double.parseDouble(textHrectangulo.getText()); //
altura
            Rectangulo miRect = new Rectangulo(Brectangulo,
Hrectangulo);

textArectangulo.setText(String.valueOf(miRect.calcularArea()));

textPrectangulo.setText(String.valueOf(miRect.calcularPerímetro()));
        }
        // en caso de excepciones
        catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(null,
"Los valores de base y altura no son válidos.
Rectifique e intente nuevamente.");
        }
    }
});
btnNewCrectangulo.setBounds(118, 248, 114, 23);
contentPane_1.add(btnNewCrectangulo);

JLabel lblNewLabel_5_2 = new JLabel("Área rectángulo (cm^2): ");
lblNewLabel_5_2.setFont(new Font("Times New Roman", Font.PLAIN, 11));
lblNewLabel_5_2.setBounds(446, 224, 153, 14);
contentPane_1.add(lblNewLabel_5_2);

textArectangulo = new JTextField();
textArectangulo.setColumns(10);

```

```

textArectangulo.setBounds(437, 249, 144, 20);
contentPane_1.add(textArectangulo);

JLabel lblNewLabel_5_1_1 = new JLabel("Perímetro rectángulo (cm): ");
lblNewLabel_5_1_1.setFont(new Font("Times New Roman", Font.PLAIN, 11));
lblNewLabel_5_1_1.setBounds(616, 224, 166, 14);
contentPane_1.add(lblNewLabel_5_1_1);

textPrectangulo = new JTextField();
textPrectangulo.setColumns(10);
textPrectangulo.setBounds(616, 249, 132, 20);
contentPane_1.add(textPrectangulo);

JLabel lblNewLabel_4_1_2 = new JLabel("Lado cuadrado (cm): ");
lblNewLabel_4_1_2.setFont(new Font("Times New Roman", Font.PLAIN, 11));
lblNewLabel_4_1_2.setBounds(118, 343, 166, 14);
contentPane_1.add(lblNewLabel_4_1_2);

textLcuadrado = new JTextField();
textLcuadrado.setColumns(10);
textLcuadrado.setBounds(104, 368, 139, 20);
contentPane_1.add(textLcuadrado);

JButton btnCcuadrado = new JButton("Determinar");
btnCcuadrado.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            // CALCULA PROPIEDADES CUADRADO
            double lado;
            lado = Double.parseDouble(textLcuadrado.getText());
            Cuadrado miCuad = new Cuadrado(lado);
            // entregar valores encontrados

            textAcuadrado.setText(String.valueOf(miCuad.calcularArea()));

            textPcuadrado.setText(String.valueOf(miCuad.calcularPerímetro()));
        }
        // en caso de haber errores al diligenciar el lado
        catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(null,
                "El valor ingresado para el lado del cuadrado
                NO es válido. Intente de nuevo.");
        }
    }
});

```

```

    }
});
btnCcuadrado.setBounds(118, 402, 114, 23);
contentPane_1.add(btnCcuadrado);

JLabel lblNewLabel_5_2_1 = new JLabel("Área cuadrado (cm^2): ");
lblNewLabel_5_2_1.setFont(new Font("Times New Roman", Font.PLAIN, 11));
lblNewLabel_5_2_1.setBounds(369, 337, 123, 14);
contentPane_1.add(lblNewLabel_5_2_1);

JLabel lblNewLabel_5_1_1_1 = new JLabel("Perímetro cuadrado (cm): ");
lblNewLabel_5_1_1_1.setFont(new Font("Times New Roman", Font.PLAIN, 11));
lblNewLabel_5_1_1_1.setBounds(522, 337, 166, 14);
contentPane_1.add(lblNewLabel_5_1_1_1);

textAcuadrado = new JTextField();
textAcuadrado.setColumns(10);
textAcuadrado.setBounds(358, 368, 132, 20);
contentPane_1.add(textAcuadrado);

textPcuadrado = new JTextField();
textPcuadrado.setColumns(10);
textPcuadrado.setBounds(522, 368, 132, 20);
contentPane_1.add(textPcuadrado);

JLabel lblNewLabel_4_1_2_1 = new JLabel("Altura triángulo (cm): ");
lblNewLabel_4_1_2_1.setFont(new Font("Times New Roman", Font.PLAIN, 11));
lblNewLabel_4_1_2_1.setBounds(276, 530, 123, 14);
contentPane_1.add(lblNewLabel_4_1_2_1);

textHtriangulo = new JTextField();
textHtriangulo.setColumns(10);
textHtriangulo.setBounds(270, 547, 139, 20);
contentPane_1.add(textHtriangulo);

JLabel lblNewLabel_4_1_3 = new JLabel("Base triángulo (cm): ");
lblNewLabel_4_1_3.setFont(new Font("Times New Roman", Font.PLAIN, 11));
lblNewLabel_4_1_3.setBounds(276, 474, 123, 14);
contentPane_1.add(lblNewLabel_4_1_3);

textBtriangulo = new JTextField();
textBtriangulo.setColumns(10);
textBtriangulo.setBounds(270, 499, 139, 20);

```

```

        contentPane_1.add(textBtriangulo);

        JButton btnCtriangulo = new JButton("Determinar");
        btnCtriangulo.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    double Btriangulo, Htriangulo;
                    Btriangulo =
Double.parseDouble(textBtriangulo.getText());
                    Htriangulo =
Double.parseDouble(textHtriangulo.getText());
                    TrianguloRectangulo miTri = new
TrianguloRectangulo(Btriangulo, Htriangulo);

                    textAtriangulo.setText(String.valueOf(miTri.calcularArea()));

                    textPtriangulo.setText(String.valueOf(miTri.calcularPerímetro()));

                    textHiptriangulo.setText(String.valueOf(miTri.calcularH()));

                    if (miTri.tipo() == 27) {
                        textTipotriangulo.setText("El triángulo ingreso es
equilátero.");
                    } else if (miTri.tipo() == 21) {
                        textTipotriangulo.setText("El triángulo es
escaleno.");
                    } else {
                        textTipotriangulo.setText("El triángulo es
isósceles.");
                    }
                }
                // para excepciones donde no se den correctamente los datos
                catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(null,
                        "Los valores indicados no son válidos. Borre
y vuelva a intentarlo.");
                }
            }
        });
        btnCtriangulo.setBounds(143, 511, 117, 23);
        contentPane_1.add(btnCtriangulo);

```

```

JLabel lblNewLabel_5_2_1_1 = new JLabel("Hipotenusa triángulo (cm): ");
lblNewLabel_5_2_1_1.setFont(new Font("Times New Roman", Font.PLAIN, 11));
lblNewLabel_5_2_1_1.setBounds(502, 474, 173, 14);
contentPane_1.add(lblNewLabel_5_2_1_1);

textHiptriangulo = new JTextField();
textHiptriangulo.setColumns(10);
textHiptriangulo.setBounds(502, 499, 132, 20);
contentPane_1.add(textHiptriangulo);

JLabel lblNewLabel_4_1_3_1 = new JLabel("Perímetro triángulo (cm): ");
lblNewLabel_4_1_3_1.setFont(new Font("Times New Roman", Font.PLAIN, 11));
lblNewLabel_4_1_3_1.setBounds(673, 474, 132, 14);
contentPane_1.add(lblNewLabel_4_1_3_1);

textPtriangulo = new JTextField();
textPtriangulo.setColumns(10);
textPtriangulo.setBounds(673, 499, 132, 20);
contentPane_1.add(textPtriangulo);

JLabel lblNewLabel_5_2_1_1_1 = new JLabel("Área triángulo (cm^2): ");
lblNewLabel_5_2_1_1_1.setFont(new Font("Times New Roman", Font.PLAIN,
11));
lblNewLabel_5_2_1_1_1.setBounds(515, 530, 173, 14);
contentPane_1.add(lblNewLabel_5_2_1_1_1);

textAtriangulo = new JTextField();
textAtriangulo.setColumns(10);
textAtriangulo.setBounds(502, 547, 132, 20);
contentPane_1.add(textAtriangulo);

JLabel lblNewLabel_4_1_3_2 = new JLabel("Diámetro menor rombo (cm): ");
lblNewLabel_4_1_3_2.setFont(new Font("Times New Roman", Font.PLAIN, 11));
lblNewLabel_4_1_3_2.setBounds(221, 617, 132, 14);
contentPane_1.add(lblNewLabel_4_1_3_2);

textDMenor_rombo = new JTextField();
textDMenor_rombo.setColumns(10);
textDMenor_rombo.setBounds(221, 633, 139, 20);
contentPane_1.add(textDMenor_rombo);

JLabel lblNewLabel_4_1_3_2_1 = new JLabel("Diámetro mayor rombo (cm): ");

```

```

11));

lblNewLabel_4_1_3_2_1.setFont(new Font("Times New Roman", Font.PLAIN,

lblNewLabel_4_1_3_2_1.setBounds(221, 665, 323, 14);
contentPane_1.add(lblNewLabel_4_1_3_2_1);

textDMayor_rombo = new JTextField();
textDMayor_rombo.setColumns(10);
textDMayor_rombo.setBounds(221, 677, 139, 20);
contentPane_1.add(textDMayor_rombo);

JButton btnCrombo = new JButton("Determinar");
btnCrombo.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            double DMenorombo, DMayor, lado;
            DMenorombo
            Double.parseDouble(textDMenor_rombo.getText());
            DMayor
            Double.parseDouble(textDMayor_rombo.getText());
            lado = Double.parseDouble(textLrombo.getText());
            Rombo miRom = new Rombo(lado, DMenorombo,
            DMayor);

            textArombo.setText(String.valueOf(miRom.calcularArea()));

            textPrombo.setText(String.valueOf(miRom.calcularPerimetro()));
        }
        // para valores no válidos
        catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(null,
            "Uno de los tres valores ingresado no es
            válido. Intente de nuevo.");
        }

    }
});
btnCrombo.setBounds(87, 661, 89, 23);
contentPane_1.add(btnCrombo);

JLabel lblNewLabel_5_2_1_1_1_1 = new JLabel("Área rombo (cm^2): ");
lblNewLabel_5_2_1_1_1_1.setFont(new Font("Times New Roman", Font.PLAIN,
11));

```

```

lblNewLabel_5_2_1_1_1.setBounds(390, 664, 173, 14);
contentPane_1.add(lblNewLabel_5_2_1_1_1);

JLabel lblNewLabel_4_1_3_1_1 = new JLabel("Perímetro rombo (cm): ");
lblNewLabel_4_1_3_1_1.setFont(new Font("Times New Roman", Font.PLAIN,
11));

lblNewLabel_4_1_3_1_1.setBounds(554, 664, 323, 14);
contentPane_1.add(lblNewLabel_4_1_3_1_1);

textArombo = new JTextField();
textArombo.setColumns(10);
textArombo.setBounds(386, 677, 132, 20);
contentPane_1.add(textArombo);

textPrombo = new JTextField();
textPrombo.setColumns(10);
textPrombo.setBounds(543, 677, 132, 20);
contentPane_1.add(textPrombo);

JLabel lblNewLabel_4_1_3_2_2 = new JLabel("Base mayor trapecio (cm): ");
lblNewLabel_4_1_3_2_2.setFont(new Font("Times New Roman", Font.PLAIN,
11));

lblNewLabel_4_1_3_2_2.setBounds(221, 779, 323, 14);
contentPane_1.add(lblNewLabel_4_1_3_2_2);

textBMayor_trapecio = new JTextField();
textBMayor_trapecio.setColumns(10);
textBMayor_trapecio.setBounds(221, 798, 139, 20);
contentPane_1.add(textBMayor_trapecio);

JLabel lblNewLabel_4_1_3_2_2_1 = new JLabel("Base menor trapecio (cm): ");
lblNewLabel_4_1_3_2_2_1.setFont(new Font("Times New Roman", Font.PLAIN,
11));

lblNewLabel_4_1_3_2_2_1.setBounds(221, 829, 323, 14);
contentPane_1.add(lblNewLabel_4_1_3_2_2_1);

textBMenor_trapecio = new JTextField();
textBMenor_trapecio.setColumns(10);
textBMenor_trapecio.setBounds(221, 840, 139, 20);
contentPane_1.add(textBMenor_trapecio);

JLabel lblNewLabel_4_1_3_2_2_1_1 = new JLabel("Altura trapecio (cm): ");
lblNewLabel_4_1_3_2_2_1_1.setFont(new Font("Times New Roman",
Font.PLAIN, 11));

```

```

lblNewLabel_4_1_3_2_2_1_1.setBounds(221, 862, 323, 14);
contentPane_1.add(lblNewLabel_4_1_3_2_2_1_1);

textHtrapecio = new JTextField();
textHtrapecio.setColumns(10);
textHtrapecio.setBounds(221, 887, 139, 20);
contentPane_1.add(textHtrapecio);

JButton btnCtrapecio = new JButton("Determinar");
btnCtrapecio.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try {
            double BMenor =
Double.parseDouble(textBMenor_trapecio.getText()),
            BMayor =
Double.parseDouble(textBMayor_trapecio.getText()),
            altura =
Double.parseDouble(textHtrapecio.getText());
            Trapecio mitrapecio = new Trapecio(BMayor, BMenor,
altura);

            textAtrapecio.setText(String.valueOf(mitrapecio.calcularArea()));

            textPtrapecio.setText(String.valueOf(mitrapecio.calcularPerimetro()));
        }
        // para las excepciones
        catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(null, "Alguno de los
valores no es válido. Revise nuevamente.");
        }

    }
});
btnCtrapecio.setBounds(79, 825, 97, 23);
contentPane_1.add(btnCtrapecio);

JLabel lblNewLabel_5_2_1_1_1_1_1 = new JLabel("Área trapecio (cm^2): ");
lblNewLabel_5_2_1_1_1_1_1.setFont(new Font("Times New Roman",
Font.PLAIN, 11));
lblNewLabel_5_2_1_1_1_1_1.setBounds(401, 810, 173, 14);
contentPane_1.add(lblNewLabel_5_2_1_1_1_1_1);

```



```

textAtrapezio = new JTextField();
textAtrapezio.setColumns(10);
textAtrapezio.setBounds(390, 826, 132, 20);
contentPane_1.add(textAtrapezio);

JLabel lblNewLabel_4_1_3_1_1_1 = new JLabel("Perímetro trapezio (cm): ");
lblNewLabel_4_1_3_1_1_1.setFont(new Font("Times New Roman", Font.PLAIN,
11));

lblNewLabel_4_1_3_1_1_1.setBounds(554, 810, 323, 14);
contentPane_1.add(lblNewLabel_4_1_3_1_1_1);

textPtrapezio = new JTextField();
textPtrapezio.setColumns(10);
textPtrapezio.setBounds(543, 826, 132, 20);
contentPane_1.add(textPtrapezio);

JLabel lblNewLabel_5_2_1_1_1_2 = new JLabel("Tipo triángulo: ");
lblNewLabel_5_2_1_1_1_2.setFont(new Font("Times New Roman", Font.PLAIN,
11));

lblNewLabel_5_2_1_1_1_2.setBounds(673, 530, 173, 14);
contentPane_1.add(lblNewLabel_5_2_1_1_1_2);

textTipotriangulo = new JTextField();
textTipotriangulo.setColumns(10);
textTipotriangulo.setBounds(673, 547, 200, 20);
contentPane_1.add(textTipotriangulo);

JLabel lblNewLabel_4_1_3_2_1_1 = new JLabel("Lado rombo (cm):");
lblNewLabel_4_1_3_2_1_1.setFont(new Font("Times New Roman", Font.PLAIN,
11));

lblNewLabel_4_1_3_2_1_1.setBounds(221, 703, 323, 14);
contentPane_1.add(lblNewLabel_4_1_3_2_1_1);

textLrombo = new JTextField();
textLrombo.setColumns(10);
textLrombo.setBounds(221, 728, 139, 20);
contentPane_1.add(textLrombo);

JButton btnBorrarC = new JButton("Borrar");
btnBorrarC.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textRcirculo.setText("");
        textAcirculo.setText("");
    }
});

```

```

        textPcirculo.setText("");

    }

});
btnBorrarC.setBounds(446, 142, 89, 23);
contentPane_1.add(btnBorrarC);

JButton btnBorrarR = new JButton("Borrar");
btnBorrarR.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textBrectangulo.setText("");
        textHrectangulo.setText("");
        textArectangulo.setText("");
        textPrectangulo.setText("");

    }
});
btnBorrarR.setBounds(554, 277, 89, 23);
contentPane_1.add(btnBorrarR);

JButton btnCuadrado = new JButton("Borrar");
btnCuadrado.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textLcuadrado.setText("");
        textAcuadrado.setText("");
        textPcuadrado.setText("");

    }
});
btnCuadrado.setBounds(463, 402, 89, 23);
contentPane_1.add(btnCuadrado);

JButton btnBorrarrec = new JButton("Borrar");
btnBorrarrec.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // las comillas vacías indican que el campo queda en blanco
        textTipotriangulo.setText("");
        textBtriangulo.setText("");
        textHtriangulo.setText("");
        textHiptriangulo.setText("");
        textAtriangulo.setText("");
        textPtriangulo.setText("");
    }
});

```

```

        }
    });
    btnBorrarrec.setBounds(870, 511, 89, 23);
    contentPane_1.add(btnBorrarrec);

    JButton btnBorrartri = new JButton("Borrar");
    btnBorrartri.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            textDMenor_rombo.setText("");
            textDMayor_rombo.setText("");
            textLrombo.setText("");
            textArombo.setText("");
            textPrombo.setText("");

        }
    });
    btnBorrartri.setBounds(482, 708, 89, 23);
    contentPane_1.add(btnBorrartri);

    JButton btnBorrartra = new JButton("Borrar");
    btnBorrartra.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

            textBMenor_trapecio.setText("");
            textBMayor_trapecio.setText("");
            textHtrapecio.setText("");
            textAtrapecio.setText("");
            textPtrapecio.setText("");

        }
    });
    btnBorrartra.setBounds(482, 854, 89, 23);
    contentPane_1.add(btnBorrartra);

    JEditorPane dtrpnSaludo = new JEditorPane();
    dtrpnSaludo.setText("GeometryNath");
    dtrpnSaludo.setFont(new Font("Times New Roman", Font.BOLD | Font.ITALIC,
25));

    dtrpnSaludo.setBackground(new Color(51, 153, 255));
    dtrpnSaludo.setBounds(357, 11, 277, 39);
    contentPane_1.add(dtrpnSaludo);

    JButton btnSalir = new JButton("Salir\r\n");
    btnSalir.addActionListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent e) {
            System.exit(WIDTH); // comando para cerrar el programa
        }
    });
    btnSalir.setBounds(842, 35, 89, 23);
    contentPane_1.add(btnSalir);

}
}

```

Círculo:

```

public class Circulo {
    double radio; // característica

    public Circulo(double radio) {
        this.radio = radio; // evitar confusiones
    }

    // área y perímetro
    double calcularArea() {
        double areaCir;
        areaCir = Math.PI * Math.pow(radio, 2);
        return areaCir;
    }

    double calcularPerimetro() {
        double perCir;
        perCir = 2 * Math.PI * radio;
        return perCir;
    }
}

```

Cuadrado:

```

public class Cuadrado {
    double lado; // característica

    public Cuadrado(double lado) {
        this.lado = lado; // evitar confusiones
    }
}

```

```

// area y perímetro
double calcularArea() {
    double areaCuad;
    areaCuad = lado * lado;
    return areaCuad;
}

double calcularPerímetro() {
    double perCuad;
    perCuad = 4 * lado;
    return perCuad;
}
}

```

Trapezio:

```

public class Trapecio {
    // características
    double B;
    double b;
    double h;

    // evitar confusiones con las variables porque se incluyen dos veces en la clase
    public Trapecio(double B, double b, double altura) {
        this.B = B;
        this.b = b;
        this.h = altura;
    }

    // area y perímetro
    double calcularArea() {
        double areaTrap;
        areaTrap = ((B + b) / 2.0) * h;
        return areaTrap;
    }

    double calcularPerimetro() {
        double perTrap;
        perTrap = b + B + (2 * Math.sqrt(Math.pow((B - b) / (2.0), 2)) + Math.pow(h, 2));
        return perTrap;
    }
}

```

Rectángulo:

```
public class Rectangulo {  
    // características  
    double b;  
    double h;  
  
    public Rectangulo(double b, double h) {  
        // evitar confusiones  
        this.b = b;  
        this.h = h;  
    }  
  
    // área y perímetro  
    double calcularArea() {  
        double areaRect;  
        areaRect = b * h;  
        return areaRect;  
    }  
  
    double calcularPerímetro() {  
        double perRect;  
        perRect = (2 * b) + (2 * h); // suma de los lados  
        return perRect;  
    }  
}
```

Triángulo y triángulo rectángulo:

```
public class TrianguloRectangulo {  
    // características  
    double b;  
    double h;  
  
    public TrianguloRectangulo(double b, double h) {  
        // evitar confusiones  
        this.b = b;  
        this.h = h;  
    }  
  
    // área  
    double calcularArea() {  
        double areaTR;  
        areaTR = (b * h) / (2.0);  
        return areaTR;  
    }  
}
```

```

// ahora el perímetro
double calcularPerímetro() {
    double perTR;
    perTR = (b + h + calcularH());
    return perTR;
}

// para el perímetro, es necesario hallar la hipotenusa
double calcularH() {
    double H;
    H = Math.pow((b * b) + (h * h), 0.5);
    return H;
}

// saber tipo de triángulo con los resultados obtenidos con condicional
int tipo() {
    int respuesta; // variable
    respuesta = 0; // inicializar
    // si toma el valor de 27, es equilatero
    // si toma el valor de 21, es escaleno
    // si toma el valor de 25, es isósceles
    if ((b == calcularH()) && (b == h) && (h == calcularH())) {
        respuesta = 27;
        return respuesta;
    } else if ((b != h) && (b != calcularH()) && (h != calcularH())) {
        respuesta = 21;
        return respuesta;
    } else {
        respuesta = 25;
        return respuesta;
    }
}
}

```

Resultado:

Figuras geométricas

GeometryNath

Salir

Círculo

Radio círculo (cm):

Área círculo (cm²):

Perímetro círculo (cm):

Determinar

Borrar

Rectángulo

Base rectángulo (cm):

Área rectángulo (cm²):

Perímetro rectángulo (cm):

Altura rectángulo (cm):

Determinar

Borrar

Cuadrado

Lado cuadrado (cm):

Área cuadrado (cm²):

Perímetro cuadrado (cm):

Determinar

Borrar

Triángulo rectángulo

Base triángulo (cm):

Hipotenusa triángulo (cm):

Perímetro triángulo (cm):

Altura triángulo (cm):

Área triángulo (cm²):

Tipo triángulo:

Determinar

Borrar

Rombo

Diámetro menor rombo (cm):

Diámetro mayor rombo (cm):

Área rombo (cm²):

Perímetro rombo (cm):

Lado rombo (cm):

Determinar

Borrar

Trapezio

Base mayor trapezio (cm):

Base menor trapezio (cm):

Área trapezio (cm²):

Perímetro trapezio (cm):

Altura trapezio (cm):

Determinar

Borrar

Diagrama de clases en StarUML:

