

Actividad 3

Programación Orientada a Objetos Grupo 4

Presentado por:
Maria Camila Agudelo Espinosa

Profesor:
Walter Arboleda



**Universidad Nacional de Colombia
Facultad de Minas
Medellín
2023**

Trabajo realizado en el IDE IntelliJ

(La clase Main y otros elementos se pueden encontrar en el repositorio)

Ejercicio 18 Cap 3

Clase Empleado

```
package Datos;

public class Empleado {

    private String codigo_empleado;
    public String nombre;
    private double horas_trabajadas=0, valor_hora;
    public double porcentaje_rtfte;

    public Empleado(String nombre, double porcentaje_rtfte) {
        this.nombre = nombre;
        this.porcentaje_rtfte = porcentaje_rtfte;
    }

    public String getCodigoEmpleado() {
        return this.codigo_empleado;
    }

    public void setCodigoEmpleado(String codigo) {
        // Se podrían agregar condiciones para establecer un código
        // por eso hice esta variable privada
        this.codigo_empleado = codigo;
    }

    public double getHorasTrabajadas() {
        return this.horas_trabajadas;
    }

    public void addHorasTrabajadas(double horas) {
        this.horas_trabajadas += Math.abs(horas);
    }

    public void resetHorasTrabajadas() {
        this.horas_trabajadas = 0;
    }

    public double getValorHora() {
        return this.valor_hora;
    }

    public void setValorHora(double valor) {
        if (valor >= 4833) {
            this.valor_hora = valor;
        }
    }
}
```

Actividad 3. Programación orientada a objetos

```
    else {
        this.valor_hora = 4833;
        System.out.println("El valor mínimo de la hora trabajada en Colombia es 4833");
    }
}

public double salarioBruto() {
    return (this.valor_hora*this.horas_trabajadas);
}

public double rtfte() {
    return (this.salarioBruto()*(this.porcentaje_rtfte/100));
}

public double salarioNeto() {
    return (this.salarioBruto() - this.rtfte());
}
}
```

Interfaz Gráfica

```
package Interfaz;

import Datos.Empleado;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class InterfazEmpleado extends JFrame implements ActionListener {

    private JLabel codigoLabel, nombreLabel, horasLabel, valorLabel, rtfteLabel, salarioBrutoLabel, salarioNetoLabel;
    private JTextField codigoText, nombreText, horasText, valorText, rtfteText, salarioBrutoText, salarioNetoText;
    private JButton calcularButton, limpiarButton;

    public InterfazEmpleado() {
        super("Calculadora de salario neto");

        codigoLabel = new JLabel("Código:");
        nombreLabel = new JLabel("Nombre:");
        horasLabel = new JLabel("Horas trabajadas al mes:");
        valorLabel = new JLabel("Valor hora trabajada:");
        rtfteLabel = new JLabel("Porcentaje de retención en la fuente:");
        salarioBrutoLabel = new JLabel("Salario bruto:");
        salarioNetoLabel = new JLabel("Salario neto:");

        codigoText = new JTextField(10);
        nombreText = new JTextField(10);
        horasText = new JTextField(10);
        valorText = new JTextField(10);
    }
}
```

Actividad 3. Programación orientada a objetos

```
rtfteText = new JTextField(10);
salarioBrutoText = new JTextField(10);
salarioNetoText = new JTextField(10);
salarioBrutoText.setEditable(false);
salarioNetoText.setEditable(false);

calcularButton = new JButton("Calcular");
limpiarButton = new JButton("Limpiar");

calcularButton.addActionListener(this);
limpiarButton.addActionListener(this);

JPanel panel = new JPanel(new GridLayout(8,2));
panel.add(codigoLabel);
panel.add(codigoText);
panel.add(nombreLabel);
panel.add(nombreText);
panel.add(horasLabel);
panel.add(horasText);
panel.add(valorLabel);
panel.add(valorText);
panel.add(rtfteLabel);
panel.add(rtfteText);
panel.add(salarioBrutoLabel);
panel.add(salarioBrutoText);
panel.add(salarioNetoLabel);
panel.add(salarioNetoText);
panel.add(calcularButton);
panel.add(limpiarButton);

setContentPane(panel);
pack();
setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == calcularButton) {
        String codigo = codigoText.getText();
        String nombre = nombreText.getText();
        double horas = Double.parseDouble(horasText.getText());
        double valor = Double.parseDouble(valorText.getText());
        double porcentaje = Double.parseDouble(rtfteText.getText());

        Empleado empleado = new Empleado(nombre, porcentaje);
        empleado.setCodigoEmpleado(codigo);
        empleado.setValorHora(valor);
        empleado.addHorasTrabajadas(horas);

        salarioBrutoText.setText(String.format("%.2f", empleado.salarioBruto()));
        salarioNetoText.setText(String.format("%.2f", empleado.salarioNeto()));
    }
}
```

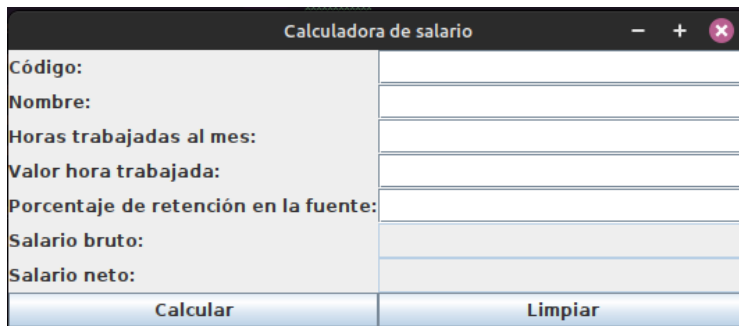
Actividad 3. Programación orientada a objetos

```

else if (e.getSource() == limpiarButton) {
    codigoText.setText("");
    nombreText.setText("");
    horasText.setText("");
    valorText.setText("");
    rtfteText.setText("");
    salarioBrutoText.setText("");
    salarioNetoText.setText("");
}
}

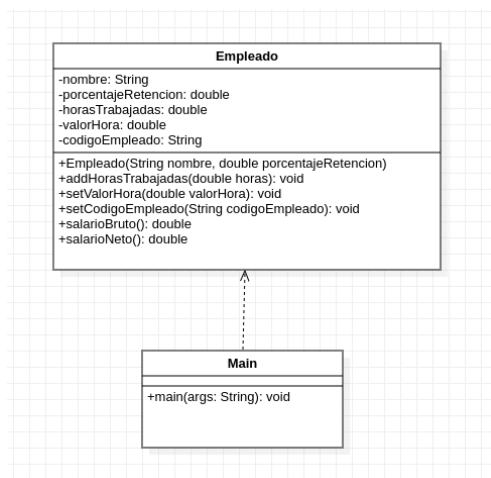
public static void main(String[] args) {
    InterfazEmpleado interfaz = new InterfazEmpleado();
    interfaz.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}

```



Calculadora de salario	
Código:	<input type="text"/>
Nombre:	<input type="text"/>
Horas trabajadas al mes:	<input type="text"/>
Valor hora trabajada:	<input type="text"/>
Porcentaje de retención en la fuente:	<input type="text"/>
Salario bruto:	<input type="text"/>
Salario neto:	<input type="text"/>
Calcular	Limpiar

Diagrama UML



https://github.com/Mila-Ag/Actividad3_POO/tree/master/Ejercicio%2018%20Cap%203

Ejercicio 19 Cap 3

Actividad 3. Programación orientada a objetos

Clase Triángulo

```
package Datos;

import java.util.Scanner;

public class Triangulo {

    public double lado;

    public Triangulo(double lado) {
        this.lado = lado;
    }

    public Triangulo() {
    }

    public double perimetro() {
        return (lado*3);
    }

    public double altura() {
        return Math.sqrt(Math.pow(lado, 2)*Math.pow((lado/2), 2));
    }

    public double area() {
        return (this.lado * this.altura())/2;
    }

}
```

Interfaz Gráfica

```
package Interfaz;

import Datos.Triangulo;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class InterfazTriangulo extends JFrame {

    private JLabel ladoLabel, perimetroLabel, alturaLabel, areaLabel;
    private JTextField ladoTextField;

    public InterfazTriangulo() {
        initComponents();
    }

}
```

Actividad 3. Programación orientada a objetos

```
private void initComponents() {
    ladoLabel = new JLabel("Ingrese el valor del lado:");
    perimetroLabel = new JLabel("Perimetro: ");
    alturaLabel = new JLabel("Altura: ");
    areaLabel = new JLabel("Área: ");
    ladoTextField = new JTextField();

    setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    setTitle("Calculadora de valores en Triangulos Equilateros");
    setResizable(false);

    JPanel panel = new JPanel();
    panel.setLayout(new java.awt.GridLayout(4, 2, 10, 10));
    panel.add(ladoLabel);
    panel.add(ladoTextField);
    panel.add(perimetroLabel);
    panel.add(new JLabel());
    panel.add(alturaLabel);
    panel.add(new JLabel());
    panel.add(areaLabel);
    panel.add(new JLabel());
    panel.setBackground(new Color(143, 188, 143));

    ladoTextField.addKeyListener(new KeyAdapter() {
        public void keyReleased(KeyEvent evt) {
            calcularValores();
        }
    });

    getContentPane().add(panel, java.awt.BorderLayout.CENTER);

    pack();
    setLocationRelativeTo(null);
}

private void calcularValores() {
    double lado = Double.parseDouble(ladoTextField.getText());
    Triangulo triangulo = new Triangulo(lado);
    double perimetro = triangulo.perimetro();
    double altura = triangulo.altura();
    double area = triangulo.area();
    perimetroLabel.setText("Perimetro: " + perimetro);
    alturaLabel.setText("Altura: " + altura);
    areaLabel.setText("Area: " + area);
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new InterfazTriangulo().setVisible(true);
        }
    });
}
```

Actividad 3. Programación orientada a objetos

```
    });  
    }  
}
```

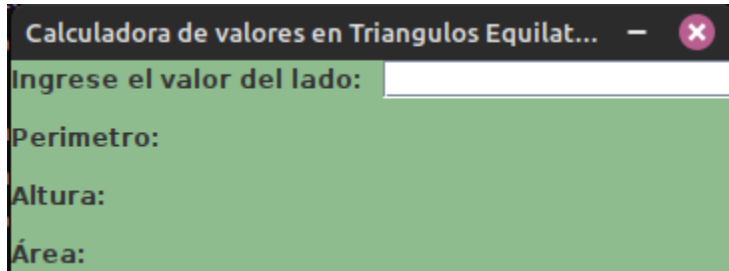
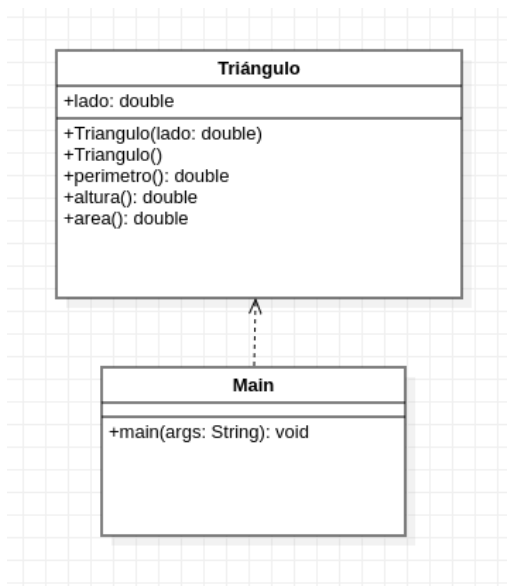


Diagrama UML



https://github.com/Mila-Ag/Actividad3_POO/tree/master/Ejercicio%2019%20Cap%203

Ejercicio 7 Cap 4

Clase NumeroMayor

```
package Clases;  
  
public class NumeroMayor {  
  
    private double a, b;  
  
    public NumeroMayor(double a, double b) {  
        this.a = a;  
    }  
}
```


Actividad 3. Programación orientada a objetos

```
this.b = b;
}

public double getA() {
    return a;
}

public void setA(double a) {
    this.a = a;
}

public double getB() {
    return b;
}

public void setB(double b) {
    this.b = b;
}

public String compararNumeros() {
    String resultado = "";
    if (a > b) {
        resultado = "A es mayor que B";
    } else if (a == b) {
        resultado = "A es igual a B";
    } else {
        resultado = "A es menor que B";
    }
    return resultado;
}
}
```

Interfaz Gráfica

```
package Interfaz;

import Clases.NumeroMayor;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class InterfazNumeroMayor extends JFrame {

    private JTextField textFieldA;
```

Actividad 3. Programación orientada a objetos

```
private JTextField textFieldB;
private JLabel labelResultado;
private JButton buttonComparar;

public InterfazNumeroMayor() {
    super("Comparador de números");
    initComponents();
}

private void initComponents() {
    JPanel panel = new JPanel();
    panel.setBackground(new Color(220, 200, 255)); // Color morado claro

    JLabel labelA = new JLabel("Valor de A:");
    labelA.setPreferredSize(new Dimension(80, 25));
    panel.add(labelA);

    textFieldA = new JTextField();
    textFieldA.setPreferredSize(new Dimension(150, 25));
    panel.add(textFieldA);

    JLabel labelB = new JLabel("Valor de B:");
    labelB.setPreferredSize(new Dimension(80, 25));
    panel.add(labelB);

    textFieldB = new JTextField();
    textFieldB.setPreferredSize(new Dimension(150, 25));
    panel.add(textFieldB);

    buttonComparar = new JButton("Comparar");
    panel.add(buttonComparar);

    labelResultado = new JLabel();
    labelResultado.setPreferredSize(new Dimension(200, 25));
    panel.add(labelResultado);

    buttonComparar.addActionListener(e -> {
        double a = Double.parseDouble(textFieldA.getText());
        double b = Double.parseDouble(textFieldB.getText());
        NumeroMayor nm = new NumeroMayor(a, b);
        labelResultado.setText(nm.compararNumeros());
    });

    setContentPane(panel);
    setSize(300, 150);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
    setResizable(false);
    setVisible(true);
}
```

Actividad 3. Programación orientada a objetos

```

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

```

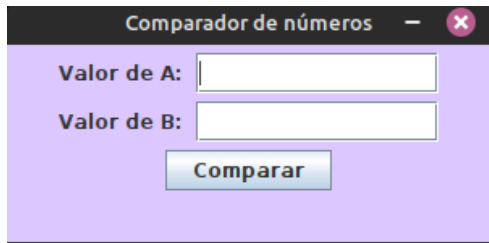
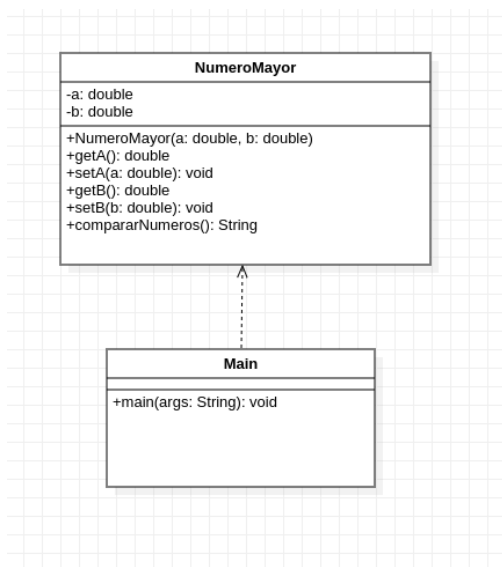


Diagrama UML



https://github.com/Mila-Ag/Actividad3_POO/tree/master/Ejercicio%207%20Cap%204

Ejercicio 10 Cap 4

Clase Estudiante

```

package Datos;
public class Estudiante {

    public String nl, nom;
    public double pat;
    public int estrato;

    public Estudiante(String nl, String nom, double pat, int estrato) {
        this.nl = nl;
        this.nom = nom;
    }
}

```

Actividad 3. Programación orientada a objetos

```
this.pat = pat;
this.estrato = estrato;
}

public double matricula() {
double pagmat = 50000;
if ((pat > 2000000) && (estrato > 3)) {
pagmat = pagmat + (0.03* pat);
}
return pagmat;
}
}
```

Interfaz Gráfica

```
package Interfaz;

import Datos.Estudiante;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class InterfazEstudiante extends JFrame {

    private JTextField nIField, nomField, patField, estratoField;
    private JLabel nILabel, nomLabel, patLabel, estratoLabel, matLabel;

    public InterfazEstudiante() {
        setTitle("Calculadora de matricula");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(500, 400);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(5, 2));
        panel.setBackground(new Color(163, 228, 215));

        nILabel = new JLabel("Numero de Inscripcion: ");
        nIField = new JTextField(4);

        nomLabel = new JLabel("Nombre: ");
        nomField = new JTextField(4);

        patLabel = new JLabel("Patrimonio: ");
        patField = new JTextField(4);
```

Actividad 3. Programación orientada a objetos

```
estratoLabel = new JLabel("Estrato social: ");
estratoField = new JTextField(4);

matLabel = new JLabel("");

JButton calcularButton = new JButton("Calcular");
calcularButton.setBackground(new Color(43, 96, 70));
calcularButton.setForeground(Color.WHITE);
calcularButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String nl = nlField.getText();
        String nom = nomField.getText();
        double pat = Double.parseDouble(patField.getText());
        int estrato = Integer.parseInt(estratoField.getText());

        Estudiante estudiante = new Estudiante(nl, nom, pat, estrato);
        double matricula = estudiante.matricula();

        matLabel.setText("Matricula: $" + String.valueOf(matricula));
    }
});

JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
buttonPanel.setBackground(new Color(163, 228, 215));
buttonPanel.add(calcularButton);

panel.add(nlLabel);
panel.add(nlField);
panel.add(nomLabel);
panel.add(nomField);
panel.add(patLabel);
panel.add(patField);
panel.add(estratoLabel);
panel.add(estratoField);
panel.add(buttonPanel);
panel.add(matLabel);

add(panel);
setLocationRelativeTo(null);
setVisible(true);
}

public static void main(String[] args) {
    InterfazEstudiante window = new InterfazEstudiante();
}
}
```

Actividad 3. Programación orientada a objetos

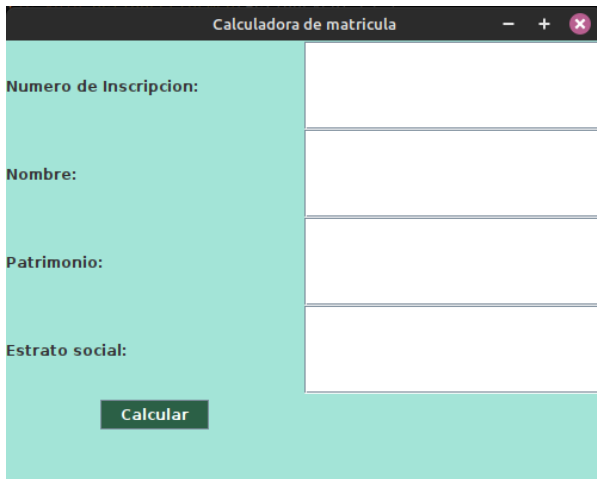
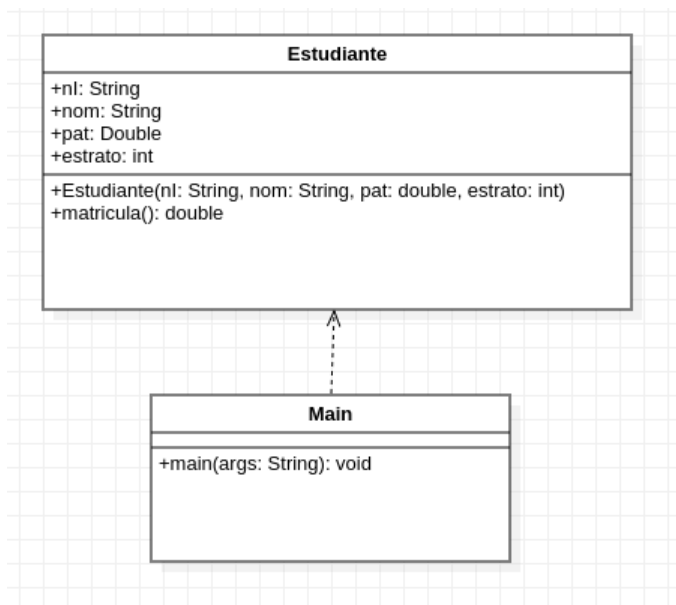


Diagrama UML



https://github.com/Mila-Ag/Actividad3_POO/tree/master/Ejercicio%2010%20Cap%204

Ejercicio 22 Cap 4

Clase Empleado

```
package Datos;
```

```
public class Empleado {
```

```
    public String nombre;
    public double valorHora, numHoras;
```

```
    public Empleado(String nombre, double valorHora, double numHoras) {
```

Actividad 3. Programación orientada a objetos

```
        this.nombre = nombre;
        this.valorHora = valorHora;
        this.numHoras = numHoras;
    }

    public double salario() {
        return valorHora * numHoras;
    }
}
```

Interfaz Gráfica

```
package Interfaz;

import Datos.Empleado;

import javax.swing.*;

import javax.swing.*;
import java.awt.*;

import javax.swing.*;
import java.awt.*;

public class InterfazEmpleado extends JFrame {

    private JTextField txtNombre;
    private JTextField txtValorHora;
    private JTextField txtNumHoras;
    private JTextArea txtResultado;

    public InterfazEmpleado() {
        setTitle("Calculadora de salario");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel lblNombre = new JLabel("Nombre:");
        JLabel lblValorHora = new JLabel("Salario por hora:");
        JLabel lblNumHoras = new JLabel("Número de horas trabajadas:");
        txtNombre = new JTextField(20);
        txtValorHora = new JTextField(10);
        txtNumHoras = new JTextField(10);
        JButton btnCalcular = new JButton("Calcular");
        btnCalcular.setBackground(new Color(70, 30, 130));
        btnCalcular.setForeground(Color.WHITE);
        txtResultado = new JTextArea(10, 30);
        JScrollPane scrollPane = new JScrollPane(txtResultado);
        JPanel panel = new JPanel();
        panel.setBackground(new Color(177, 150, 246));
        panel.add(lblNombre);
```

Actividad 3. Programación orientada a objetos

```

panel.add(txtNombre);
panel.add(lblValorHora);
panel.add(txtValorHora);
panel.add(lblNumHoras);
panel.add(txtNumHoras);
panel.add(btnCalcular);
panel.add(scrollPane);
add(panel);

btnCalcular.addActionListener(e -> calcularSalario());
}

private void calcularSalario() {
String nombre = txtNombre.getText();
double valorHora = Double.parseDouble(txtValorHora.getText());
double numHoras = Double.parseDouble(txtNumHoras.getText());

Empleado empleado = new Empleado(nombre, valorHora, numHoras);
double salario = empleado.salario();

if (salario > 450000) {
txtResultado.setText(nombre + " gana un salario mensual de $" + salario);
} else {
txtResultado.setText(nombre);
}
}
}

```

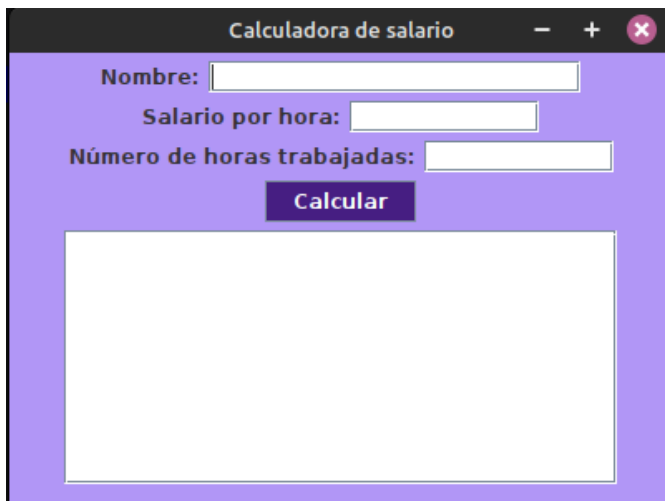
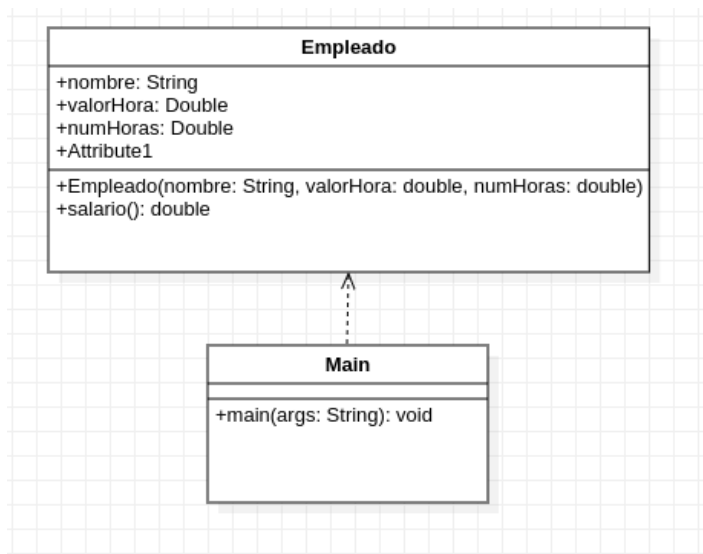


Diagrama UML

Actividad 3. Programación orientada a objetos



https://github.com/Mila-Ag/Actividad3_POO/tree/master/Ejercicio%2022%20Cap%204

Ejercicio 23 Cap 4

Clase Soluciones

```
package Datos;

import javax.swing.JOptionPane;

public class Soluciones {

    public double a, b, c;

    public Soluciones(double a, double b, double c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }

    public double[] resultado() {
        double discriminante = Math.pow(b, 2)-(4*a*c);
        if (discriminante < 0) {
            JOptionPane.showMessageDialog(null, "Esta ecuación no tiene solución en los reales");
            double[] array = {0, 0};
            return array;
        }
        else {
```

Actividad 3. Programación orientada a objetos

```
        double[]      array      =      {(-(this.b)+Math.sqrt(discriminante))/(2*this.a),  
(-(this.b)-Math.sqrt(discriminante))/(2*this.a)};  
        return array;  
    }  
}
```

Interfaz Gráfica

```
package Interfaz;  
  
import Datos.Soluciones;  
  
import javax.swing.*.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
import java.awt.Color;  
import java.math.RoundingMode;  
import java.text.DecimalFormat;  
  
public class InterfazSolucion extends JFrame {  
  
    private JLabel labelA, labelB, labelC, labelX1, labelX2;  
    private JTextField campoA, campoB, campoC, campoX1, campoX2;  
    private JButton botonCalcular;  
  
    public InterfazSolucion() {  
        initComponents();  
        getContentPane().setBackground(new Color(255, 255, 204));  
        setLayout(null);  
        setSize(400, 300);  
        setLocationRelativeTo(null);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setVisible(true);  
    }  
  
    private void initComponents() {  
        labelA = new JLabel("Valor de a:");  
        labelA.setBounds(50, 30, 100, 30);  
        add(labelA);  
  
        campoA = new JTextField();  
        campoA.setBounds(150, 30, 100, 30);  
        add(campoA);  

```

Actividad 3. Programación orientada a objetos

```
labelB = new JLabel("Valor de b:");
labelB.setBounds(50, 70, 100, 30);
add(labelB);

campoB = new JTextField();
campoB.setBounds(150, 70, 100, 30);
add(campoB);

labelC = new JLabel("Valor de c:");
labelC.setBounds(50, 110, 100, 30);
add(labelC);

campoC = new JTextField();
campoC.setBounds(150, 110, 100, 30);
add(campoC);

botonCalcular = new JButton("Calcular");
botonCalcular.setBounds(150, 150, 100, 30);
botonCalcular.setBackground(new Color(255, 204, 0));
add(botonCalcular);

labelX1 = new JLabel("x1 = ");
labelX1.setBounds(50, 200, 50, 30);
add(labelX1);

campoX1 = new JTextField();
campoX1.setBounds(80, 200, 100, 30);
campoX1.setEditable(false);
add(campoX1);

labelX2 = new JLabel("x2 = ");
labelX2.setBounds(200, 200, 50, 30);
add(labelX2);

campoX2 = new JTextField();
campoX2.setBounds(230, 200, 100, 30);
campoX2.setEditable(false);
add(campoX2);

botonCalcular.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            double a = Double.parseDouble(campoA.getText());
            double b = Double.parseDouble(campoB.getText());
```

Actividad 3. Programación orientada a objetos

```
double c = Double.parseDouble(campoC.getText());

Soluciones ecuacion = new Soluciones(a, b, c);
double[] soluciones = ecuacion.resultado();

if (soluciones.length == 2) {
    DecimalFormat df = new DecimalFormat("#.#####");
    df.setRoundingMode(RoundingMode.CEILING);
    campoX1.setText(df.format(soluciones[0]));
    campoX2.setText(df.format(soluciones[1]));
} else {
    campoX1.setText("");
    campoX2.setText("");
    JOptionPane.showMessageDialog(null, "Esta ecuación no tiene solución en los reales");
}
} catch (NumberFormatException ex) {
    campoX1.setText("");
    campoX2.setText("");
    JOptionPane.showMessageDialog(null, "Error: Uno o más campos contienen valores inválidos");
}
}
});
}

public static void main(String[] args) {
    InterfazSolucion solucion = new InterfazSolucion();
}
}
```

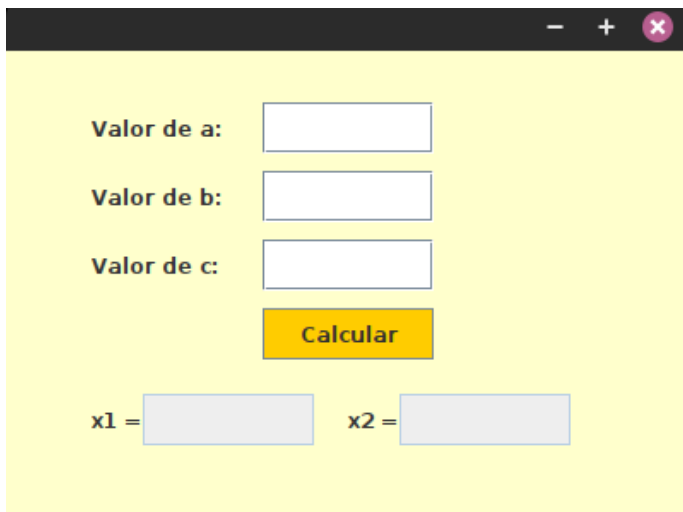
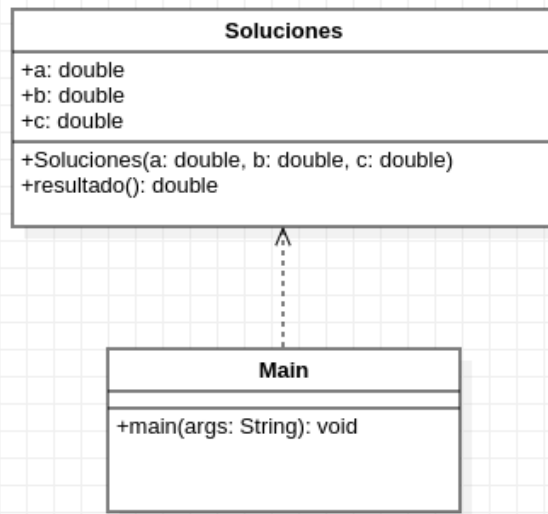


Diagrama UML

Actividad 3. Programación orientada a objetos



https://github.com/Mila-Ag/Actividad3_POO/tree/master/Ejercicio%2023%20Cap%204

Ejercicio 40 Cap 5

Clase OperacionesAritmeticas

```
package Datos;
```

```
public class OperacionesAritmeticas {

    public double[] valores;

    public static double redondear(double valor, int lugares) {
        if (lugares < 0) throw new IllegalArgumentException();

        long factor = (long) Math.pow(10, lugares);
        valor = valor * factor;
        long tmp = Math.round(valor);
        return (double) tmp / factor;
    }

    public OperacionesAritmeticas(double[] valores) {
        this.valores = valores;
    }

    public double[] raizCuadrada() {
        double[] ans = new double[this.valores.length];
        for (int i=0; i<this.valores.length; i++) {
            ans[i] = Math.sqrt(this.valores[i]);
        }
    }
}
```

Actividad 3. Programación orientada a objetos

```
}
return ans;
}

public double[] cuadrado() {
    double[] ans = new double[this.valores.length];
    for (int i=0; i<this.valores.length; i++) {
        ans[i] = Math.pow(this.valores[i], 2);
    }
    return ans;
}

public double[] cubo() {
    double[] ans = new double[this.valores.length];
    for (int i=0; i<this.valores.length; i++) {
        ans[i] = Math.pow(this.valores[i], 3);
    }
    return ans;
}

public String obtenerOperaciones() {
    double[] raices_cuadradas = this.raizCuadrada();
    double[] cuadrados = this.cuadrado();
    double[] cubos = this.cubo();

    String ans = "";
    for (int i=0; i<this.valores.length; i++) {
        ans += "Para: "+String.valueOf(redondear(this.valores[i], 2))+"\n";
        ans += "-> Su raíz cuadrada es: "+String.valueOf(redondear(raices_cuadradas[i], 2))+"\n-> Su cuadrado es: = "+String.valueOf(redondear(cuadrados[i], 2))+"\n-> Su cubo es: "+String.valueOf(redondear(cubos[i], 2))+"\n\n";
    }

    return ans;
}

}
```

Interfaz Gráfica

```
package Interfaz;

import Datos.OperacionesAritmeticas;
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.Color;

public class InterfazOperaciones extends JFrame implements ActionListener {

    private JTextField entradaNumeros;
    private JTextArea resultados;
```

Actividad 3. Programación orientada a objetos

```
public InterfazOperaciones() {

    setTitle("Operaciones Aritméticas");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(400, 300);
    setLocationRelativeTo(null);

    JLabel etiquetaNumeros = new JLabel("Introduzca los números (separados por coma):");
    entradaNumeros = new JTextField(20);
    JButton botonCalcular = new JButton("Calcular");
    botonCalcular.addActionListener(this);
    botonCalcular.setBackground(new Color(102, 51, 0));
    botonCalcular.setForeground(Color.WHITE);
    resultados = new JTextArea(10, 20);
    resultados.setEditable(false);

    // Añadir los componentes a la ventana
    JPanel panel = new JPanel();
    panel.setBackground(new Color(204, 153, 102));
    panel.add(etiquetaNumeros);
    panel.add(entradaNumeros);
    panel.add(botonCalcular);
    panel.add(new JScrollPane(resultados));
    add(panel);

    setVisible(true);
}

public void actionPerformed(ActionEvent e) {

    String entrada = entradaNumeros.getText();
    String[] numerosString = entrada.split(",");
    double[] numeros = new double[numerosString.length];
    for (int i=0; i<numerosString.length; i++) {
        numeros[i] = Double.parseDouble(numerosString[i]);
    }

    OperacionesAritmeticas operaciones = new OperacionesAritmeticas(numeros);
    String resultadoString = operaciones.obtenerOperaciones();
    resultados.setText(resultadoString);
}

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

}
```

Actividad 3. Programación orientada a objetos

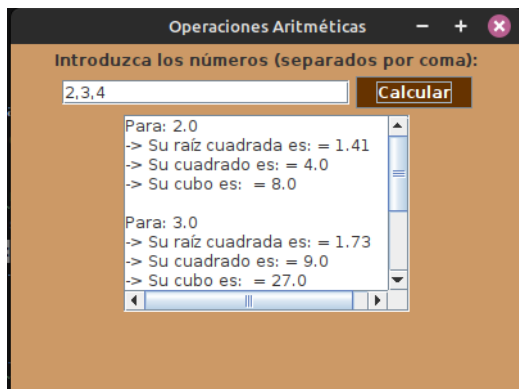
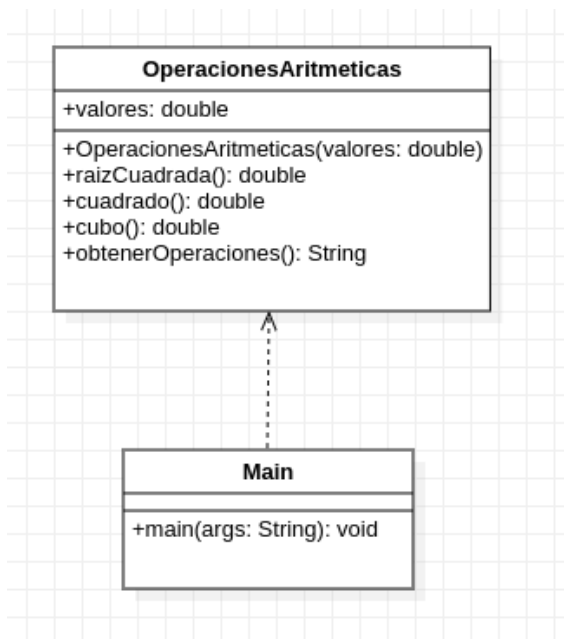


Diagrama UML



https://github.com/Mila-Ag/Actividad3_POO/tree/master/Ejercicio%2040%20Cap%205

Ejercicio 41 Cap 5

Clase Numeros

```
package clases;
```

```
import java.util.Arrays;
```

```
public class Numeros {
    public double[] valores;

    public Numeros(double[] valores) {
        this.valores = valores;
    }
}
```


Actividad 3. Programación orientada a objetos

```
    }

    public void ordenarSerie() {
        for (int i = 0; i < this.valores.length; i++) {
            for (int j = 0; j < this.valores.length - i - 1; j++) {
                if (this.valores[j] > this.valores[j + 1]) {
                    double aux = this.valores[j];
                    this.valores[j] = this.valores[j + 1];
                    this.valores[j + 1] = aux;
                }
            }
        }
    }

    public double[] obtenerSerieOrdenada() {
        double[] ans = Arrays.copyOf(this.valores, this.valores.length);

        for (int i = 0; i < (ans.length - 1); i++) {
            for (int j = 0; j < ans.length - i - 1; j++) {
                if (ans[j] > ans[j + 1]) {
                    double aux = ans[j];
                    ans[j] = ans[j + 1];
                    ans[j + 1] = aux;
                }
            }
        }

        return ans;
    }

    public String obtenerCsv() {
        String ans = "";

        for (int i = 0; i < this.valores.length; i++) {
            ans += String.valueOf(this.valores[i]) + ",";
        }
        ans = ans.substring(0, ans.length() - 1);

        return ans;
    }
}
```

Interfaz Gráfica

```
package Interfaz;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class InterfazNumeros extends JFrame implements ActionListener {
```

Actividad 3. Programación orientada a objetos

```
private final JTextField inputField;
private final JButton calcularButton;
private final JLabel resultadoLabel;

public InterfazNumeros() {

    setTitle("Encuentra el número mayor");
    setSize(400, 200);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JPanel inputPanel = new JPanel(new GridLayout(0, 1));
    JLabel inputLabel = new JLabel("Ingrese hasta 10 números separados por comas:");
    inputPanel.add(inputLabel);

    inputField = new JTextField();
    inputPanel.add(inputField);

    calcularButton = new JButton("Calcular");
    calcularButton.addActionListener(this);
    calcularButton.setBackground(Color.BLUE.darker());
    calcularButton.setForeground(Color.WHITE);
    inputPanel.add(calcularButton);

    JPanel resultadoPanel = new JPanel(new FlowLayout());
    resultadoPanel.setBackground(new Color(125, 187, 246)); // Azul claro
    resultadoLabel = new JLabel("");
    resultadoPanel.add(resultadoLabel);

    setLayout(new BorderLayout());
    add(inputPanel, BorderLayout.CENTER);
    add(resultadoPanel, BorderLayout.SOUTH);
    setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == calcularButton) {

        String inputString = inputField.getText();
        String[] inputArray = inputString.split(",");

        double[] numeros = new double[inputArray.length];
        for (int i = 0; i < inputArray.length; i++) {
            try {
                numeros[i] = Double.parseDouble(inputArray[i]);
            }
        }
    }
}
```

Actividad 3. Programación orientada a objetos

```

    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(this, "Ingrese solo números válidos.");
        return;
    }
}

```

```

clases.Numeros serieNumeros = new clases.Numeros(numeros);
serieNumeros.ordenarSerie();
double mayor = serieNumeros.valores[numeros.length - 1];

```

```

resultadoLabel.setText("El mayor valor es: " + mayor);
}
}

```

```

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

```

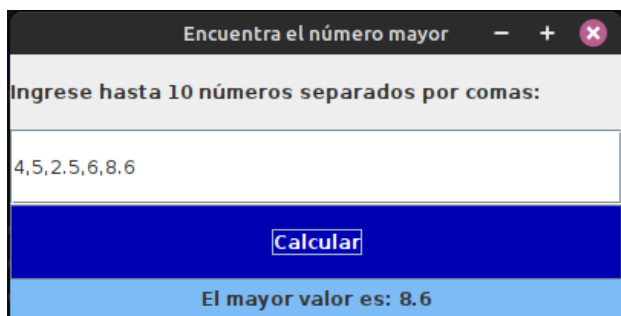
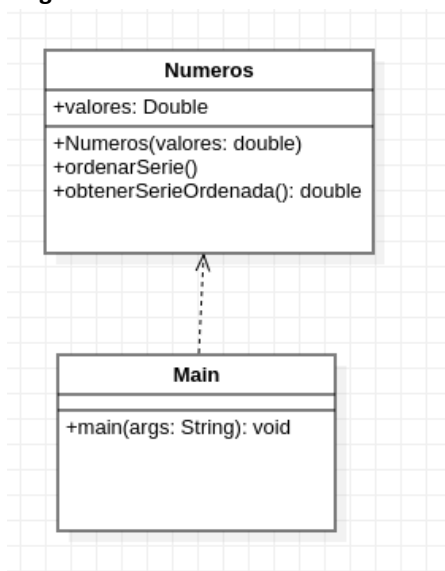


Diagrama UML



Actividad 3. Programación orientada a objetos

https://github.com/Mila-Ag/Actividad3_POO/tree/master/Ejercicio%2041%20Cap%205

Figuras geométricas

Clase circulo

```
package Datos;
public class Circulo implements FiguraBase {

    public double radio;

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

    public double area() {
        return (Math.PI * Math.pow(this.radio, 2));
    }

    public double perimetro() {
        return (2 * Math.PI * this.radio);
    }

}
```

Clase Cuadrado

```
package Datos;

public class Cuadrado implements FiguraBase {

    public double lado;

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

    public double area() {
        return Math.pow(this.lado, 2);
    }

    public double perimetro() {
        return (4 * this.lado);
    }

}
```

Clase rectángulo

Actividad 3. Programación orientada a objetos

```
package Datos;

public class Rectangulo implements FiguraBase {

    public double base, altura;

    public Rectangulo(double base, double altura) {
        this.base = base;
        this.altura = altura;
    }

    public double area() {
        return (base * altura);
    }

    public double perimetro() {
        return ((2*base) + (2*altura));
    }

}
```

Clase Triángulo

```
package Datos;

public class Triangulo implements FiguraBase {

    public double base, altura;

    public Triangulo(double base, double altura) {
        this.base = base;
        this.altura = altura;
    }

    public double area() {
        return this.base * this.altura;
    }

    public double hipotenusa() {
        return Math.sqrt(Math.pow(this.base, 2) + Math.pow(this.altura, 2));
    }

    public double perimetro() {
        return (this.base + this.altura + this.hipotenusa());
    }

    public String tipo() {
        if ((this.base == this.altura) || (this.base == 0) || (this.altura == 0)) {
            return "Isóceles";
        }
        else {

```

Actividad 3. Programación orientada a objetos

```
        return "Escaleno";  
    }  
}
```

Clase FiguraBase

```
package Datos;  
  
public interface FiguraBase {  
  
    public double area();  
    public double perimetro();  
  
}
```

Interfaz Gráfica

```
package Interfaz;  
  
import Datos.*;  
  
import javax.swing.*;  
import java.awt.*;  
  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
import java.awt.Color;  
import java.awt.GridLayout;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.Objects;  
  
import javax.swing.JButton;  
import javax.swing.JComboBox;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JTextField;  
  
public class InterfazFiguras extends JFrame implements ActionListener {  
  
    private JComboBox<String> opcionesFiguras;  
    private JLabel label1, label2, label3, label4, resultado1, resultado2;  
    private JTextField textField1, textField2;  
    private JButton calcularButton;  
  
    public InterfazFiguras() {
```

Actividad 3. Programación orientada a objetos

```
super("Calcular datos de figuras geométricas");
setSize(500, 300);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JPanel panel = new JPanel(new GridLayout(6, 2));
panel.setBackground(new Color(177, 150, 246)); // Verde claro

opcionesFiguras = new JComboBox<String>(new String[]{"Círculo", "Cuadrado", "Rectángulo", "Triángulo"});
opcionesFiguras.addActionListener(this);
panel.add(new JLabel("Seleccione una figura geométrica:"));
panel.add(opcionesFiguras);

label1 = new JLabel("Ingrese el valor del primer parámetro:");
panel.add(label1);

textField1 = new JTextField(10);
panel.add(textField1);

label2 = new JLabel("Ingrese el valor del segundo parámetro:");
panel.add(label2);

textField2 = new JTextField(10);
panel.add(textField2);

label3 = new JLabel("Área:");
panel.add(label3);

resultado1 = new JLabel("");
panel.add(resultado1);

label4 = new JLabel("Perímetro:");
panel.add(label4);

resultado2 = new JLabel("");
panel.add(resultado2);

calcularButton = new JButton("Calcular");
calcularButton.addActionListener(this);
calcularButton.setBackground(new Color(102, 0, 102)); // Morado oscuro
calcularButton.setForeground(Color.WHITE); // Letras blancas
panel.add(calcularButton);

add(panel);
setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == opcionesFiguras) {
        String figura = (String) opcionesFiguras.getSelectedItem();
        switch (figura) {
```

Actividad 3. Programación orientada a objetos

```
case "Círculo":
    label1.setText("Ingrese el valor del radio:");
    label2.setVisible(false);
    textField2.setVisible(false);
    break;
case "Cuadrado":
    label1.setText("Ingrese el valor del lado:");
    label2.setVisible(false);
    textField2.setVisible(false);
    break;
case "Rectángulo":
    label1.setText("Ingrese el valor de la base:");
    label2.setText("Ingrese el valor de la altura:");
    label2.setVisible(true);
    textField2.setVisible(true);
    break;
case "Triángulo":
    label1.setText("Ingrese el valor de la base:");
    label2.setText("Ingrese el valor de la altura:");
    label2.setVisible(true);
    textField2.setVisible(true);
    break;
}
resultado1.setText("");
resultado2.setText("");
} else if (e.getSource() == calcularButton) {
    String figura = (String) opcionesFiguras.getSelectedItem();
    try {
        double valor1 = Double.parseDouble(textField1.getText());
        double valor2 = Double.parseDouble(textField2.getText());
        FiguraBase figuraGeometrica;
        switch (figura) {
            case "Círculo":
                figuraGeometrica = new Circulo(valor1);
                resultado1.setText(Double.toString(figuraGeometrica.area()));
                resultado2.setText(Double.toString(figuraGeometrica.perimetro()));
                break;
            case "Cuadrado":
                figuraGeometrica = new Cuadrado(valor1);
                resultado1.setText(Double.toString(figuraGeometrica.area()));
                resultado2.setText(Double.toString(figuraGeometrica.perimetro()));
                break;
            case "Rectángulo":
                figuraGeometrica = new Rectangulo(valor1, valor2);
                resultado1.setText(Double.toString(figuraGeometrica.area()));
                resultado2.setText(Double.toString(figuraGeometrica.perimetro()));
                break;
            case "Triángulo":
                figuraGeometrica = new Triangulo(valor1, valor2);
                resultado1.setText(Double.toString(figuraGeometrica.area()));
                resultado2.setText(Double.toString(figuraGeometrica.perimetro()));
        }
    } catch (Exception e) {
        // Manejo de excepciones
    }
}
```


Actividad 3. Programación orientada a objetos

```
        break;
        default:
            JOptionPane.showMessageDialog(this, "Seleccione una figura geométrica válida.", "Error",
JOptionPane.ERROR_MESSAGE);
        break;
    }
    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(this, "Ingrese valores numéricos válidos.", "Error",
JOptionPane.ERROR_MESSAGE);
    }
    }
    }
```

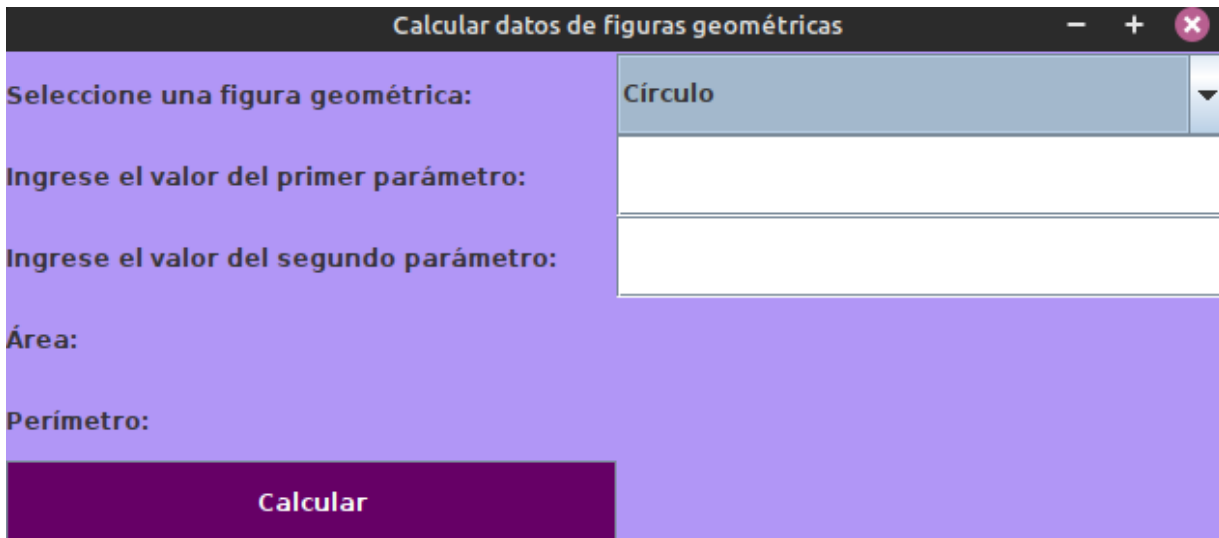
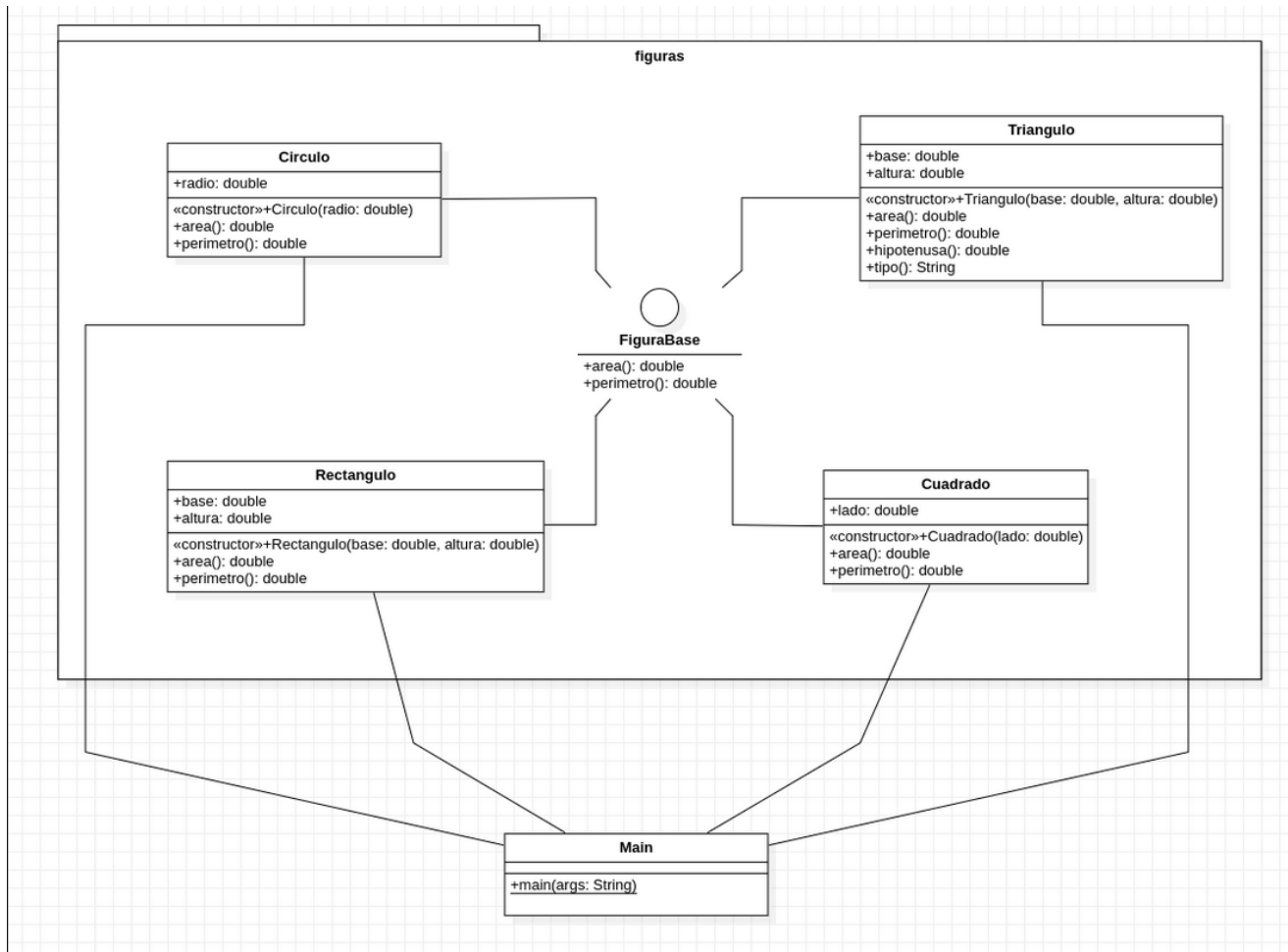


Diagrama UML

Actividad 3. Programación orientada a objetos



https://github.com/Mila-Ag/Actividad3_POO/tree/master/Clases%20sobre%20figuras%20geometricas