Universidad Nacional de Colombia

Tarea5 - Parte1

Autores:

Juan José Jiménez Maya

Programa: Programación Orientada a Objetos

Grupo: 3

Código: Figura.java

```
package Tareas.Tarea5.Parte1.Figuras;

import java.util.List;
import java.util.Map;
import java.util.Optional;
import java.util.stream.Collectors;

public abstract class Figura {
    public abstract String nombre();
    public abstract double volumen(Map<String, FulfilledVariable> valores);
    public abstract double superficie(Map<String, FulfilledVariable> valores);
    public abstract List<Variable> variables();

    public record Variable(String nombre, String id) { }
    public record FulfilledVariable(Variable variable, double valor) { }

    public static Optional<Double> getVariableValue(String id, Map<String, FulfilledVariable> variables) {
        return Optional.ofNullable(variables.get(id)).map(FulfilledVariable::valor);
    }

    public static Map<String, FulfilledVariable> toVariableMap(List<FulfilledVariable> variables) {
        return variables.stream().collect(Collectors.toMap(v -> v.variable().id(), v -> v));
    }
}
```

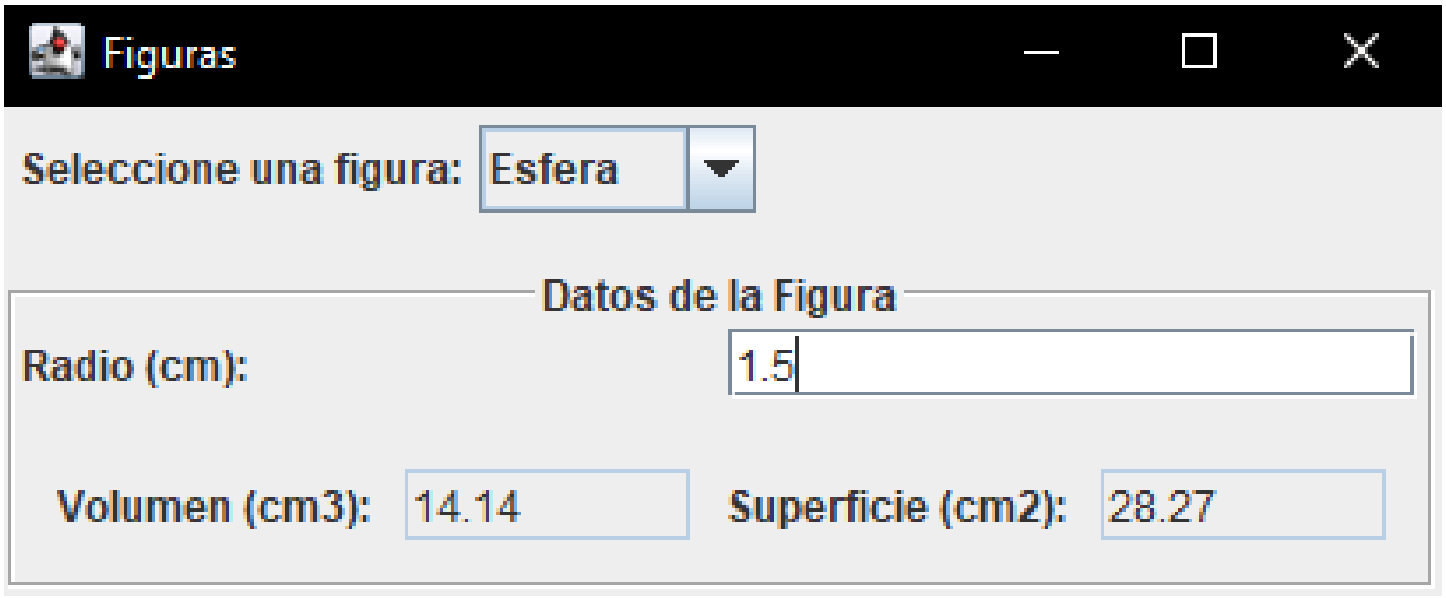
Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Figuras/Figura.java

Código: Piramide.java

```
package Tareas.Tarea5.Parte1.Figuras;

import java.util.List;
import java.util.Map;

public class Piramide extends Figura {
    private final Variable BASE = new Variable("Base (cm)", "base");
    private final Variable ALTURA = new Variable("Altura (cm)", "altura");
    private final Variable APOTEMA = new Variable("Apotema (cm)", "apotema");

    @Override
    public String nombre() {
        return "Piramide";
    }

    @Override
    public double volumen(Map<String, FulfilledVariable> valores) {
        double base = getVariableValue("base", valores).orElse(0.0);
        double altura = getVariableValue("altura", valores).orElse(0.0);

        return (base * base * altura) / 3;
    }

    @Override
    public double superficie(Map<String, FulfilledVariable> valores) {
        double base = getVariableValue("base", valores).orElse(0.0);
        double apotema = getVariableValue("apotema", valores).orElse(0.0);

        return base * (base + 2 * apotema);
    }

    @Override
    public List<Variable> variables() {
        return List.of(BASE, ALTURA, APOTEMA);
    }
}
```

Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Figuras/Piramide.java

Imagen: Esfera.png



Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Figuras/Esfera.png

Código: Cilindro.java

```
package Tareas.Tarea5.Parte1.Figuras;

import java.util.List;
import java.util.Map;

public class Cilindro extends Figura {

    private final Variable RADIO = new Variable("Radio (cm)", "radio");
    private final Variable ALTURA = new Variable("Altura (cm)", "altura");

    @Override
    public String nombre() {
        return "Cilindro";
    }

    @Override
    public double volumen(Map<String, FulfilledVariable> valores) {
        double radio = getVariableValue("radio", valores).orElse(0.0);
        double altura = getVariableValue("altura", valores).orElse(0.0);

        return Math.PI * Math.pow(radio, 2) * altura;
    }

    @Override
    public double superficie(Map<String, FulfilledVariable> valores) {
        double radio = getVariableValue("radio", valores).orElse(0.0);
        double altura = getVariableValue("altura", valores).orElse(0.0);

        return 2 * Math.PI * radio * (radio + altura);
    }

    @Override
    public List<Variable> variables() {
        return List.of(RADIO, ALTURA);
    }
}
```

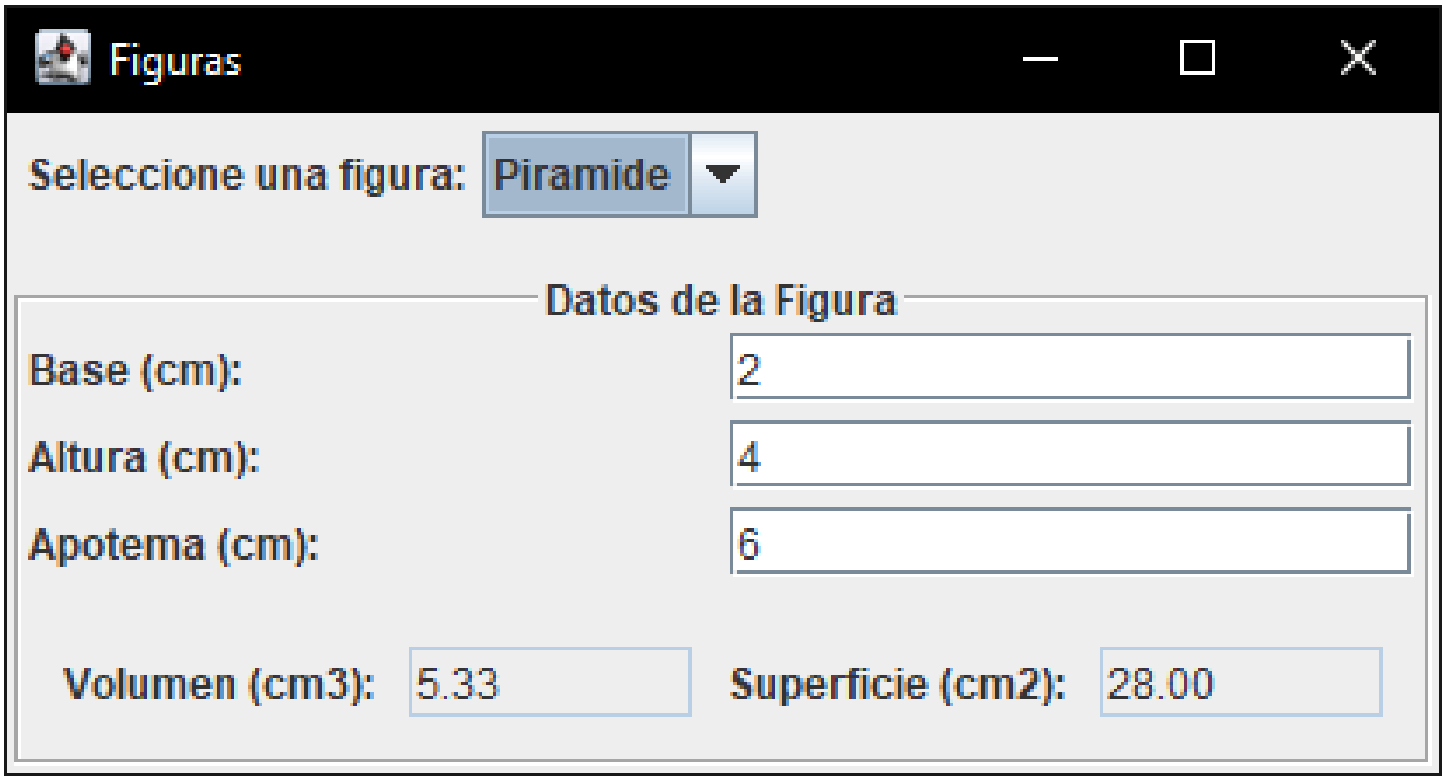
Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Figuras/Cilindro.java

Código: Esfera.java

```
package Tareas.Tarea5.Parte1.Figuras;

import java.util.List;
import java.util.Map;

public class Esfera extends Figura {
    private static final Variable RADIO = new Variable("Radio (cm)", "radio");

    @Override
    public String nombre() {
        return "Esfera";
    }

    @Override
    public double volumen(Map<String, FulfilledVariable> valores) {
        double radio = getVariableValue("radio", valores).orElse(0.0);
        return (4.0 / 3.0) * Math.PI * Math.pow(radio, 3);
    }

    @Override
    public double superficie(Map<String, FulfilledVariable> valores) {
        double radio = getVariableValue("radio", valores).orElse(0.0);
        return 4 * Math.PI * Math.pow(radio, 2);
    }

    @Override
    public List<Variable> variables() {
        return List.of(RADIO);
    }


}
```

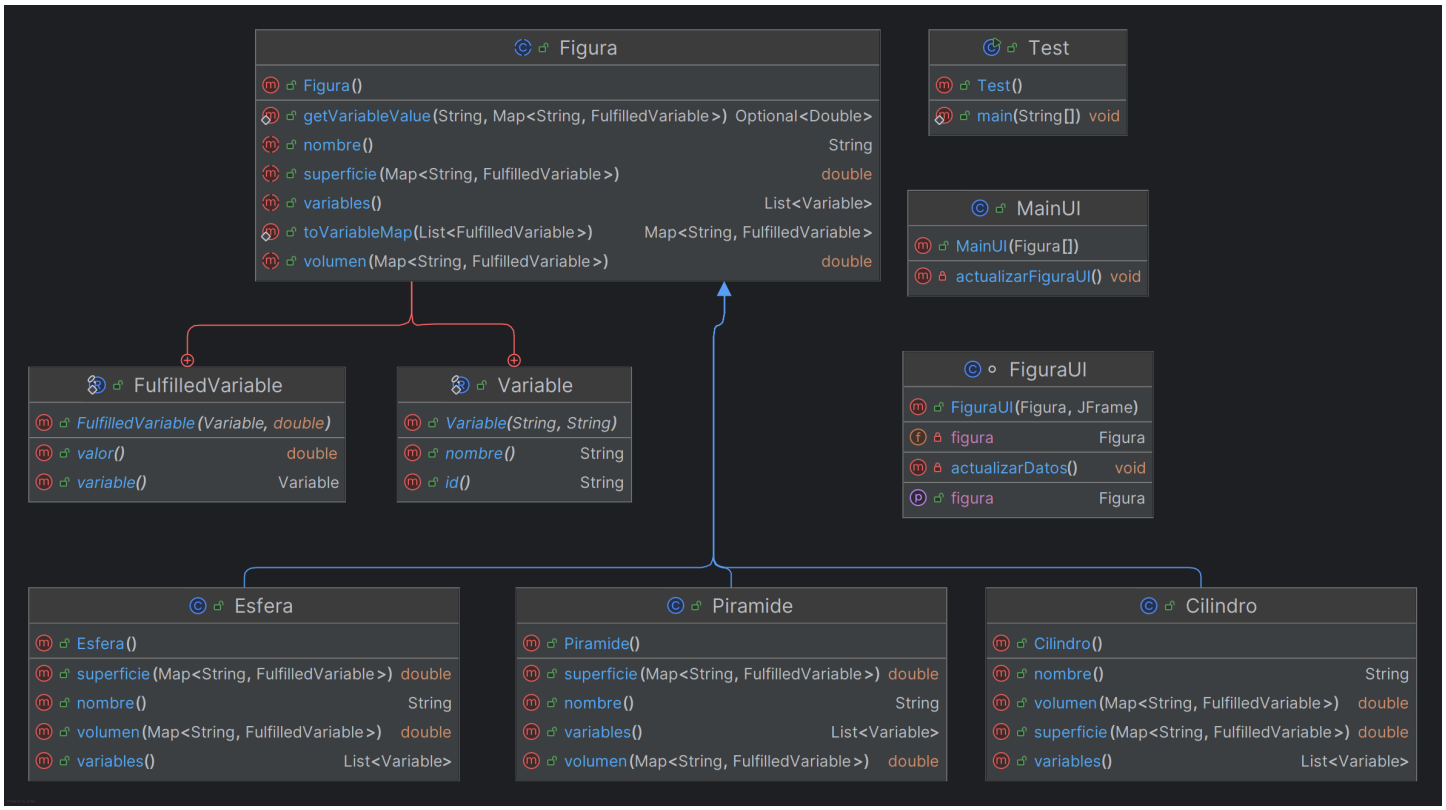
Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Figuras/Esfera.java

Imagen: Piramide.png



Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Figuras/Piramide.png

Imagen: DiagramaUML.png



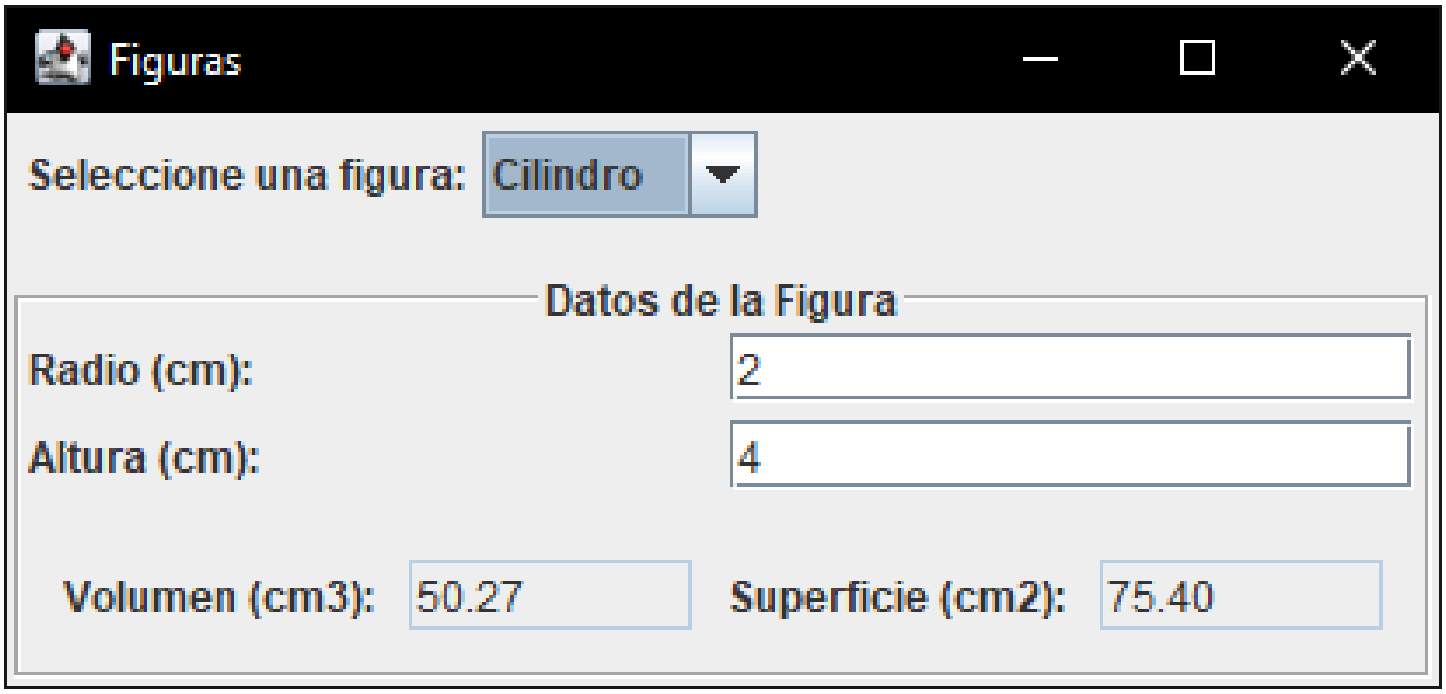
Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Figuras/DiagramaUML.png

Código: Test.java

```
package Tareas.Tarea5.Parte1.Figuras;

public class Test {

    public static void main(String[] args) {
        Figura[] figuras = new Figura[] {
            new Esfera(), new Cilindro(), new Piramide()
        };

        MainUI mainUI = new MainUI(figuras);
    }

}
```

Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Figuras/Test.java

Imagen: Cilindro.png



Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Figuras/Cilindro.png

Código: MainUI.java

```
package Tareas.Tarea5.Parte1.Figuras;

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

import java.util.HashMap;
import java.util.List;
import java.util.Map;

import javax.swing.border.TitledBorder;

public class MainUI extends JFrame {
    private final JComboBox<String> comboBox;
    private final Figura[] figuras;
    private final FiguraUI figuraUI;

    public MainUI(Figura[] figuras) {
        super("Figuras");
        this.figuras = figuras;
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(600, 200);
        this.setLocationRelativeTo(null);
        this.setLayout(new BorderLayout(10, 10));

        JPanel topPanel = new JPanel();
        topPanel.setLayout(new FlowLayout(FlowLayout.LEFT));
        JLabel label = new JLabel("Seleccione una figura:");
        topPanel.add(label);

        comboBox = new JComboBox<>();
        for (Figura figura : figuras) comboBox.addItem(figura.nombre());
        comboBox.addActionListener(e -> actualizarFiguraUI());
        topPanel.add(comboBox);

        this.add(topPanel, BorderLayout.NORTH);

        figuraUI = new FiguraUI(figuras[0], this);
        this.add(figuraUI, BorderLayout.CENTER);

        this.setVisible(true);
    }

    private void actualizarFiguraUI() {
        int index = comboBox.getSelectedIndex();
        if (index >= 0) {
            figuraUI.setFigura(figuras[index]);
        }
    }
}

class FiguraUI extends JPanel {
    private Figura figura;
    private JTextField[] textFields;
    private final JTextField volumenTextField;
    private final JTextField superficieTextField;
    private final JPanel variablesPanel;
    private final JFrame parent;

    public FiguraUI(Figura figura, JFrame parent) {
        this.parent = parent;
```

```
        this.figura = figura;
        this.setLayout(new BorderLayout(10, 10));
             this.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder(), "Datos de la
Figura", TitledBorder.CENTER, TitledBorder.TOP));

        variablesPanel = new JPanel();
        variablesPanel.setLayout(new GridLayout(0, 2, 5, 5));
        this.add(variablesPanel, BorderLayout.CENTER);

        volumenTextField = new JTextField(7);
        volumenTextField.setEditable(false);

        superficieTextField = new JTextField(7);
        superficieTextField.setEditable(false);

        JPanel bottomPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 10, 10));
        bottomPanel.add(new JLabel("Volumen (cm3):"));
        bottomPanel.add(volumenTextField);
        bottomPanel.add(new JLabel("Superficie (cm2):"));
        bottomPanel.add(superficieTextField);
        this.add(bottomPanel, BorderLayout.SOUTH);

        setFigura(figura);
    }

    public void setFigura(Figura figura) {
        this.figura = figura;
        variablesPanel.removeAll();
        List<Figura.Variable> variables = figura.variables();
        textFields = new JTextField[variables.size()];

        for (int i = 0; i < variables.size(); i++) {
            Figura.Variable variable = variables.get(i);
            JLabel label = new JLabel(variable.nombre() + ":");
            variablesPanel.add(label);

            JTextField textField = new JTextField();
            textField.getDocument().addDocumentListener(new DocumentListener() {
                @Override
                public void insertUpdate(DocumentEvent e) { actualizarDatos(); }
                @Override
                public void removeUpdate(DocumentEvent e) { actualizarDatos(); }
                @Override
                public void changedUpdate(DocumentEvent e) { actualizarDatos(); }
            });
            variablesPanel.add(textField);
            textFields[i] = textField;
        }

        revalidate();
        repaint();
        this.parent.pack();
    }

    private void actualizarDatos() {
        Map<String, Figura.FulfilledVariable> valores = new HashMap<>();
        List<Figura.Variable> variables = figura.variables();

        for (int i = 0; i < variables.size(); i++) {
            Figura.Variable variable = variables.get(i);
            String texto = textFields[i].getText();
            try {
                valores.put(variable.id(), new Figura.FulfilledVariable(variable, Double.parseDouble(texto)));
            } catch (Exception ignored) {
                volumenTextField.setText("N/A");
                superficieTextField.setText("N/A");
```

```
                return;
            }
        }
        double volumen = figura.volumen(valores);
        double superficie = figura.superficie(valores);

        volumenTextField.setText(String.format("%.2f", volumen));
        superficieTextField.setText(String.format("%.2f", superficie));
    }
}
```

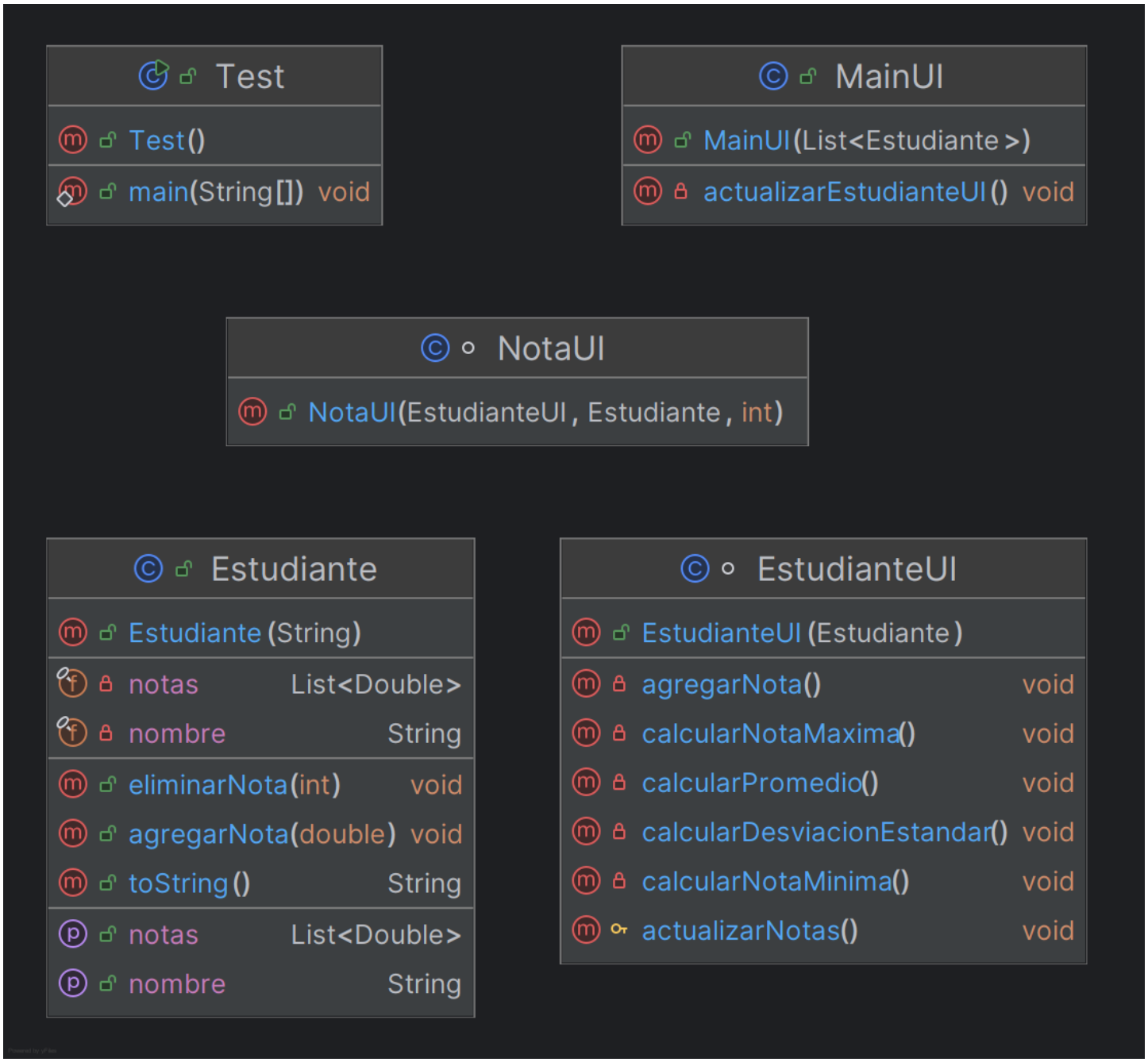
Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Figuras/MainUI.java

Código: Estudiante.java

```
package Tareas.Tarea5.Parte1.Notas;

import java.util.ArrayList;
import java.util.List;

public class Estudiante {
    private final String nombre;
    private final List<Double> notas;

    public Estudiante(String nombre) {
        this.nombre = nombre;
        this.notas = new ArrayList<>();
    }

    public String getNombre() {
        return nombre;
    }

    public List<Double> getNotas() {
        return notas;
    }

    public void agregarNota(double nota) {
        notas.add(nota);
    }

    public void eliminarNota(int index) {
        if (index >= 0 && index < notas.size()) {
            notas.remove(index);
        }
    }

    @Override
    public String toString() {
        return nombre;
    }
}
```

Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Notas/Estudiante.java

Imagen: DiagramaUML.png



Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Notas/DiagramaUML.png

Código: Test.java

```
package Tareas.Tarea5.Parte1.Notas;

import java.util.ArrayList;
import java.util.List;

public class Test {

    public static void main(String[] args) {
        List<Estudiante> estudiantes = new ArrayList<>();
        estudiantes.add(new Estudiante("Juan"));
        estudiantes.add(new Estudiante("Pedro"));
        estudiantes.add(new Estudiante("María"));

        MainUI mainUI = new MainUI(estudiantes);
    }
}
```

Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Notas/Test.java

Imagen: ui.png

Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Notas/ui.png

Código: MainUI.java

```
package Tareas.Tarea5.Parte1.Notas;

import javax.swing.*;

import java.awt.*;
import java.util.List;

public class MainUI extends JFrame {

    private final JComboBox<Estudiante> estudianteComboBox;
    private EstudianteUI activeEstudianteUI;
    private final JPanel mainPanel;

    public MainUI(List<Estudiante> estudiantes) {
        super("Calculadora de Notas");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(800, 300);
        setLocationRelativeTo(null);
        setLayout(new BorderLayout());
        setResizable(false);

        estudianteComboBox = new JComboBox<>(estudiantes.toArray(new Estudiante[0]));
        estudianteComboBox.addActionListener(e -> actualizarEstudianteUI());
        add(estudianteComboBox, BorderLayout.NORTH);

        mainPanel = new JPanel(new BorderLayout());
        add(mainPanel, BorderLayout.CENTER);

        actualizarEstudianteUI();
        setVisible(true);
    }

    private void actualizarEstudianteUI() {
        Estudiante estudiante = (Estudiante) estudianteComboBox.getSelectedItem();
        if (estudiante == null) return;

        if (activeEstudianteUI != null) {
            mainPanel.remove(activeEstudianteUI);
        }

        activeEstudianteUI = new EstudianteUI(estudiante);
        mainPanel.add(activeEstudianteUI, BorderLayout.CENTER);
        mainPanel.revalidate();
        mainPanel.repaint();
    }
}

class EstudianteUI extends JPanel {
    private final Estudiante estudiante;
    private final JTextField promedioTextField;
    private final JTextField maxNoteTextField;
    private final JTextField minNoteTextField;
    private final JTextField desvEstTextField;
    private final JPanel notasPanel;

    public EstudianteUI(Estudiante estudiante) {
        this.estudiante = estudiante;
        setLayout(new BorderLayout());

        JLabel nombreLabel = new JLabel("Estudiante: " + estudiante.getNombre());
        add(nombreLabel, BorderLayout.NORTH);
```

```
        notasPanel = new JPanel();
        notasPanel.setLayout(new BoxLayout(notasPanel, BoxLayout.Y_AXIS));
        JScrollPane scrollPane = new JScrollPane(notasPanel);
        add(scrollPane, BorderLayout.CENTER);

        JPanel bottomPanel = new JPanel();
        bottomPanel.setLayout(new FlowLayout());

        JButton agregarNotaButton = new JButton("Agregar Nota");
        agregarNotaButton.addActionListener(e -> agregarNota());
        bottomPanel.add(agregarNotaButton);

        promedioTextField = new JTextField(5);
        promedioTextField.setEditable(false);
        bottomPanel.add(new JLabel("Promedio: "));
        bottomPanel.add(promedioTextField);

        maxNoteTextField = new JTextField(5);
        maxNoteTextField.setEditable(false);
        bottomPanel.add(new JLabel("Nota Máxima: "));
        bottomPanel.add(maxNoteTextField);

        minNoteTextField = new JTextField(5);
        minNoteTextField.setEditable(false);
        bottomPanel.add(new JLabel("Nota Mínima: "));
        bottomPanel.add(minNoteTextField);

        desvEstTextField = new JTextField(5);
        desvEstTextField.setEditable(false);
        bottomPanel.add(new JLabel("Desviación Estándar: "));
        bottomPanel.add(desvEstTextField);

        add(bottomPanel, BorderLayout.SOUTH);
        actualizarNotas();
    }

    private void agregarNota() {
                String input = JOptionPane.showInputDialog(this, "Ingrese nueva nota:", "Agregar Nota",
JOptionPane.PLAIN_MESSAGE);
        try {
            double nota = Double.parseDouble(input);
            estudiante.agregarNota(nota);
            actualizarNotas();
        } catch (NumberFormatException e) {
                            JOptionPane.showMessageDialog(this, "Ingrese un número válido.", "Error",
JOptionPane.ERROR_MESSAGE);
        }
    }

    protected void actualizarNotas() {
        notasPanel.removeAll();
        for (int i = 0; i < estudiante.getNotas().size(); i++) {
            notasPanel.add(new NotaUI(this, estudiante, i));
        }
        calcularPromedio();
        calcularNotaMaxima();
        calcularNotaMinima();
        calcularDesviacionEstandar();

        notasPanel.revalidate();
        notasPanel.repaint();
    }

    private void calcularPromedio() {
        List<Double> notas = estudiante.getNotas();
        double promedio = notas.stream().mapToDouble(Double::doubleValue).average().orElse(0.0);
```

```
        promedioTextField.setText(String.format("%.2f", promedio));
    }

    private void calcularNotaMaxima() {
        List<Double> notas = estudiante.getNotas();
        double max = notas.stream().mapToDouble(Double::doubleValue).max().orElse(0.0);
        maxNoteTextField.setText(String.format("%.2f", max));
    }

    private void calcularNotaMinima() {
        List<Double> notas = estudiante.getNotas();
        double min = notas.stream().mapToDouble(Double::doubleValue).min().orElse(0.0);
        minNoteTextField.setText(String.format("%.2f", min));
    }

    private void calcularDesviacionEstandar() {
        List<Double> notas = estudiante.getNotas();
        double promedio = notas.stream().mapToDouble(Double::doubleValue).average().orElse(0.0);
        double sum = notas.stream().mapToDouble(n -> Math.pow(n - promedio, 2)).sum();
        double desvEst = Math.sqrt(sum / notas.size());
        desvEstTextField.setText(String.format("%.2f", desvEst));
    }
}

class NotaUI extends JPanel {
    public NotaUI(EstudianteUI estudianteUI, Estudiante estudiante, int index) {
        setLayout(new FlowLayout());

        double nota = estudiante.getNotas().get(index);
        JLabel notaLabel = new JLabel(String.format("Nota %d: %.2f", index + 1, nota));
        add(notaLabel);

        JButton eliminarButton = new JButton("Eliminar");
        eliminarButton.addActionListener(e -> {
            estudiante.eliminarNota(index);
            estudianteUI.actualizarNotas();
        });
        add(eliminarButton);
    }
}
```

Enlace: https://github.com/Simpplay/POO-2024-2/tree/master//Tareas/Tarea5/Parte1/Notas/MainUI.java