**“Año de la recuperación y consolidación de la economía peruana”**

**UNIVERSIDAD PERUANA LOS ANDES**



**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**Asignatura:**

**TALLER VII: Desarrollo De Aplicaciones I**  
  
**Apache Maven: Semana 4**

**DOCENTE:** Mg. Raúl Fernández Bejarano

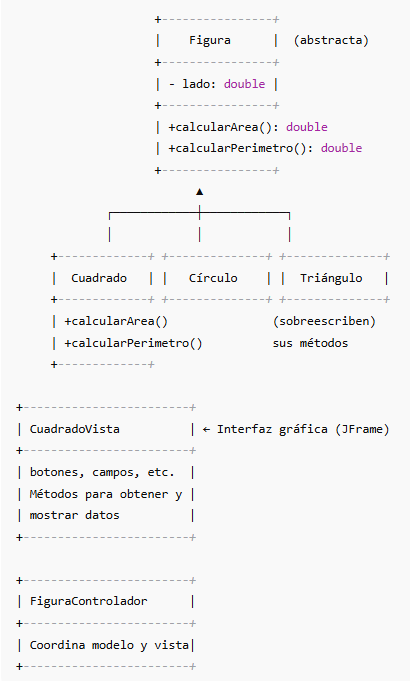
**ESTUDIANTE:**

Huaman Cárdenas, Jhon Frank

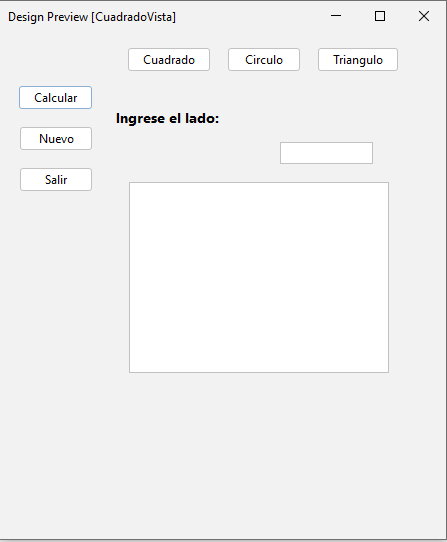
**CICLO – SECCIÓN:** 4to – A1

**Huancayo – 2025**

**1. Enunciado**   
  
Este proyecto consiste en una aplicación de escritorio desarrollada en Java con interfaz gráfica (Swing), que permite al usuario seleccionar una figura geométrica (cuadrado, círculo o triángulo equilátero), ingresar el valor de un lado (o radio) y calcular automáticamente el área y el perímetro de la figura.  
  
 **1.1. Diagrama de Clase**



**1.2. Diseño de la Aplicación**



**1.3. Código de la Aplicación**  
  
 **Clase abstracta Figura**  
 **package Modelo;**

**public abstract class Figura {**

**protected double lado;**

**public Figura(double lado) {**

**this.lado = lado;**

**}**

**public void setLado(double lado) {**

**this.lado = lado;**

**}**

**public double getLado() {**

**return lado;**

**}**

**public abstract double calcularArea();**

**public abstract double calcularPerimetro();**

**}**

**Clase Cuadrado**  
 **package Modelo;**

**public class Cuadrado extends Figura {**

**public Cuadrado(double lado) {**

**super(lado);**

**}**

**@Override**

**public double calcularArea() {**

**return lado \* lado;**

**}**

**@Override**

**public double calcularPerimetro() {**

**return 4 \* lado;**

**}**

**}**

**Clase Circulo**

package Modelo;

public class Circulo extends Figura {

public Circulo(double radio) {  
 super(radio);  
}  
  
@Override  
public double calcularArea() {  
 return Math.PI \* lado \* lado;  
}  
  
@Override  
public double calcularPerimetro() {  
 return 2 \* Math.PI \* lado;  
}

}

**Clase Triangulo**

package Modelo;

public class Triangulo extends Figura {

public Triangulo(double lado) {  
 super(lado);  
}  
  
@Override  
public double calcularArea() {  
 return (Math.sqrt(3) / 4) \* lado \* lado;  
}  
  
@Override  
public double calcularPerimetro() {  
 return 3 \* lado;  
}

}

**Clase FiguraControlador**

**package Controlador;**

**import Modelo.\*;**

**import Vista.CuadradoVista;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**public class FiguraControlador {**

**private Figura modelo;**

**private final CuadradoVista vista;**

**public FiguraControlador(CuadradoVista vista) {**

**this.vista = vista;**

**this.vista.agregarCalcularListener(new ActionListener() {**

**@Override**

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

**calcular();**

**}**

**});**

**this.vista.agregarNuevoListener(new ActionListener() {**

**@Override**

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

**vista.limpiarCampos();**

**}**

**});**

**this.vista.agregarSairListener(new ActionListener() {**

**@Override**

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

**System.exit(0);**

**}**

**});**

**}**

**public void calcular() {**

**double lado = vista.obtenerLado();**

**String figura = vista.getFiguraSeleccionada();**

**switch (figura) {**

**case "cuadrado":**

**modelo = new Cuadrado(lado);**

**break;**

**case "circulo":**

**modelo = new Circulo(lado);**

**break;**

**case "triangulo":**

**modelo = new Triangulo(lado);**

**break;**

**default:**

**vista.mostrarResultados(0, 0);**

**return;**

**}**

**double area = modelo.calcularArea();**

**double perimetro = modelo.calcularPerimetro();**

**vista.mostrarResultados(area, perimetro);**

**}**

**}**

**JFrame Form (Source)**

**Importamos:**   
 package Vista; import javax.swing.*; import java.awt.*; import java.awt.event.\*; import java.awt.image.BufferedImage;

public class CuadradoVista extends javax.swing.JFrame { // Botones figura private String figuraSeleccionada; private PanelDibujo panelFigura;

public CuadradoVista() {  
 initComponents();  
 formulario();  
 this.setTitle("Área y Perímetro de Figuras");  
 setLocationRelativeTo(null); // Centra la ventana  
  
 btnCalcular.setEnabled(false);  
 btnNuevo.setEnabled(false);  
  
 btnSalir.addActionListener(e -> System.exit(0));  
  
 btnCuadrado.addActionListener(e -> seleccionarFigura("cuadrado"));  
 btnCirculo.addActionListener(e -> seleccionarFigura("circulo"));  
 btnTriangulo.addActionListener(e -> seleccionarFigura("triangulo"));  
  
 txtLado.addKeyListener(new KeyAdapter() {  
 @Override  
 public void keyReleased(KeyEvent e) {  
 habilitarBotones();  
 }  
  
 @Override  
 public void keyTyped(KeyEvent evt) {  
 char key = evt.getKeyChar();  
 if (!Character.isDigit(key) && key != '.') {  
 evt.consume();  
 }  
 }  
 });  
  
 btnNuevo.addActionListener(e -> limpiarCampos());  
  
 btnCalcular.addActionListener(e -> {  
 double lado = obtenerLado();  
 if (lado <= 0) {  
 JOptionPane.showMessageDialog(this, "Ingrese un valor válido mayor que 0", "Error", JOptionPane.ERROR\_MESSAGE);  
 return;  
 }  
  
 double area = 0, perimetro = 0;  
 switch (figuraSeleccionada) {  
 case "cuadrado":  
 area = lado \* lado;  
 perimetro = 4 \* lado;  
 break;  
 case "circulo":  
 area = Math.PI \* lado \* lado;  
 perimetro = 2 \* Math.PI \* lado;  
 break;  
 case "triangulo":  
 double altura = Math.sqrt(3) \* lado / 2;  
 area = (lado \* altura) / 2;  
 perimetro = 3 \* lado;  
 break;  
 }  
 mostrarResultados(area, perimetro);  
 });  
}  
  
private void formulario() {  
 this.setSize(650, 450);  
 this.setLayout(null);  
 this.setLocationRelativeTo(null);  
 this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);  
  
 panelSeleccionar.setLayout(null);  
 panelSeleccionar.setBounds(20, 20, 200, 150);  
  
 btnCuadrado.setBounds(20, 10, 150, 30);  
 btnCirculo.setBounds(20, 50, 150, 30);  
 btnTriangulo.setBounds(20, 90, 150, 30);  
  
 panelSeleccionar.add(btnCuadrado);  
 panelSeleccionar.add(btnCirculo);  
 panelSeleccionar.add(btnTriangulo);  
  
 this.add(panelSeleccionar);  
  
 panelDatos.setLayout(null);  
 panelDatos.setBounds(250, 20, 350, 120);  
 panelDatos.setVisible(false);  
  
 txtLado.setBounds(10, 10, 100, 25);  
 panelDatos.add(txtLado);  
  
 txtLado.setBounds(120, 10, 100, 25);  
 panelDatos.add(txtLado);  
  
 JScrollPane scroll = new JScrollPane(txtSalida);  
 scroll.setBounds(10, 50, 300, 60);  
 panelDatos.add(scroll);  
  
 this.add(panelDatos);  
  
 panelCalcular.setLayout(null);  
 panelCalcular.setBounds(250, 160, 350, 50);  
 panelCalcular.setVisible(false);  
  
 btnCalcular.setBounds(10, 10, 100, 30);  
 btnCalcular.setEnabled(false);  
  
 btnNuevo.setBounds(120, 10, 100, 30);  
 btnNuevo.setEnabled(false);  
  
 btnSalir.setBounds(230, 10, 100, 30);  
  
 panelCalcular.add(btnCalcular);  
 panelCalcular.add(btnNuevo);  
 panelCalcular.add(btnSalir);  
  
 this.add(panelCalcular);  
  
 panelFigura = new PanelDibujo();  
 panelFigura.setBounds(20, 220, 580, 180);  
 panelFigura.setBackground(Color.LIGHT\_GRAY);  
 panelFigura.setVisible(false);  
  
 this.add(panelFigura);  
}  
  
  
private void seleccionarFigura(String figura) {  
 this.figuraSeleccionada = figura;  
 panelDatos.setVisible(true);  
 panelCalcular.setVisible(true);  
 panelFigura.setVisible(true);  
 limpiarCampos();  
}  
  
private void habilitarBotones() {  
 boolean habilitar = !txtLado.getText().trim().isEmpty();  
 btnCalcular.setEnabled(habilitar);  
 btnNuevo.setEnabled(habilitar);  
}  
  
public double obtenerLado() {  
 try {  
 return Double.parseDouble(txtLado.getText());  
 } catch (NumberFormatException e) {  
 return 0;  
 }  
}  
  
public void mostrarResultados(double area, double perimetro) {  
 txtSalida.setText("Resultados:\n");  
 txtSalida.append("Área: " + String.format("%.2f", area) + "\n");  
 txtSalida.append("Perímetro: " + String.format("%.2f", perimetro));  
 int ladoInt = (int) obtenerLado();  
 panelFigura.setFigura(figuraSeleccionada);  
 panelFigura.setLado(ladoInt);  
}  
  
public void limpiarCampos() {  
 txtLado.setText("");  
 txtSalida.setText("");  
 panelFigura.setFigura("");  
 panelFigura.setLado(0);  
 btnCalcular.setEnabled(false);  
 btnNuevo.setEnabled(false);  
}  
  
public void agregarCalcularListener(ActionListener listener) {  
 btnCalcular.addActionListener(listener);  
}  
  
public void agregarNuevoListener(ActionListener listener) {  
 btnNuevo.addActionListener(listener);  
}  
  
public void agregarSairListener(ActionListener listener) {  
 btnSalir.addActionListener(listener);  
}  
  
private class PanelDibujo extends JPanel {  
 private String figura = "";  
 private int lado = 0;  
  
 @Override  
 protected void paintComponent(Graphics g) {  
 super.paintComponent(g);  
 if (lado <= 0) return;  
  
 switch (figura) {  
 case "cuadrado":  
 g.setColor(Color.BLUE);  
 g.fillRect(10, 10, lado, lado);  
 break;  
 case "circulo":  
 g.setColor(Color.RED);  
 g.fillOval(10, 10, lado \* 2, lado \* 2);  
 break;  
 case "triangulo":  
 g.setColor(Color.GREEN);  
 int altura = (int) (Math.sqrt(3) \* lado / 2);  
 g.fillPolygon(  
 new int[]{10, 10 + lado / 2, 10 + lado},  
 new int[]{10 + altura, 10, 10 + altura},  
 3);  
 break;  
 }  
 }  
  
 public void setFigura(String f) {  
 this.figura = f;  
 repaint();  
 }  
  
 public void setLado(int l) {  
 this.lado = l;  
 repaint();  
 }  
}  
  
public String getFiguraSeleccionada() {  
 return figuraSeleccionada;  
}  
  
private void txtLadoKeyTyped(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

int key = evt.getKeyChar();

boolean numeros=key >=48 &&key<=57;

if(!numeros){

evt.consume();

}

}  
}

**Clase Main**

**package com.zonajava.enunciado01;**

**import Controlador.FiguraControlador;**

**import Vista.CuadradoVista;**

**public class Enunciado01 {**

**public static void main(String[] args) {**

**CuadradoVista vista = new CuadradoVista();**

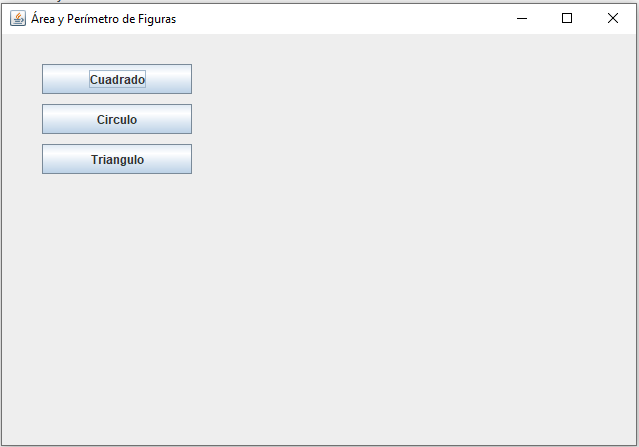
**FiguraControlador controlador = new FiguraControlador(vista);**

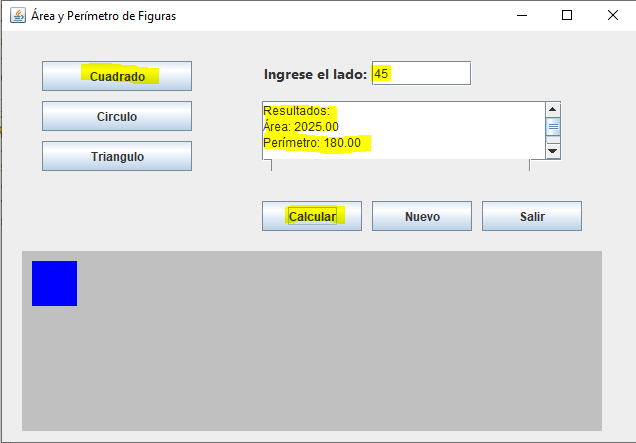
**vista.setVisible(true);**

**}**

**}**

**1.4. Resultado**





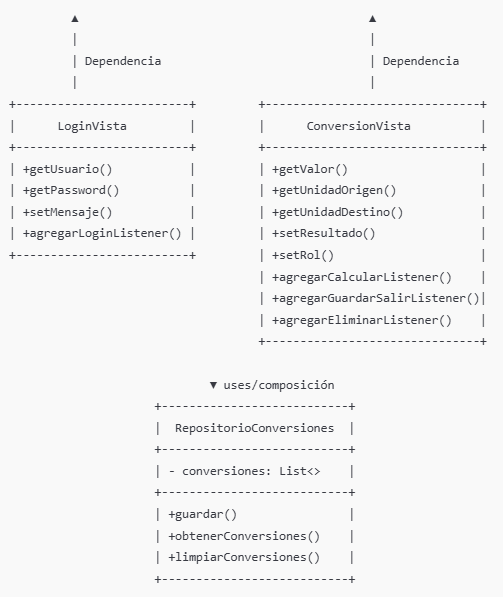
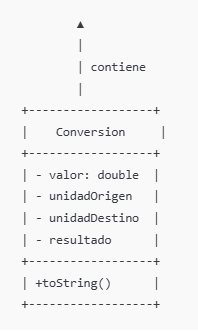
**2. Enunciado**   
Este proyecto es una aplicación de escritorio desarrollada en Java utilizando el paradigma de programación orientada a objetos (POO) y el patrón arquitectónico MVC (Modelo-Vista-Controlador). La finalidad del sistema es ofrecer una herramienta educativa que permite realizar conversiones de unidades de longitud (metros, centímetros, milímetros y kilómetros) a estudiantes y profesores.  
**Objetivos Específicos:**

* Permitir a los usuarios autenticarse según su rol: **Alumno** o **Profesor**.
* Facilitar la conversión de unidades de longitud.
* Almacenar las conversiones realizadas (modo alumno).
* Mostrar las conversiones almacenadas (modo profesor).
* Permitir eliminar todas las conversiones guardadas.

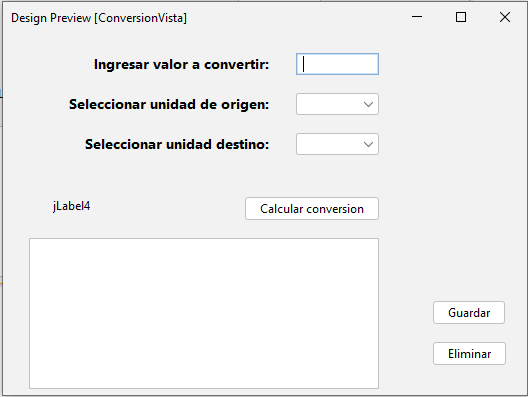
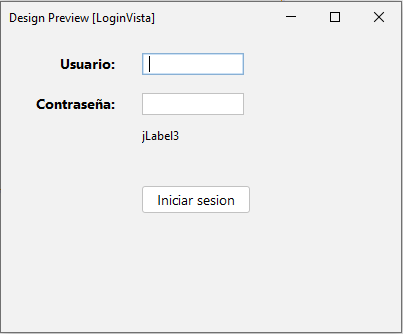
#### **Roles del Sistema:**

* **Alumno:** puede realizar conversiones y guardar los resultados.
* **Profesor:** solo puede visualizar las conversiones almacenadas y eliminarlas.

**2.1. Diagrama de Clase**

**2.2. Diseño de la Aplicación**

ConversionVista(JFrame)  
  
  
LoginVista(JFrame)  


**2.3. Código de la Aplicación**

**Java Class (Conversion.java)**  
 **package Modelo;**

**public class Conversion {**

**private double valor;**

**private String unidadOrigen;**

**private String unidadDestino;**

**private double resultado;**

**public Conversion(double valor, String unidadOrigen, String unidadDestino, double resultado) {**

**this.valor = valor;**

**this.unidadOrigen = unidadOrigen;**

**this.unidadDestino = unidadDestino;**

**this.resultado = resultado;**

**}**

**@Override**

**public String toString() {**

**return valor + " " + unidadOrigen + " = " + resultado + " " + unidadDestino;**

**}**

**}**

**Java Class (RepositorioConversiones.java)**  
 **package Modelo;**

**import java.util.ArrayList;**

**import java.util.List;**

**public class RepositorioConversiones {**

**public static List<Conversion> conversiones = new ArrayList<>();**

**public static void guardar(Conversion c) {**

**conversiones.add(c);**

**}**

**public static List<Conversion> obtenerConversiones() {**

**return conversiones;**

**}**

**public static void limpiarConversiones() {**

**conversiones.clear();**

**}**

**}**

**Java Class (Usuario.java)**

**package Modelo;**

**public class Usuario {**

**private String username;**

**private String password;**

**private String rol; // "ALUMNO" o "PROFESOR"**

**public Usuario(String username, String password, String rol) {**

**this.username = username;**

**this.password = password;**

**this.rol = rol;**

**}**

**public String getUsername() {**

**return username;**

**}**

**public String getPassword() {**

**return password;**

**}**

**public String getRol() {**

**return rol;**

**}**

**}**

**Java Class(ConversionControlador.java)**

**package Controlador;**

**import Modelo.\*;**

**import Vista.ConversionVista;**

**import Vista.LoginVista;**

**import java.util.List;**

**public class ConversionControlador {**

**private ConversionVista vista;**

**private Usuario usuario;**

**private List<Usuario> usuarios;**

**public ConversionControlador(ConversionVista vista, Usuario usuario, List<Usuario> usuarios) {**

**this.vista = vista;**

**this.usuario = usuario;**

**this.usuarios = usuarios;**

**vista.setRol(usuario.getRol());**

**vista.agregarCalcularListener(e -> calcular());**

**vista.agregarGuardarSalirListener(e -> guardarYSalir());**

**vista.agregarEliminarListener(e -> eliminarTodo());**

**if ("PROFESOR".equalsIgnoreCase(usuario.getRol())) {**

**mostrarConversionesGuardadas();**

**}**

**}**

**private void calcular() {**

**try {**

**double valor = Double.parseDouble(vista.getValor());**

**String origen = vista.getUnidadOrigen();**

**String destino = vista.getUnidadDestino();**

**double resultado = convertir(valor, origen, destino);**

**vista.setResultado(valor + " " + origen + " = " + resultado + " " + destino);**

**} catch (NumberFormatException ex) {**

**vista.setResultado("Por favor, ingrese un número válido.");**

**}**

**}**

**private double convertir(double valor, String origen, String destino) {**

**double valorEnMetros = switch (origen) {**

**case "cm" -> valor / 100;**

**case "mm" -> valor / 1000;**

**case "km" -> valor \* 1000;**

**default -> valor;**

**};**

**return switch (destino) {**

**case "cm" -> valorEnMetros \* 100;**

**case "mm" -> valorEnMetros \* 1000;**

**case "km" -> valorEnMetros / 1000;**

**default -> valorEnMetros;**

**};**

**}**

**private void guardarYSalir() {**

**try {**

**double valor = Double.parseDouble(vista.getValor());**

**String origen = vista.getUnidadOrigen();**

**String destino = vista.getUnidadDestino();**

**double resultado = convertir(valor, origen, destino);**

**RepositorioConversiones.guardar(new Conversion(valor, origen, destino, resultado));**

**} catch (NumberFormatException ex) {**

**vista.setResultado("Error al guardar: valor no numérico.");**

**}**

**volverAlLogin();**

**}**

**private void eliminarTodo() {**

**RepositorioConversiones.limpiarConversiones();**

**vista.setResultado("Conversiones eliminadas.");**

**}**

**private void mostrarConversionesGuardadas() {**

**StringBuilder sb = new StringBuilder();**

**for (Conversion c : RepositorioConversiones.obtenerConversiones()) {**

**sb.append(c.toString()).append("\n");**

**}**

**vista.mostrarConversiones(sb.toString());**

**}**

**private void volverAlLogin() {**

**vista.dispose();**

**LoginVista loginVista = new LoginVista();**

**new LoginControlador(loginVista, usuarios); // Corrección: ahora pasamos la lista**

**loginVista.setVisible(true);**

**}**

**}**

**Java Class(LoginControlador.java)**

**package Controlador;**

**import Modelo.Usuario;**

**import Vista.LoginVista;**

**import Vista.ConversionVista;**

**import javax.swing.\*;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.util.List;**

**public class LoginControlador {**

**private final LoginVista loginVista;**

**private final List<Usuario> usuarios;**

**public LoginControlador(LoginVista vista, List<Usuario> usuarios) {**

**this.loginVista = vista;**

**this.usuarios = usuarios;**

**this.loginVista.agregarLoginListener(new ActionListener() {**

**@Override**

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

**login();**

**}**

**});**

**}**

**private void login() {**

**String usuario = loginVista.getUsuario();**

**String password = loginVista.getPassword();**

**Usuario userEncontrado = usuarios.stream()**

**.filter(u -> u.getUsername().equals(usuario) && u.getPassword().equals(password))**

**.findFirst()**

**.orElse(null);**

**if (userEncontrado == null) {**

**loginVista.setMensaje("Usuario o contraseña incorrectos.");**

**} else {**

**loginVista.setMensaje("Bienvenido " + userEncontrado.getUsername());**

**abrirConversionVista(userEncontrado);**

**}**

**}**

**private void abrirConversionVista(Usuario usuario) {**

**SwingUtilities.invokeLater(() -> {**

**ConversionVista conversionVista = new ConversionVista();**

**ConversionControlador conversionControlador = new ConversionControlador(conversionVista, usuario, usuarios);**

**conversionVista.setRol(usuario.getRol());**

**conversionVista.setVisible(true);**

**loginVista.dispose();**

**});**

**}**

**}**

**JFrame Form (ConversionVista.** **java)**

**package Vista;**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.ActionListener;**

**public class ConversionVista extends JFrame {**

**public ConversionVista() {**

**initComponents();**

**inicializarElementos();**

**btnCalcular.addActionListener(e->{**

**});**

**}**

**public String getValor() {**

**return txtValor.getText().trim();**

**}**

**public String getUnidadOrigen() {**

**return (String) cmbOrigen.getSelectedItem();**

**}**

**public String getUnidadDestino() {**

**return (String) cmbDestino.getSelectedItem();**

**}**

**public void setResultado(String texto) {**

**txtResultado.setText(texto);**

**}**

**public void setRol(String rol) {**

**lblRol.setText("Rol: " + rol);**

**if ("PROFESOR".equalsIgnoreCase(rol)) {**

**txtValor.setEditable(false);**

**btnCalcular.setEnabled(false);**

**btnGuardarSalir.setVisible(false);**

**} else {**

**txtValor.setEditable(true);**

**btnCalcular.setEnabled(true);**

**btnGuardarSalir.setVisible(true);**

**}**

**}**

**public void agregarCalcularListener(ActionListener listener) {**

**btnCalcular.addActionListener(listener);**

**}**

**public void inicializarElementos() {**

**DefaultListCellRenderer listRenderer = new DefaultListCellRenderer();**

**listRenderer.setHorizontalAlignment(DefaultListCellRenderer.CENTER);**

**cmbOrigen.setRenderer(listRenderer);**

**cmbDestino.setRenderer(listRenderer);**

**cmbOrigen.removeAllItems();**

**cmbDestino.removeAllItems();**

**String[] unidades = {"m", "cm", "mm", "km"};**

**for (String unidad : unidades) {**

**cmbOrigen.addItem(unidad);**

**cmbDestino.addItem(unidad);**

**}**

**cmbOrigen.setSelectedIndex(0);**

**cmbDestino.setSelectedIndex(1);**

**}**

**public void mostrarConversiones(String texto) {**

**txtResultado.setText(texto);**

**}**

**public void agregarGuardarSalirListener(ActionListener listener) {**

**btnGuardarSalir.addActionListener(listener);**

**}**

**public void agregarEliminarListener(ActionListener listener) {**

**btnEliminarTodo.addActionListener(listener);**

**}**

**}**

**JFrame Form (LoginVista)**

**package Vista;**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.ActionListener;**

**public class LoginVista extends javax.swing.JFrame {**

**public LoginVista() {**

**initComponents();**

**btnLogin.addActionListener(e -> {**

**});**

**}**

**public String getUsuario() {**

**return txtUsuario.getText().trim();**

**}**

**public String getPassword() {**

**return new String(txtPassword.getPassword());**

**}**

**public void setMensaje(String mensaje) {**

**lblMensaje.setText(mensaje);**

**}**

**public void agregarLoginListener(ActionListener listener) {**

**btnLogin.addActionListener(listener);**

**}**

**}**

**Java Main Class (Enunciado02)**

package com.zonajava.enunciado02;

import Controlador.LoginControlador; import Modelo.Usuario; import Vista.LoginVista;

import javax.swing.*; import java.util.List; /*\* \*

* @author La Zona Vip

\*/ public class Enunciado02 {

public static void main(String[] args) {  
 SwingUtilities.invokeLater(() -> {  
 List<Usuario> usuarios = List.of(  
 new Usuario("alumno1", "pass1", "ALUMNO"),  
 new Usuario("profesor1", "pass2", "PROFESOR")  
 );  
  
 LoginVista login = new LoginVista();  
 LoginControlador controlador = new LoginControlador(login, usuarios);  
 login.setVisible(true);  
 });  
}

}