

Trabajo 4 Programación Orientada a Objetos

Presentado por:

Jean Carlos Perilla Garcia - Jperillag@unal.edu.co - cc 1122506087
Max Steven Hurtado Ramos - mhurtador@unal.edu.co -cc 1087791006

Profesor:

Walter Hugo Arboleda Mazo
ia.walterarboleda@gmail.com

Sábado 19 de noviembre



Universidad Nacional de Colombia
Facultad de Minas

2022



Aplicació Universidad Nacional de Colombia

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
 this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package universidadnacional;

import javax.swing.JFrame;

public class Universidadnacional {

    public static void main(String[] args) {

        frmprinciapal Principal= new frmprinciapal();
        Principal.setExtendedState(JFrame.MAXIMIZED_BOTH);
        Principal.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Principal.setVisible(true);

    }

}

frmprincipal:

Archivo:
this.pouse

pitágoras :
calcular pitágoras:

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
```



```
frmpitagoras pitagoras= new frmpitagoras();  
dp.add(pitagoras);  
pitagoras.setClosable(true);  
pitagoras.setVisible(true);  
}
```

Ohm:

calcular voltaje:

```
private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    frmvoltaje voltaje= new frmvoltaje();  
    dp.add(voltaje);  
    voltaje.setClosable(true);  
    voltaje.setVisible(true);  
}
```

calcular corriente:

```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    frmCorriente corriente= new frmCorriente();  
    dp.add(corriente);  
    corriente.setClosable(true);  
    corriente.setVisible(true);  
}
```

Acerca:

Acerca de:

```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    frmCorriente corriente= new frmCorriente();  
    dp.add(corriente);  
    corriente.setClosable(true);  
    corriente.setVisible(true);  
}
```



}

pitagoras:

*** Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license**
*** Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template**
***/**

package universidadnacional;

public class pitagorass{

public static double calcular_hipotenusa(double catetoa, double catetob){

double hipotenusa;

hipotenusa= Math.sqrt(Math.pow(catetoa,2) + Math.pow(catetob,2));

return hipotenusa;

}

}

frmpitágoras:

private void btncalcularActionPerformed(java.awt.event.ActionEvent evt) {

double hipotenusa, catetoa, catetob;

catetoa= Double.parseDouble(txtladoA.getText());

catetob= Double.parseDouble(txtladoB.getText());

hipotenusa=pitagorass.calcular_hipotenusa(catetoa, catetob);

txtHIPOTENUSA.setText(String.valueOf(hipotenusa));

}



voltaje:

package universidadnacional;

/*

*** Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license**

*** Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template**
***/**

public class voltaje{

public static double calcular_voltaje(double corriente, double resistencia){

double voltaje;

voltaje= corriente * resistencia;

return voltaje;

}

}

frmvoltaje:

private void txtCalcularActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

double Voltaje, corriente, resistencia;

corriente= Double.parseDouble(txtcorriente.getText());

resistencia= Double.parseDouble(txtResistencia.getText());

Voltaje=voltaje.calcular_voltaje(corriente, resistencia);

txtVoltaje.setText(String.valueOf(Voltaje));

}

corriente:

/*



* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

*/

```
package universidadnacional;
```

```
public class corriente{
```

```
public static double calcular_corriente(double voltaje, double resistencia){
```

```
    double corriente;
```

```
    corriente= voltaje/ resistencia;
```

```
    return corriente;
```

```
}
```

```
}
```

```
frmcorriente:
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    double voltaje, Corriente, resistencia;
```

```
    voltaje= Double.parseDouble(txtVoltaje.getText());
```

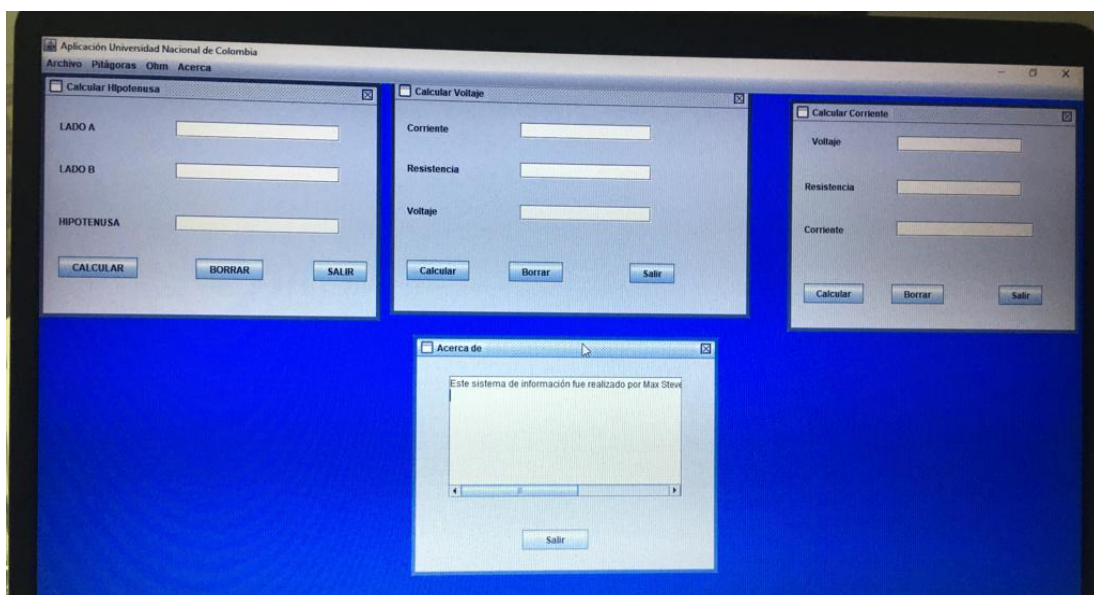
```
    resistencia= Double.parseDouble(txtResistencia.getText());
```

```
    Corriente=corriente.calcular_corriente(voltaje, resistencia);
```

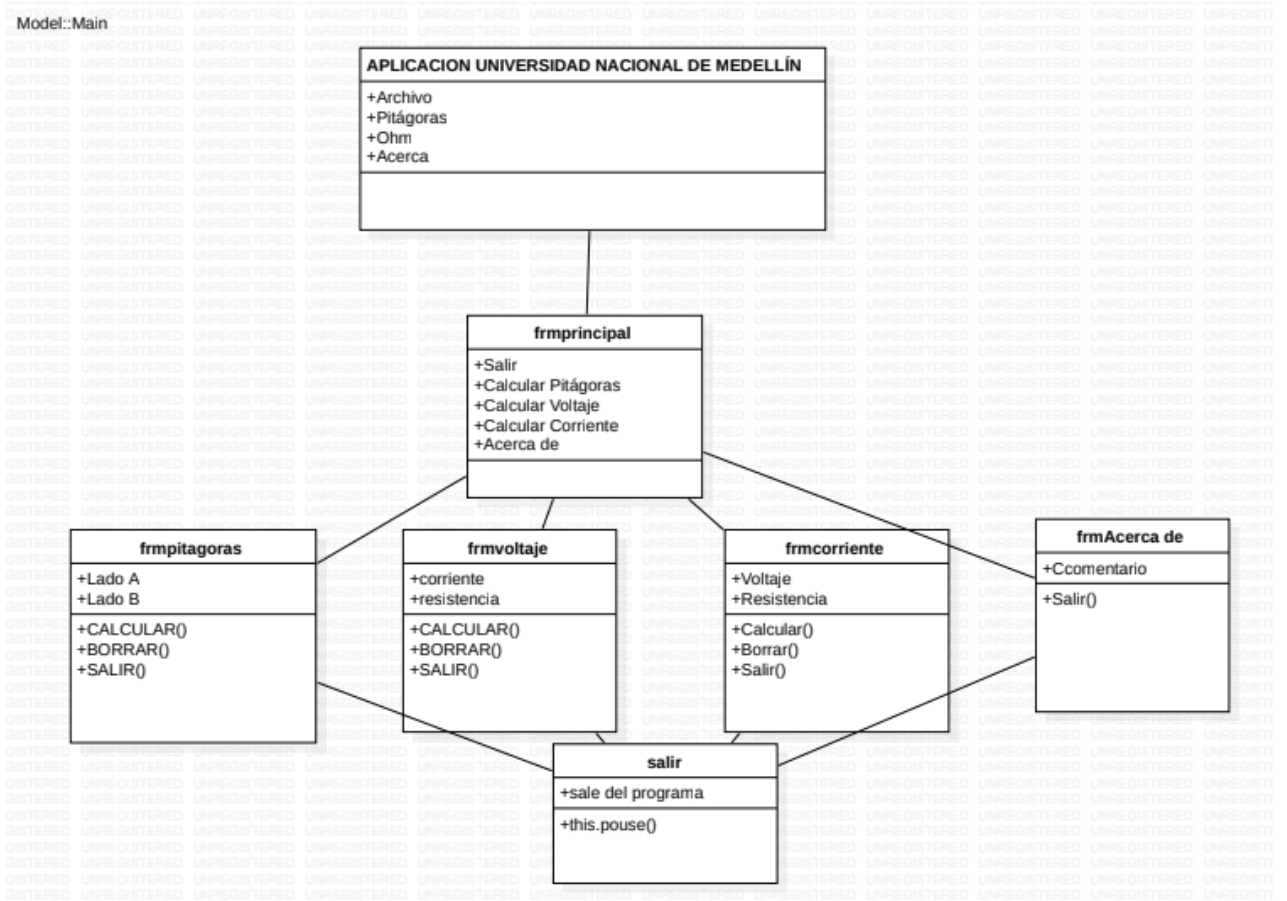
```
    txtCorriente.setText(String.valueOf(Corriente));
```

```
}
```

Imagen parte Gráfica



uml



ejercicio 8.2

Clase Notas:

```
public class Notas {  
    double[] listaNotas;  
    public Notas() {  
        listaNotas = new double[5];  
    }  
  
    double calcularPromedio() {  
        double suma = 0;  
        for(int i=1; i < listaNotas.length; i++) {  
            suma = suma + listaNotas[i];  
        }  
  
        return (suma / listaNotas.length);  
    }  
  
    double calcularDesviación() {  
        double prom = calcularPromedio();  
        double suma = 0;  
        for(int i=0; i < listaNotas.length; i++) {  
  
            suma += Math.pow(listaNotas[i] - prom, 2 );  
        }  
        return Math.sqrt (suma/listaNotas.length );  
    }  
  
    double calcularMenor() {  
        double menor = listaNotas[0];
```




```
for(int i=0; i < listaNotas.length; i++) {  
if (listaNotas[i] < menor) {
```

```
menor = listaNotas[i];  
}  
}  
return menor;  
}
```

```
double calcularMayor() {  
double mayor = listaNotas[0];  
for(int i=0; i < listaNotas.length; i++) {  
if (listaNotas[i] > mayor) {
```

```
mayor = listaNotas[i];  
}  
}  
return mayor;  
}  
}
```

Clase Principal:

```
public class Principal {  
  
public static void main(String[] args) {  
VentanaPrincipal miVentanaPrincipal;  
miVentanaPrincipal= new VentanaPrincipal();  
miVentanaPrincipal.setVisible(true);  
}  
}
```

Clase VentanaPrincipal:

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



```
public class VentanaPrincipal extends JFrame implements
ActionListener {

private Container contenedor;

private JLabel nota1, nota2, nota3, nota4, nota5, promedio,
desviación, mayor, menor;

private JTextField campoNota1, campoNota2, campoNota3,
campoNota4, campoNota5;

private JButton calcular, limpiar;

public VentanaPrincipal(){
inicio();

setTitle("Notas");
setSize(280,380);
setLocationRelativeTo(null);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setResizable(false);
}

private void inicio() {
contenedor = getContentPane();
contenedor.setLayout(null);
nota1 = new JLabel();
nota1.setText("Nota 1:");
nota1.setBounds(20, 20, 135, 23);
campoNota1 = new JTextField();

campoNota1.setBounds(105, 20, 135, 23);
nota2 = new JLabel();
nota2.setText("Nota 2:");
nota2.setBounds(20, 50, 135, 23);
campoNota2 = new JTextField();

campoNota2.setBounds(105, 50, 135, 23);

nota3 = new JLabel();
nota3.setText("Nota 3:");
```



```
nota3.setBounds(20, 80, 135, 23);
campoNota3 = new JTextField();

campoNota3.setBounds(105, 80, 135, 23);

nota4 = new JLabel();
nota4.setText("Nota 4:");
nota4.setBounds(20, 110, 135, 23);
campoNota4 = new JTextField();

campoNota4.setBounds(105, 110, 135, 23);

nota5 = new JLabel();
nota5.setText("Nota 5:");
nota5.setBounds(20, 140, 135, 23);
campoNota5 = new JTextField();

campoNota5.setBounds(105, 140, 135, 23);

calcular = new JButton();
calcular.setText("Calcular");
calcular.setBounds(20, 170, 100, 23);
calcular.addActionListener(this);
// Establece el botón Limpiar
limpiar = new JButton();
limpiar.setText("Limpiar");
limpiar.setBounds(125, 170, 80, 23);
limpiar.addActionListener(this);

promedio = new JLabel();
promedio.setText("Promedio = ");
promedio.setBounds(20, 210, 135, 23);

desviación = new JLabel();
desviación.setText("Desviación = ");
desviación.setBounds(20, 240, 200, 23);
mayor = new JLabel();
mayor.setText("Nota mayor = ");
mayor.setBounds(20, 270, 120, 23);
menor = new JLabel();
menor.setText("Nota menor = ");
menor.setBounds(20, 300, 120, 23);
```



```
contenedor.add(nota1);
contenedor.add(campoNota1);
contenedor.add(nota2);
contenedor.add(campoNota2);
contenedor.add(nota3);
contenedor.add(campoNota3);
contenedor.add(nota4);
contenedor.add(campoNota4);
contenedor.add(nota5);
contenedor.add(campoNota5);
contenedor.add(calcular);
contenedor.add(limpiar);
contenedor.add(promedio);
contenedor.add(desviación);
contenedor.add(mayor);
contenedor.add(menor);
}
```

@Override

```
public void actionPerformed(ActionEvent evento) {

    if (evento.getSource() == calcular) {
        Notas notas = new Notas();
        notas.listaNotas[0] = Double.parseDouble(campoNota1.
            getText());

        notas.listaNotas[1] = Double.parseDouble(campoNota2.
            getText());

        notas.listaNotas[2] = Double.parseDouble(campoNota3.
            getText());

        notas.listaNotas[3] = Double.parseDouble(campoNota4.
            getText());

        notas.listaNotas[4] = Double.parseDouble(campoNota5.
            getText());
        notas.calcularPromedio();
        notas.calcularDesviación();
        promedio.setText("Promedio = " + String.valueOf(String.
            format("%.2f",
            notas.calcularPromedio())));
    }
}
```



```
double desv = notas.calcularDesviación();

desviación.setText("Desviación estándar = " + String.
format("%.2f", desv));

mayor.setText("Valor mayor = " + String.valueOf(notas.
calcularMayor()));

menor.setText("Valor menor = " + String.valueOf(notas.
calcularMenor()));
}

if (evento.getSource() == limpiar) {
campoNota1.setText("");
campoNota2.setText("");
campoNota3.setText("");
campoNota4.setText("");

campoNota5.setText("");
}
}
}
```

Imágenes Parte Gráfica

Notas
— □ ×

Nota 1:

Nota 2:

Nota 3:

Nota 4:

Nota 5:

Calcular
Limpiar

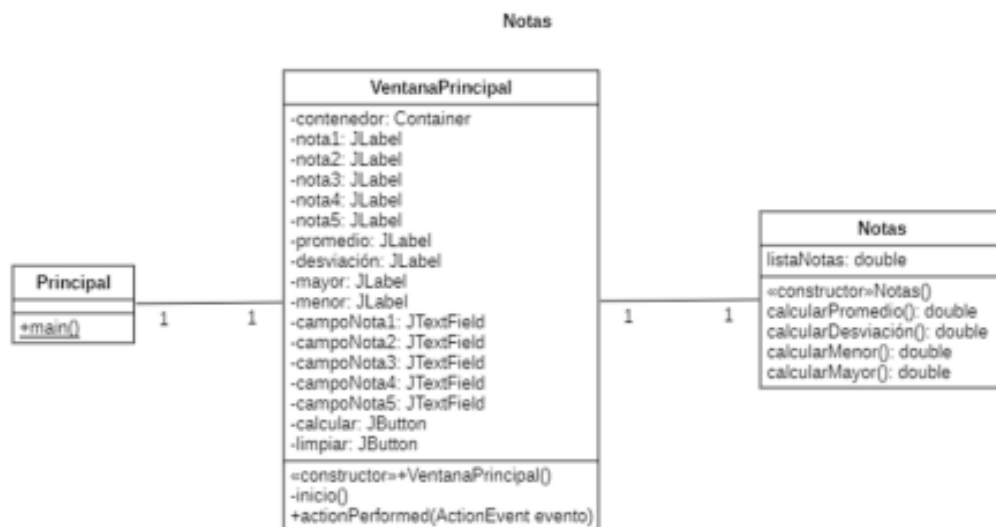
Promedio = 2,60

Desviación estándar = 1,47

Valor mayor = 5.0

Valor menor = 1.0

UML



Ejercicio 8.3

Clase Cilindro:



```
public class Cilindro extends FiguraGeometrica {
    private double radio;
    private double altura;

    public Cilindro(double radio, double altura) {
        this.radio = radio;
        this.altura = altura;
        this.setVolumen(calcularVolumen());
        this.setSuperficie(calcularSuperficie());
    }

    public double calcularVolumen() {
        double volumen = Math.PI * altura * Math.pow(radio, 2.0);
        return volumen;
    }

    public double calcularSuperficie() {
        double áreaLadoA = 2.0 * Math.PI * radio * altura;
        double áreaLadoB = 2.0 * Math.PI * Math.pow(radio, 2.0);
        return áreaLadoA + áreaLadoB;
    }
}
```

clase Esfera:

```
public class Esfera extends FiguraGeometrica {
    private double radio;

    public Esfera(double radio) {
        this.radio = radio;
        this.setVolumen(calcularVolumen());
        this.setSuperficie(calcularSuperficie());
    }

    public double calcularVolumen() {
        double volumen = 1.333 * Math.PI * Math.pow(this.radio, 3.0);
        return volumen;
    }

    public double calcularSuperficie() {
        double superficie = 4.0 * Math.PI * Math.pow(this.radio, 2.0);
        return superficie;
    }
}
```

```
}  
}
```

Clase FiguraGeometrica:

```
public class FiguraGeometrica {  
    private double volumen;  
    private double superficie;  
  
    public void setVolumen(double volumen) {  
        this.volumen = volumen;  
    }  
  
    public void setSuperficie(double superficie) {  
        this.superficie = superficie;  
    }  
  
    public double getVolumen() {  
        return this.volumen;  
    }  
  
    public double getSuperficie() {  
        return this.superficie;  
    }  
}
```

Clase Piramide:

```
public class Piramide extends FiguraGeometrica {  
    private double base;  
    private double altura;  
  
    private double apotema;  
  
    public Piramide(double base, double altura, double apotema) {  
        this.base = base;  
        this.altura = altura;  
        this.apotema = apotema;  
    }  
}
```




```
this.setVolumen(calcularVolumen());  
this.setSuperficie(calcularSuperficie());  
}
```

```
public double calcularVolumen() {  
    double volumen = (Math.pow(base, 2.0) * altura) / 3.0;  
    return volumen;  
}
```

```
public double calcularSuperficie() {  
    double areaBase = Math.pow(base, 2.0);  
    double areaLado = 2.0 * base * apotema;  
    return areaBase + areaLado;  
}  
}
```

Clase Principal:

```
public class Principal {  
  
    public static void main(String[] args) {  
        VentanaPrincipal miVentanaPrincipal;  
        miVentanaPrincipal= new VentanaPrincipal();  
        miVentanaPrincipal.setVisible(true);  
        miVentanaPrincipal.setResizable(false);  
    }  
}
```

Clase VentanaCilindro:

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



```
public class VentanaCilindro extends JFrame implements ActionListener {

    private Container contenedor;

    private JLabel radio, altura, volumen, superficie;

    private JTextField campoRadio, campoAltura;

    private JButton calcular;

    public VentanaCilindro() {
        inicio();
        setTitle("Cilindro");
        setSize(280,210);
        setLocationRelativeTo(null);
        setResizable(false);
    }

    private void inicio() {
        contenedor = getContentPane();
        contenedor.setLayout(null);

        radio = new JLabel();
        radio.setText("Radio (cms):");
        radio.setBounds(20, 20, 135, 23);
        campoRadio = new JTextField();

        campoRadio.setBounds(100, 20, 135, 23);

        altura = new JLabel();
        altura.setText("Altura (cms):");
        altura.setBounds(20, 50, 135, 23);
        campoAltura = new JTextField();

        campoAltura.setBounds(100, 50, 135, 23);

        calcular = new JButton();
        calcular.setText("Calcular");
        calcular.setBounds(100, 80, 135, 23);
        calcular.addActionListener(this);
    }
}
```



```
volumen = new JLabel();
volumen.setText("Volumen (cm3):");

volumen.setBounds(20, 110, 135, 23);

superficie = new JLabel();
superficie.setText("Superficie (cm2):");

superficie.setBounds(20, 140, 135, 23);

contenedor.add(radio);
contenedor.add(campoRadio);
contenedor.add(altura);
contenedor.add(campoAltura);
contenedor.add(calcular);
contenedor.add(volumen);
contenedor.add(superficie);
}

@Override
public void actionPerformed(ActionEvent event) {

    boolean error = false;
    double radio = 0;
    double altura = 0;
    try {

        radio = Double.parseDouble(campoRadio.getText());

        altura = Double.parseDouble(campoAltura.getText());
        Cilindro cilindro = new Cilindro(radio, altura);
        volumen.setText("Volumen (cm3): " + String.format("%.2f", cilindro.calcularVolumen()));

        superficie.setText("Superficie (cm2): " + String.format("%.2f", cilindro.calcularSuperficie()));
    }
    catch (Exception e){
        error = true;

    } finally {
        if(error) {
            JOptionPane.showMessageDialog(null,"Campo nulo o error en formato de numero",
            "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```



```
}  
}  
}  
}
```

Clase VentanaEsfera:

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class VentanaEsfera extends JFrame implements ActionListener {  
    private Container contenedor;  
    private JLabel radio, volumen, superficie;  
    private JTextField campoRadio;  
    private JButton calcular;  
  
    public VentanaEsfera() {  
        inicio();  
        setTitle("Esfera");  
        setSize(280,200);  
        setLocationRelativeTo(null);  
        setResizable(false);  
    }  
  
    private void inicio() {  
        contenedor = getContentPane();  
        contenedor.setLayout(null);  
        radio = new JLabel();  
        radio.setText("Radio (cms):");  
        radio.setBounds(20, 20, 135, 23);  
        campoRadio = new JTextField();  
        campoRadio.setBounds(100, 20, 135, 23);  
        calcular = new JButton();  
        calcular.setText("Calcular");  
        calcular.setBounds(100, 50, 135, 23);  
        calcular.addActionListener(this);  
        volumen = new JLabel();  
        volumen.setText("Volumen (cm3):");
```



```
volumen.setBounds(20, 90, 135, 23);
superficie = new JLabel();
superficie.setText("Superficie (cm2):");
superficie.setBounds(20, 120, 135, 23);
contenedor.add(radio);
contenedor.add(campoRadio);
contenedor.add(calcular);
contenedor.add(volumen);
contenedor.add(superficie);
}
```

@Override

```
public void actionPerformed(ActionEvent evento) {
    if (evento.getSource() == calcular) {
```

```
        boolean error = false;
        try {
```

```
            double radio = Double.parseDouble(campoRadio.getText());
            Esfera esfera = new Esfera(radio);
            volumen.setText("Volumen (cm3): " + String.format("%.2f", esfera.calcularVolumen()));
```

```
            superficie.setText("Superficie (cm2): " + String.format("%.2f", esfera.calcularSuperficie()));
        } catch (Exception e) {
            error = true;
        } finally {
            if(error) {
                JOptionPane.showMessageDialog(null,"Campo nulo o error en formato de número","Error",
                JOptionPane.ERROR_MESSAGE);
            }
        }
    }
}
```

Clase VentanaPiramide:

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



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

public class VentanaPiramide extends JFrame implements
ActionListener {

private Container contenedor;

private JLabel base, altura, apotema, volumen, superficie;

private JTextField campoBase, campoAltura, campoApotema;

private JButton calcular;

public VentanaPiramide() {
inicio();
setTitle("Piramide");
setSize(280,240);

setLocationRelativeTo(null);
setResizable(false);
}

private void inicio() {
contenedor = getContentPane();
contenedor.setLayout(null);
base = new JLabel();
base.setText("Base (cms):");

base.setBounds(20, 20, 135, 23);
campoBase = new JTextField();

campoBase.setBounds(120, 20, 135, 23);

altura = new JLabel();
altura.setText("Altura (cms):");

altura.setBounds(20, 50, 135, 23);
campoAltura = new JTextField();

campoAltura.setBounds(120, 50, 135, 23);
```



```
apotema = new JLabel();
apotema.setText("Apotema (cms):");

apotema.setBounds(20, 80, 135, 23);

campoApotema = new JTextField();

campoApotema.setBounds(120, 80, 135, 23);

calcular = new JButton();
calcular.setText("Calcular");
calcular.setBounds(120, 110, 135, 23);
calcular.addActionListener(this);

volumen = new JLabel();
volumen.setText("Volumen (cm3):");

volumen.setBounds(20, 140, 135, 23);

superficie = new JLabel();
superficie.setText("Superficie (cm2):");

superficie.setBounds(20, 170, 135, 23);

contenedor.add(base);
contenedor.add(campoBase);
contenedor.add(altura);
contenedor.add(campoAltura);
contenedor.add(apotema);
contenedor.add(campoApotema);
contenedor.add(calcular);
contenedor.add(volumen);
contenedor.add(superficie);
}

@Override
public void actionPerformed(ActionEvent event) {
    Piramide piramide;
    boolean error = false;
    double base = 0;
    double altura = 0;
    double apotema = 0;
```



```
try {

base = Double.parseDouble(campoBase.getText());

altura = Double.parseDouble(campoAltura.getText());

apotema = Double.parseDouble(campoApotema.getText());

piramide = new Piramide(base, altura, apotema);

volumen.setText("Volumen (cm3): " + String.format("%.2f",
piramide.calcularVolumen()));

superficie.setText("Superficie (cm2): " + String.format("%.2f",piramide.calcularSuperficie()));
} catch (Exception e) {
error = true;
} finally {
if (error) {
JOptionPane.showMessageDialog(null, "Campo nulo o error en formato de número",
"Error",JOptionPane.ERROR_MESSAGE);
}
}
}
}
```

Clase VentanaPrincipal:

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

public class VentanaPrincipal extends JFrame implements
ActionListener {

private Container contenedor;

private JButton cilindro, esfera, piramide;
```




```
public VentanaPrincipal(){
    inicio();
    setTitle("Figuras");
    setSize(350,160);
    setLocationRelativeTo(null);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
```

```
private void inicio() {
    contenedor = getContentPane();
    contenedor.setLayout(null);
    cilindro = new JButton();
    cilindro.setText("Cilindro");
    cilindro.setBounds(20, 50, 80, 23);
    cilindro.addActionListener(this);
    esfera = new JButton();
    esfera.setText("Esfera");
    esfera.setBounds(125, 50, 80, 23);
    esfera.addActionListener(this);
    piramide = new JButton();
    piramide.setText("Piramide");
    piramide.setBounds(225, 50, 100, 23);
    piramide.addActionListener(this);
    contenedor.add(cilindro);
    contenedor.add(esfera);
    contenedor.add(piramide);
}
```

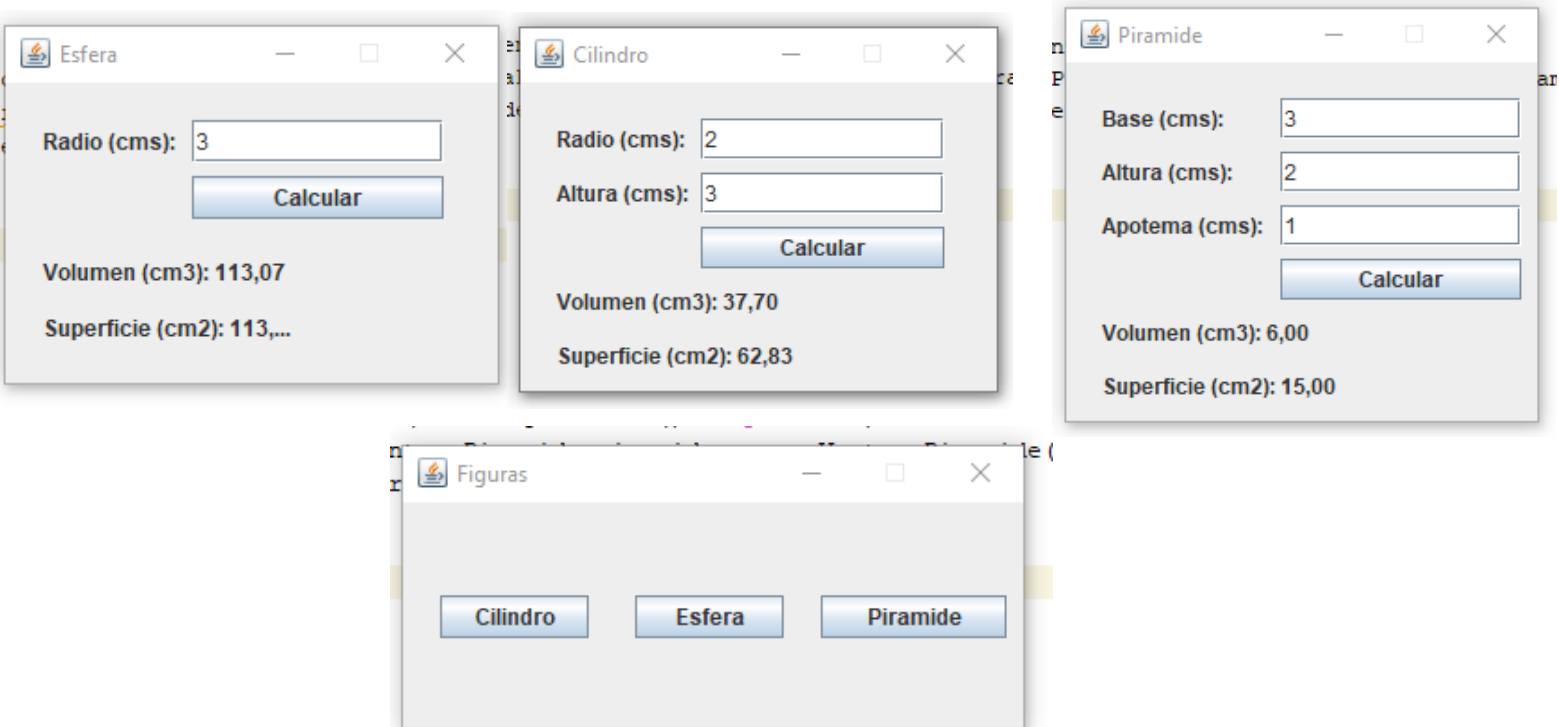
```
@Override
public void actionPerformed(ActionEvent evento) {
    if (evento.getSource() == esfera) {
        VentanaEsfera esfera = new VentanaEsfera();
        esfera.setVisible(true);
    }
    if (evento.getSource() == cilindro) {
        VentanaCilindro cilindro = new VentanaCilindro();
        cilindro.setVisible(true);
    }
    if (evento.getSource() == piramide) {
        VentanaPiramide piramide = new VentanaPiramide();
        piramide.setVisible(true);
    }
}
```

}

}

}

Imagenes parte Gráfica



Esfera

Radio (cms):

Calcular

Volumen (cm3): 113,07

Superficie (cm2): 113,09

Cilindro

Radio (cms):

Altura (cms):

Calcular

Volumen (cm3): 37,70

Superficie (cm2): 62,83

Piramide

Base (cms):

Altura (cms):

Apotema (cms):

Calcular

Volumen (cm3): 6,00

Superficie (cm2): 15,00

Figuras

Cilindro **Esfera** **Piramide**

UNIVERSIDAD NACIONAL DE COLOMBIA

Figuras

