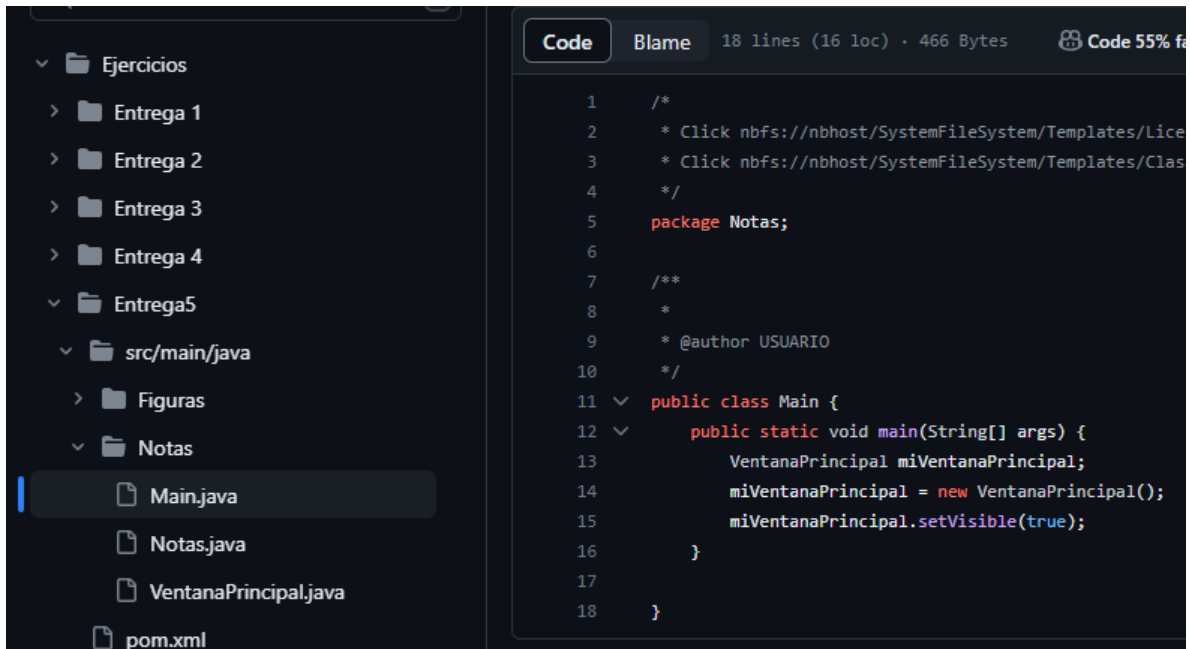


Entrega 5 POO 2023-2
Nombre: Mateo Sánchez
Correo: matsanchez@unal.edu.co
Profesor: Walter Hugo Arboleda Mazo
Link GitHub: <https://github.com/MateoS0/Entregas-POO-2023-2>



```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses to see the template
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
4   */
5  package Notas;
6
7  /**
8   *
9   * @author USUARIO
10   */
11 public class Main {
12     public static void main(String[] args) {
13         VentanaPrincipal miVentanaPrincipal;
14         miVentanaPrincipal = new VentanaPrincipal();
15         miVentanaPrincipal.setVisible(true);
16     }
17
18 }
```

main

+ 🔍

Go to file

Ejercicios

Entrega 1

Entrega 2

Entrega 3

Entrega 4

Entrega5

src/main/java

Figuras

Notas

Main.java

Notas.java

VentanaPrincipal.java

pom.xml

Entregas

.gitignore

README.md

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

/*

* Click nbfs://nbhost/SystemFileSystem/Templates/Lic

* Click nbfs://nbhost/SystemFileSystem/Templates/Clas

*/

package Notas;

/**

*

* @author USUARIO

*/

public class Notas {

double[] listaNotas;

public Notas() {

listaNotas = new double[5];

}

double calcularPromedio() {

double suma = 0;

for(int i = 0; i < listaNotas.length; i++) {

suma = suma + listaNotas[i];

}

return (suma/listaNotas.length);

}

double calcularDesviacion() {

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];

}

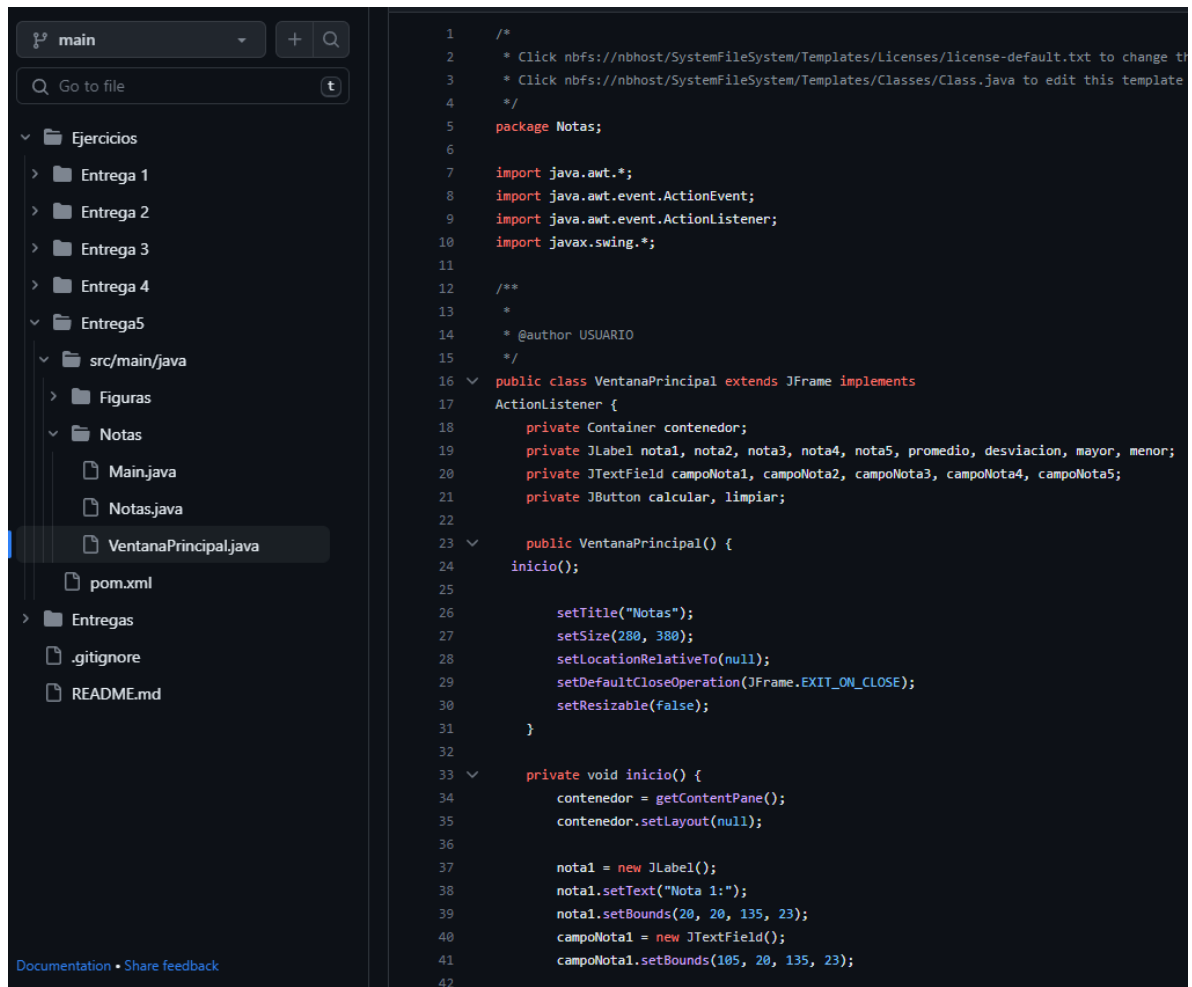
}

return menor;

Documentation

Share feedback

```
43     }
44
45     double calcularMayor() {
46         double mayor = listaNotas[0];
47         for(int i = 0; i < listaNotas.length; i++) {
48             if (listaNotas[i] > mayor) {
49                 mayor = listaNotas[i];
50             }
51         }
52         return mayor;
53     }
54
55 }
```



```
43         nota2 = new JLabel();
44         nota2.setText("Nota 2:");
45         nota2.setBounds(20, 50, 135, 23);
46         campoNota2 = new JTextField();
47         campoNota2.setBounds(105, 50, 135, 23);
48
49         nota3 = new JLabel();
50         nota3.setText("Nota 3:");
51         nota3.setBounds(20, 80, 135, 23);
52         campoNota3 = new JTextField();
53         campoNota3.setBounds(105, 80, 135, 23);
54
55         nota4 = new JLabel();
56         nota4.setText("Nota 4:");
57         nota4.setBounds(20, 110, 135, 23);
58         campoNota4 = new JTextField();
59         campoNota4.setBounds(105, 110, 135, 23);
60
61         nota5 = new JLabel();
62         nota5.setText("Nota 5:");
63         nota5.setBounds(20, 140, 135, 23);
64         campoNota5 = new JTextField();
65         campoNota5.setBounds(105, 140, 135, 23);
66
67         calcular = new JButton();
68         calcular.setText("Calcular");
69         calcular.setBounds(20, 170, 100, 23);
70         calcular.addActionListener(this);
71
72         limpiar = new JButton();
73         limpiar.setText("Limpiar");
74         limpiar.setBounds(125, 170, 80, 23);
75         limpiar.addActionListener(this);
76
77         promedio = new JLabel();
78         promedio.setText("Promedio= ");
79         promedio.setBounds(20, 210, 135, 23);
80
```

```

81         desviacion = new JLabel();
82         desviacion.setText("Desviación= ");
83         desviacion.setBounds(20, 240, 200, 23);
84
85         mayor = new JLabel();
86         mayor.setText("Nota mayor= ");
87         mayor.setBounds(20, 270, 120, 23);
88
89         menor = new JLabel();
90         menor.setText("Nota menor= ");
91         menor.setBounds(20, 300, 120, 23);
92
93         contenedor.add(nota1);
94         contenedor.add(nota2);
95         contenedor.add(nota3);
96         contenedor.add(nota4);
97         contenedor.add(nota5);
98         contenedor.add(campoNota1);
99         contenedor.add(campoNota2);
100        contenedor.add(campoNota3);
101        contenedor.add(campoNota4);
102        contenedor.add(campoNota5);
103        contenedor.add(calcular);
104        contenedor.add(limpiar);
105        contenedor.add(promedio);
106        contenedor.add(desviacion);
107        contenedor.add(mayor);
108        contenedor.add(menor);
109    }
110
111    @Override
112    public void actionPerformed(ActionEvent evento) {
113        if (evento.getSource() == calcular) {
114            Notas notas = new Notas();
115            notas.listaNotas[0] = Double.parseDouble(campoNota1.getText());
116            notas.listaNotas[1] = Double.parseDouble(campoNota2.getText());
117            notas.listaNotas[2] = Double.parseDouble(campoNota3.getText());
118            notas.listaNotas[3] = Double.parseDouble(campoNota4.getText());

```

```
119         notas.listaNotas[4] = Double.parseDouble(campoNota5.getText());
120
121         notas.calcularPromedio();
122         notas.calcularDesviacion();
123         promedio.setText("Promedio= " + String.valueOf(String.format("%.2f", notas.calcularPromedio())));
124         double desv = notas.calcularDesviacion();
125         desviacion.setText("Desviación estándar= " + String.format("%.2f", desv));
126         mayor.setText("Valor mayor= " + String.valueOf(notas.calcularMayor()));
127         menor.setText("Valor menor= " + String.valueOf(notas.calcularMenor()));
128     }
129
130     if (evento.getSource() == limpiar) {
131         campoNota1.setText("");
132         campoNota2.setText("");
133         campoNota3.setText("");
134         campoNota4.setText("");
135         campoNota5.setText("");
136     }
137 }
138
139 }
```

Notas

Nota 1:

Nota 2:

Nota 3:

Nota 4:

Nota 5:

Promedio=

Desviación=

Nota mayor=

Nota menor=

Notas

Nota 1:

Nota 2:

Nota 3:

Nota 4:

Nota 5:

Promedio= 4,00

Desviación estándar= 0,71

Valor mayor= 5.0

Valor menor= 3.0

Notas

Nota 1:

Nota 2:

Nota 3:

Nota 4:

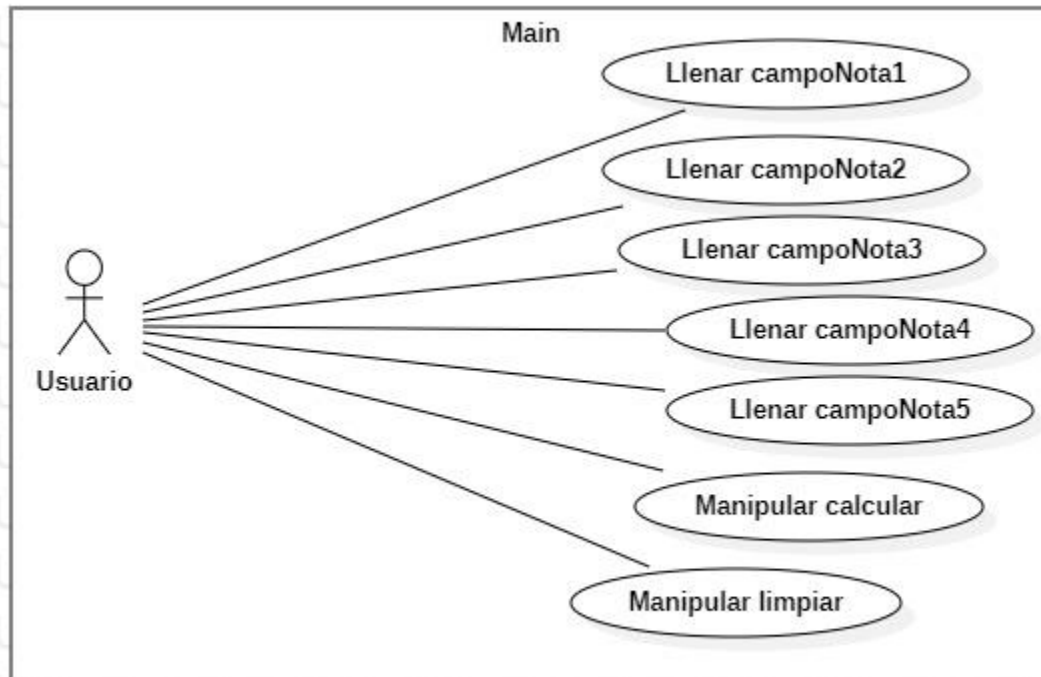
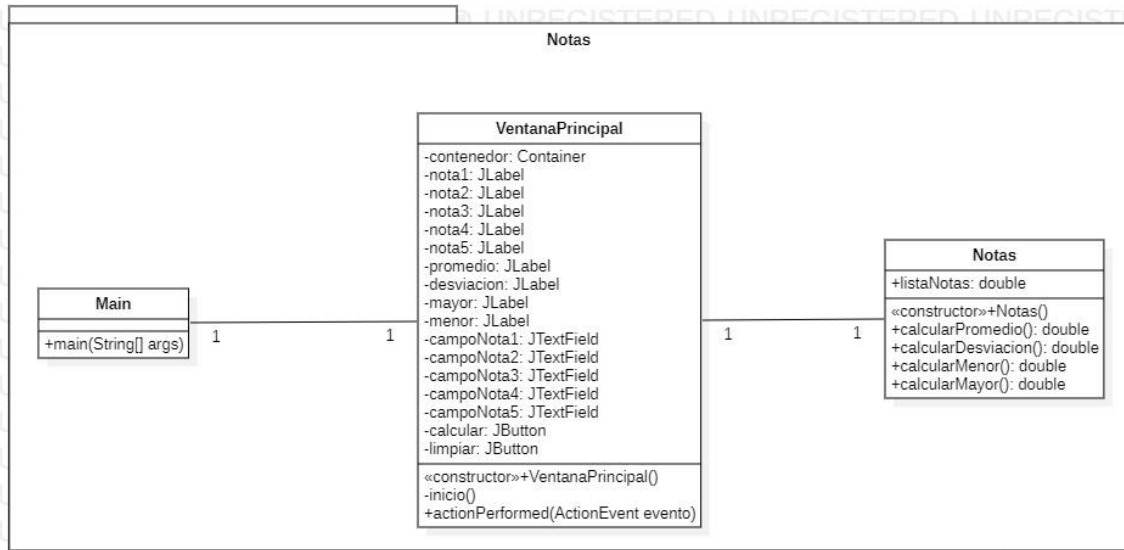
Nota 5:

Promedio= 4,00

Desviación estándar= 0,71

Valor mayor= 5.0

Valor menor= 3.0



Ejercicios

Entrega 1

Entrega 2

Entrega 3

Entrega 4

Entrega5

src/main/java

Figuras

Cilindro.java

Esfera.java

FiguraGeometrica.java

Main.java

Piramide.java

Code

Blame

19 lines (17 loc) · 516 Bytes

Code 55% fa

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/lic
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class
4   */
5   package Figuras;
6
7   /**
8    *
9    * @author USUARIO
10   */
11  public class Main {
12      public static void main(String[] args) {
13          VentanaPrincipal miVentanaPrincipal;
14          miVentanaPrincipal = new VentanaPrincipal();
15          miVentanaPrincipal.setVisible(true);
16          miVentanaPrincipal.setResizable(false);
17      }
18
19  }
```

Ejercicios

Entrega 1

Entrega 2

Entrega 3

Entrega 4

Entrega5

src/main/java

Figuras

Cilindro.java

Esfera.java

FiguraGeometrica.java

Main.java

Piramide.java

VentanaCilindro.java

VentanaEsfera.java

VentanaPiramide.java

VentanaPrincipal.java

Notas

pom.xml

Entregas

.gitignore

README.md

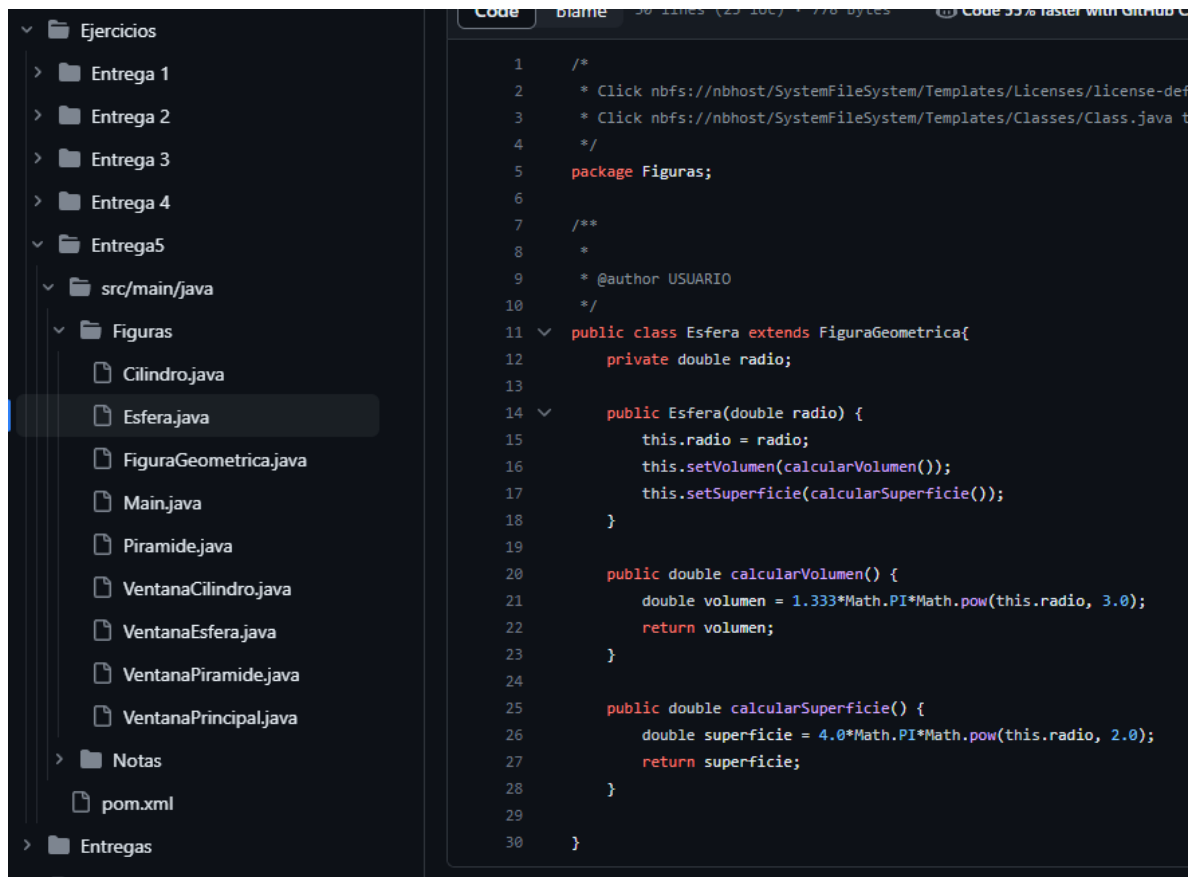
Code

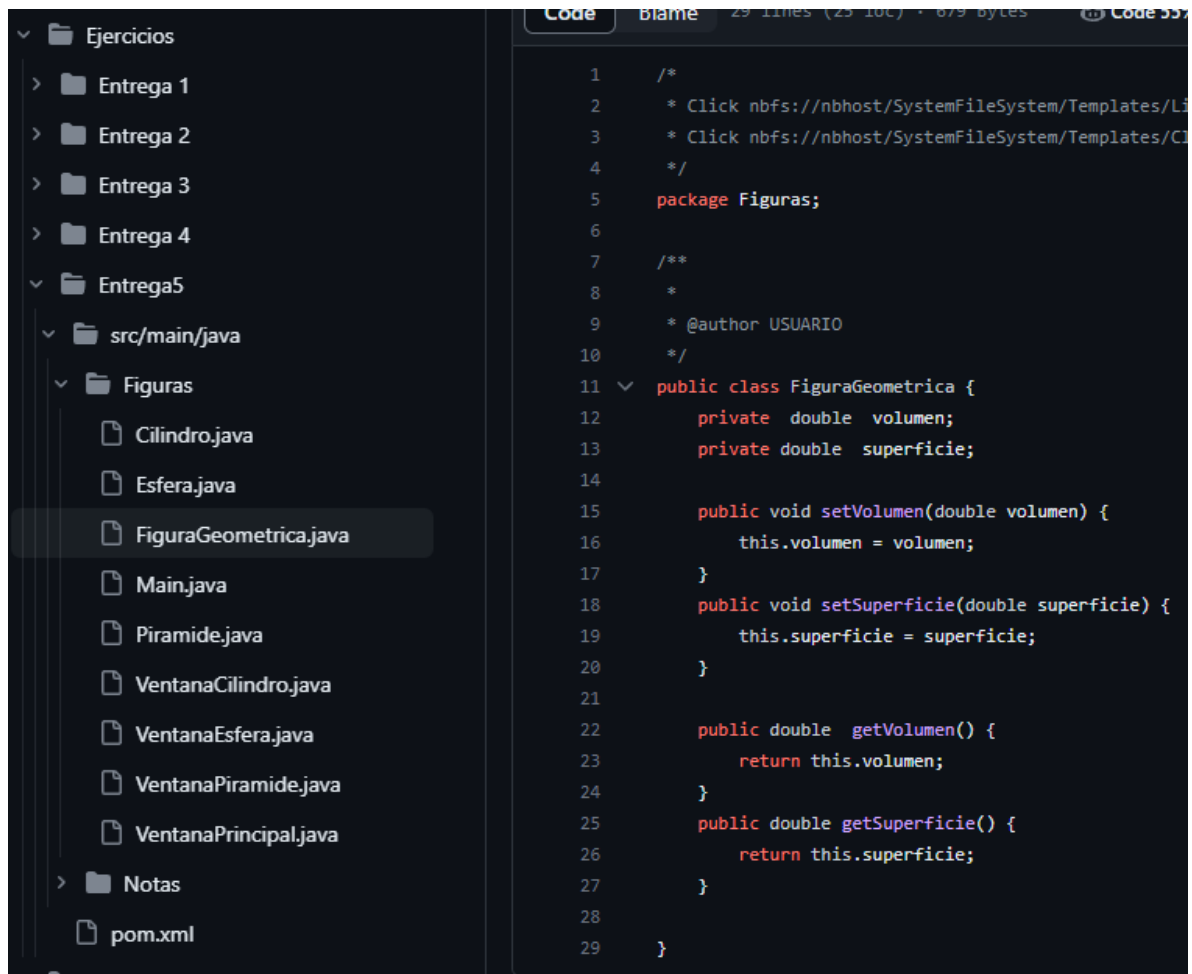
Blame

33 lines (28 loc) · 118 Bytes

Code 55% fa

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/lic
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class
4   */
5   package Figuras;
6
7   /**
8    *
9    * @author USUARIO
10   */
11  public class Cilindro extends FiguraGeometrica{
12      private double radio;
13      private double altura;
14
15      public Cilindro(double radio, double altura) {
16          this.radio = radio;
17          this.altura = altura;
18
19          this.setVolumen(calcularVolumen());
20          this.setSuperficie(calcularSuperficie());
21      }
22
23      public double calcularVolumen() {
24          double volumen = Math.PI*altura*Math.pow(radio, 2.0);
25          return volumen;
26      }
27
28      public double calcularSuperficie() {
29          double areaLadoA = 2.0*Math.PI*radio*altura;
30          double areaLadoB = 2.0*Math.PI*Math.pow(radio, 2.0);
31          return areaLadoA+areaLadoB;
32      }
33  }
```





Ejercicios

Entrega 1

Entrega 2

Entrega 3

Entrega 4

Entrega5

src/main/java

Figuras

Cilindro.java

Esfera.java

FiguraGeometrica.java

Main.java

Piramide.java

VentanaCilindro.java

VentanaEsfera.java

VentanaPiramide.java

VentanaPrincipal.java

Notas

pom.xml

Entregas

.gitignore

README.md

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.jav
4   */
5   package Figuras;
6
7   /**
8    *
9    * @author USUARIO
10   */
11  public class Piramide extends FiguraGeometrica{
12      private double base;
13      private double altura;
14      private double apotema;
15
16      public Piramide(double base, double altura, double apotema) {
17          this.base = base;
18          this.altura = altura;
19          this.apotema = apotema;
20          this.setVolumen(calcularVolumen());
21          this.setSuperficie(calcularSuperficie());
22      }
23
24      public double calcularVolumen() {
25          double volumen = (Math.pow(base, 2.0) * altura) / 3.0;
26          return volumen;
27      }
28
29      public double calcularSuperficie() {
30          double areaBase = Math.pow(base, 2.0);
31          double areaLado = 2.0*base*apotema;
32          return areaBase+areaLado;
33      }
34
35  }
```

main

+

Q

Go to file

t

Ejercicios

Entrega 1

Entrega 2

Entrega 3

Entrega 4

Entrega5

src/main/java

Figuras

Cilindro.java

Esfera.java

FiguraGeometrica.java

Main.java

Piramide.java

VentanaCilindro.java

VentanaEsfera.java

VentanaPiramide.java

VentanaPrincipal.java

Notas

pom.xml

Entregas

.gitignore

README.md

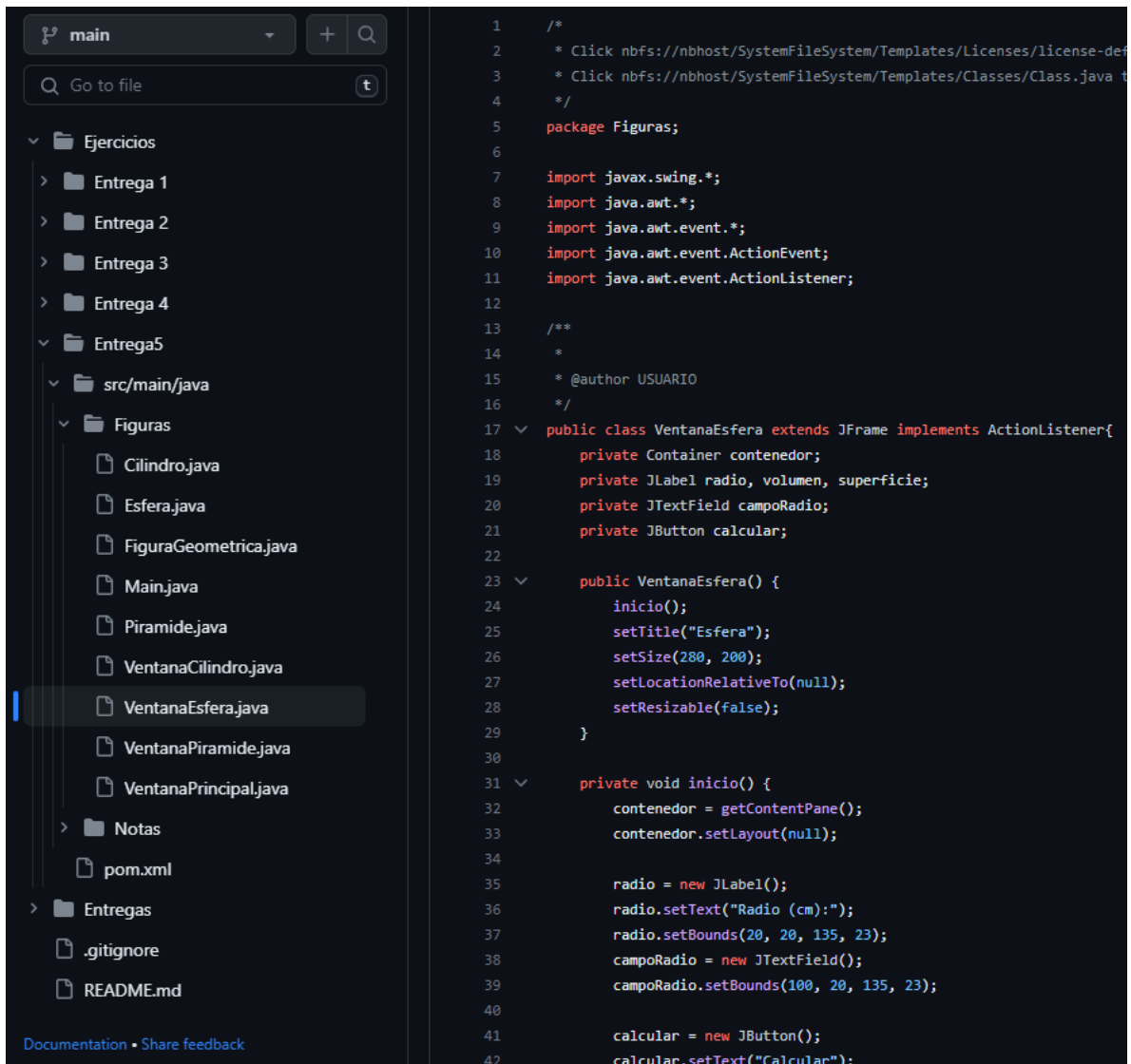
Documentation • Share feedback

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to view and
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
4   */
5   package Figuras;
6
7   import javax.swing.*;
8   import java.awt.*;
9   import java.awt.event.*;
10  import java.awt.event.ActionEvent;
11  import java.awt.event.ActionListener;
12
13  /**
14   *
15   * @author USUARIO
16   */
17  public class VentanaCilindro extends JFrame implements ActionListener{
18      private Container contenedor;
19      private JLabel radio, altura, volumen, superficie;
20      private JTextField campoRadio, campoAltura;
21      private JButton calcular;
22
23      public VentanaCilindro() {
24          inicio();
25          setTitle("Cilindro");
26          setSize(280, 210);
27          setLocationRelativeTo(null);
28          setResizable(false);
29      }
30
31      private void inicio() {
32          contenedor = getContentPane();
33          contenedor.setLayout(null);
34
35          radio = new JLabel();
36          radio.setText("Radio (cm):");
37          radio.setBounds(20, 20, 135, 23);
38          campoRadio = new JTextField();
39          campoRadio.setBounds(100, 20, 135, 23);
40
41          altura = new JLabel();
42          altura.setText("Altura (cm):");
```

```

43         altura.setBounds(20, 50, 135, 23);
44         campoAltura = new JTextField();
45         campoAltura.setBounds(100, 50, 135, 23);
46
47         calcular = new JButton();
48         calcular.setText("Calcular");
49         calcular.setBounds(100, 80, 135, 23);
50         calcular.addActionListener(this);
51
52         volumen = new JLabel();
53         volumen.setText("Volumen (cm3):");
54         volumen.setBounds(20, 110, 135, 23);
55
56         superficie = new JLabel();
57         superficie.setText("Superficie (cm2):");
58         superficie.setBounds(20, 140, 135, 23);
59
60         contenedor.add(radio);
61         contenedor.add(campoRadio);
62         contenedor.add(altura);
63         contenedor.add(campoAltura);
64         contenedor.add(calcular);
65         contenedor.add(volumen);
66         contenedor.add(superficie);
67     }
68
69     public void actionPerformed(ActionEvent event) {
70         boolean error = false;
71         double radio = 0;
72         double altura = 0;
73
74         try {
75             radio = Double.parseDouble(campoRadio.getText());
76             altura = Double.parseDouble(campoAltura.getText());
77
78             Cilindro cilindro = new Cilindro(radio, altura);
79             volumen.setText("Volumen (cm3): " + String.format("%.2f", cilindro.calcularVolumen()));
80             superficie.setText("Superficie (cm2): " + String.format("%.2f", cilindro.calcularSuperficie()));
81
82         } catch (Exception e) {
83             error = true;
84         } finally {
85             if (error) {
86                 JOptionPane.showMessageDialog(null, "Campo nulo o error en formato de número", "Error", JOptionPane.ERROR_MESSAGE);
87             }
88         }
89     }
90 }

```




```

43     calcular.setBounds(100, 50, 135, 23);
44     calcular.addActionListener(this);
45
46     volumen = new JLabel();
47     volumen.setText("Volumen (cm3):");
48     volumen.setBounds(20, 90, 135, 23);
49
50     superficie = new JLabel();
51     superficie.setText("Superficie (cm2):");
52     superficie.setBounds(20, 120, 135, 23);
53
54     contenedor.add(radio);
55     contenedor.add(campoRadio);
56     contenedor.add(calcular);
57     contenedor.add(volumen);
58     contenedor.add(superficie);
59 }
60
61 ✓ public void actionPerformed(ActionEvent evento) {
62     if(evento.getSource() == calcular) {
63         boolean error = false;
64
65         try {
66             double radio = Double.parseDouble(campoRadio.getText());
67
68             Esfera esfera = new Esfera(radio);
69             volumen.setText("Volumen (cm3): " + String.format("%.2f", esfera.calcularVolumen()));
70             superficie.setText("Superficie (cm2): " + String.format("%.2f", esfera.calcularSuperficie()));
71         } catch (Exception e) {
72             error = true;
73         } finally{
74             if(error) {
75                 JOptionPane.showMessageDialog(null, "Campo nulo o error en formato de número", "Error", JOptionPane.ERROR_MESSAGE);
76             }
77         }
78     }
79 }
80

```

```

81 }

```

main

Go to file

Ejercicios

Entrega 1

Entrega 2

Entrega 3

Entrega 4

Entrega5

src/main/java

Figuras

Cilindro.java

Esfera.java

FiguraGeometrica.java

Main.java

Piramide.java

VentanaCilindro.java

VentanaEsfera.java

VentanaPiramide.java

VentanaPrincipal.java

Notas

pom.xml

Entregas

.gitignore

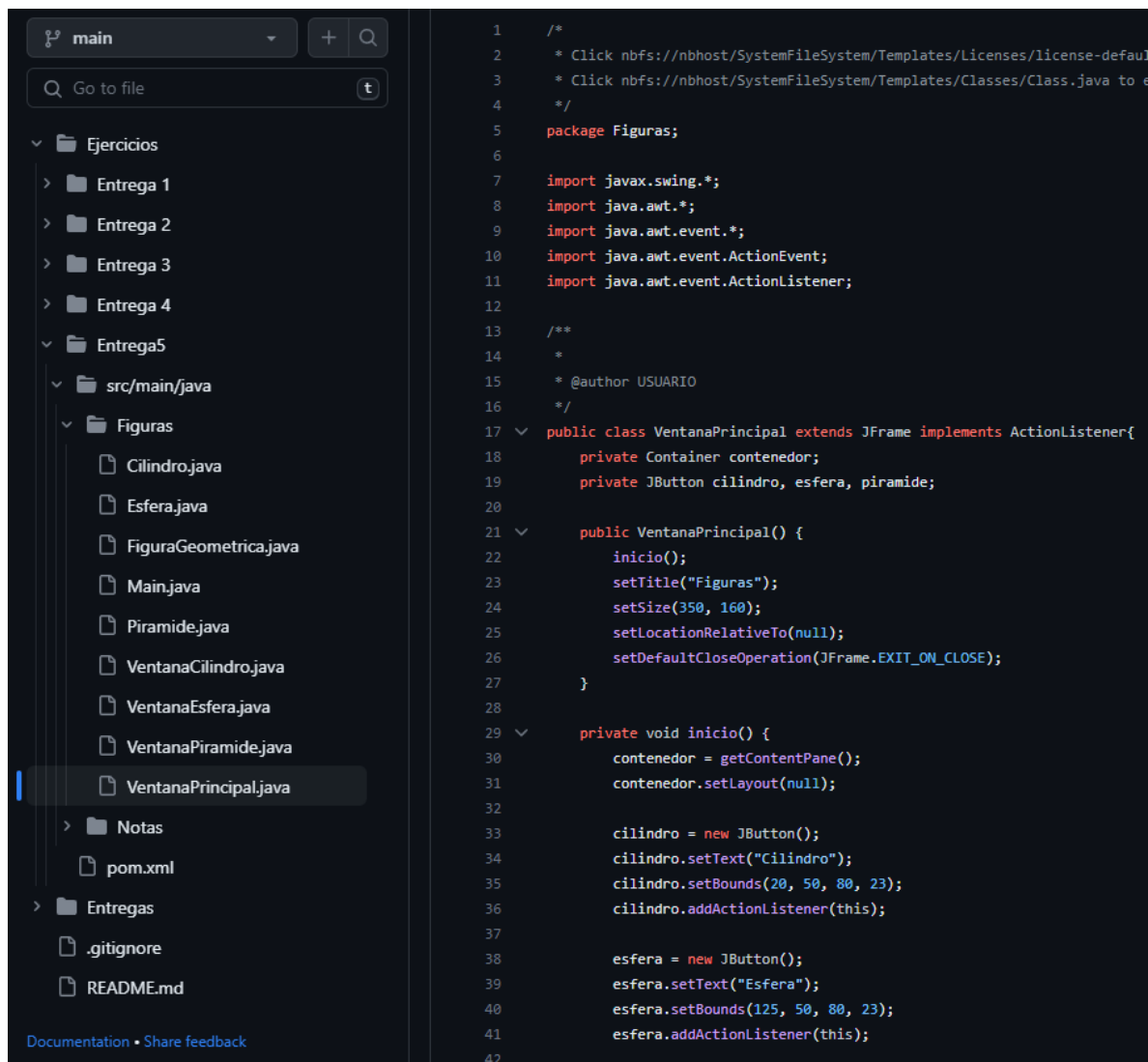
README.md

[Documentation](#) • [Share feedback](#)

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
4   */
5  package Figuras;
6
7  import javax.swing.*;
8  import java.awt.*;
9  import java.awt.event.*;
10 import java.awt.event.ActionEvent;
11 import java.awt.event.ActionListener;
12
13 /**
14  *
15  * @author USUARIO
16  */
17 public class VentanaPiramide extends JFrame implements ActionListener{
18     private Container contenedor;
19     private JLabel base, altura, apotema, volumen, superficie;
20     private JTextField campoBase, campoAltura, campoApotema;
21     private JButton calcular;
22
23     public VentanaPiramide() {
24         inicio();
25         setTitle("Pirámide");
26         setSize(280, 240);
27         setLocationRelativeTo(null);
28         setResizable(false);
29     }
30
31     private void inicio() {
32         contenedor = getContentPane();
33         contenedor.setLayout(null);
34
35         base = new JLabel();
36         base.setText("Base (cm):");
37         base.setBounds(20, 20, 135, 23);
38         campoBase = new JTextField();
39         campoBase.setBounds(120, 20, 135, 23);
40
41         altura = new JLabel();
42         altura.setText("Altura (cm):");
```

```
43         altura.setBounds(20, 50, 135, 23);
44         campoAltura = new JTextField();
45         campoAltura.setBounds(120, 50, 135, 23);
46
47         apotema = new JLabel();
48         apotema.setText("Apotema (cm):");
49         apotema.setBounds(20, 80, 135, 23);
50         campoApotema = new JTextField();
51         campoApotema.setBounds(120, 80, 135, 23);
52
53         calcular = new JButton();
54         calcular.setText("Calcular");
55         calcular.setBounds(120, 110, 135, 23);
56         calcular.addActionListener(this);
57
58         volumen = new JLabel();
59         volumen.setText("Volumen (cm3):");
60         volumen.setBounds(20, 140, 135, 23);
61
62         superficie = new JLabel();
63         superficie.setText("Superficie (cm2):");
64         superficie.setBounds(20, 170, 135, 23);
65
66         contenedor.add(base);
67         contenedor.add(campoBase);
68         contenedor.add(altura);
69         contenedor.add(campoAltura);
70         contenedor.add(apotema);
71         contenedor.add(campoApotema);
72         contenedor.add(calcular);
73         contenedor.add(volumen);
74         contenedor.add(superficie);
75     }
76
77     public void actionPerformed(ActionEvent event) {
78         Piramide piramide;
79         boolean error = false;
80         double base = 0;
```

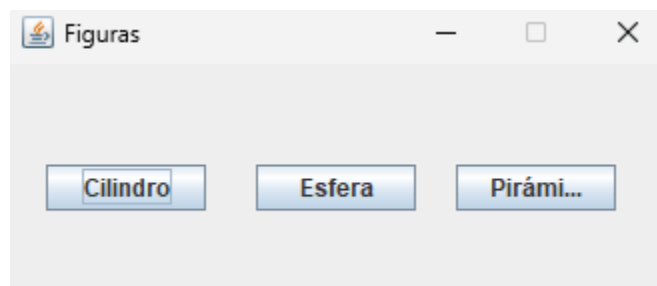
```
81         double altura = 0;
82         double apotema = 0;
83
84         try {
85             base = Double.parseDouble(campoBase.getText());
86             altura = Double.parseDouble(campoAltura.getText());
87             apotema = Double.parseDouble(campoApotema.getText());
88
89             piramide = new Piramide(base, altura, apotema);
90             volumen.setText("Volumen (cm3): " + String.format("%.2f", piramide.calcularVolumen()));
91             superficie.setText("Superficie (cm2): " + String.format("%.2f", piramide.calcularSuperficie()));
92         } catch (Exception e) {
93             error = true;
94         } finally {
95             if (error) {
96                 JOptionPane.showMessageDialog(null, "Campo nulo o error en formato de número", "Error", JOptionPane.ERROR_MESSAGE);
97             }
98         }
99     }
100
101 }
```



```

43         piramide = new JButton();
44         piramide.setText("Pirámide");
45         piramide.setBounds(225, 50, 80, 23);
46         piramide.addActionListener(this);
47
48         contenedor.add(cilindro);
49         contenedor.add(esfera);
50         contenedor.add(piramide);
51     }
52
53     public void actionPerformed(ActionEvent evento) {
54         if (evento.getSource() == esfera) {
55             VentanaEsfera esfera = new VentanaEsfera();
56             esfera.setVisible(true);
57         }
58
59         if (evento.getSource() == cilindro) {
60             VentanaCilindro cilindro = new VentanaCilindro();
61             cilindro.setVisible(true);
62         }
63
64         if (evento.getSource() == piramide) {
65             VentanaPiramide piramide = new VentanaPiramide();
66             piramide.setVisible(true);
67         }
68     }
69
70 }

```



Cilindro		Cilindro	
Radio (cm):	<input type="text"/>	Radio (cm):	<input type="text" value="2"/>
Altura (cm):	<input type="text"/>	Altura (cm):	<input type="text" value="4"/>
<input type="button" value="Calcular"/>		<input type="button" value="Calcular"/>	
Volumen (cm3):		Volumen (cm3):	50,27
Superficie (cm2):		Superficie (cm2):	75,40

Esfera		Esfera	
Radio (cm):	<input type="text"/>	Radio (cm):	<input type="text" value="1.5"/>
<input type="button" value="Calcular"/>		<input type="button" value="Calcular"/>	
Volumen (cm3):		Volumen (cm3):	14,13
Superficie (cm2):		Superficie (cm2):	28,27

Pirámide		Pirámide	
Base (cm):	<input type="text"/>	Base (cm):	<input type="text" value="2"/>
Altura (cm):	<input type="text"/>	Altura (cm):	<input type="text" value="4"/>
Apotema (cm):	<input type="text"/>	Apotema (cm):	<input type="text" value="6"/>
<input type="button" value="Calcular"/>		<input type="button" value="Calcular"/>	
Volumen (cm3):		Volumen (cm3):	5,33
Superficie (cm2):		Superficie (cm2):	28,00

