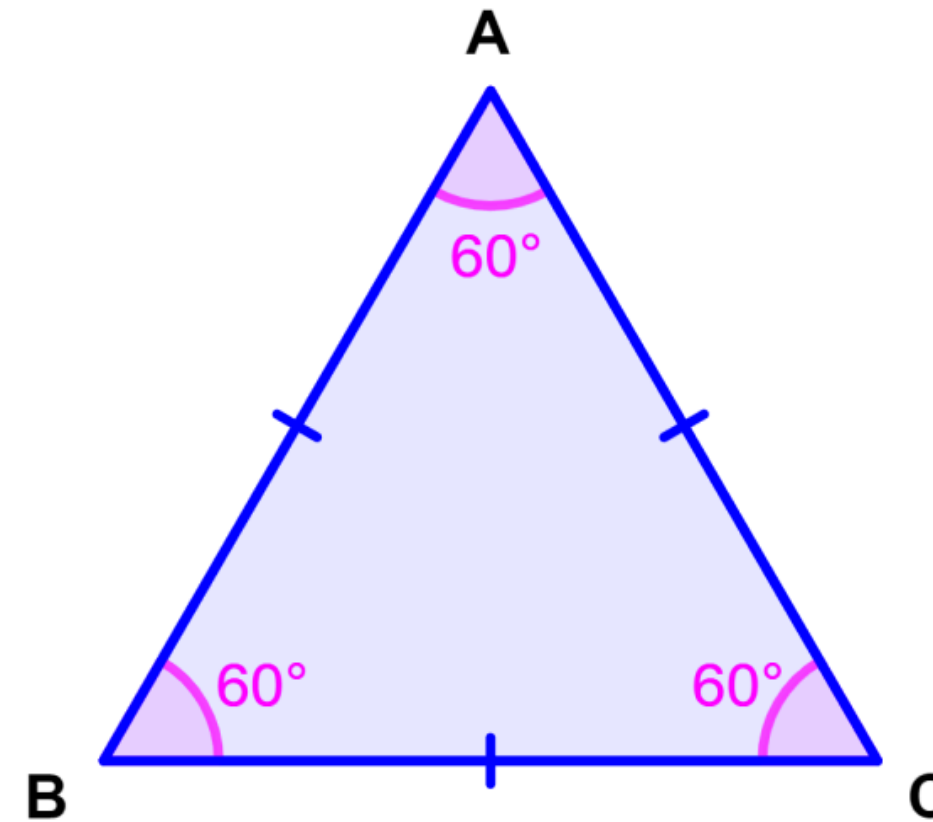


J Reto #4

Triángulo equilátero

Equipo 6

Contreras Martínez Alan Gael
Gómez Rosales Roberto Josué.
Guzmán Fernández Andrés Rogelio
Martínez Ruiz Abdiel Barush
Santos Mateos Oswaldo



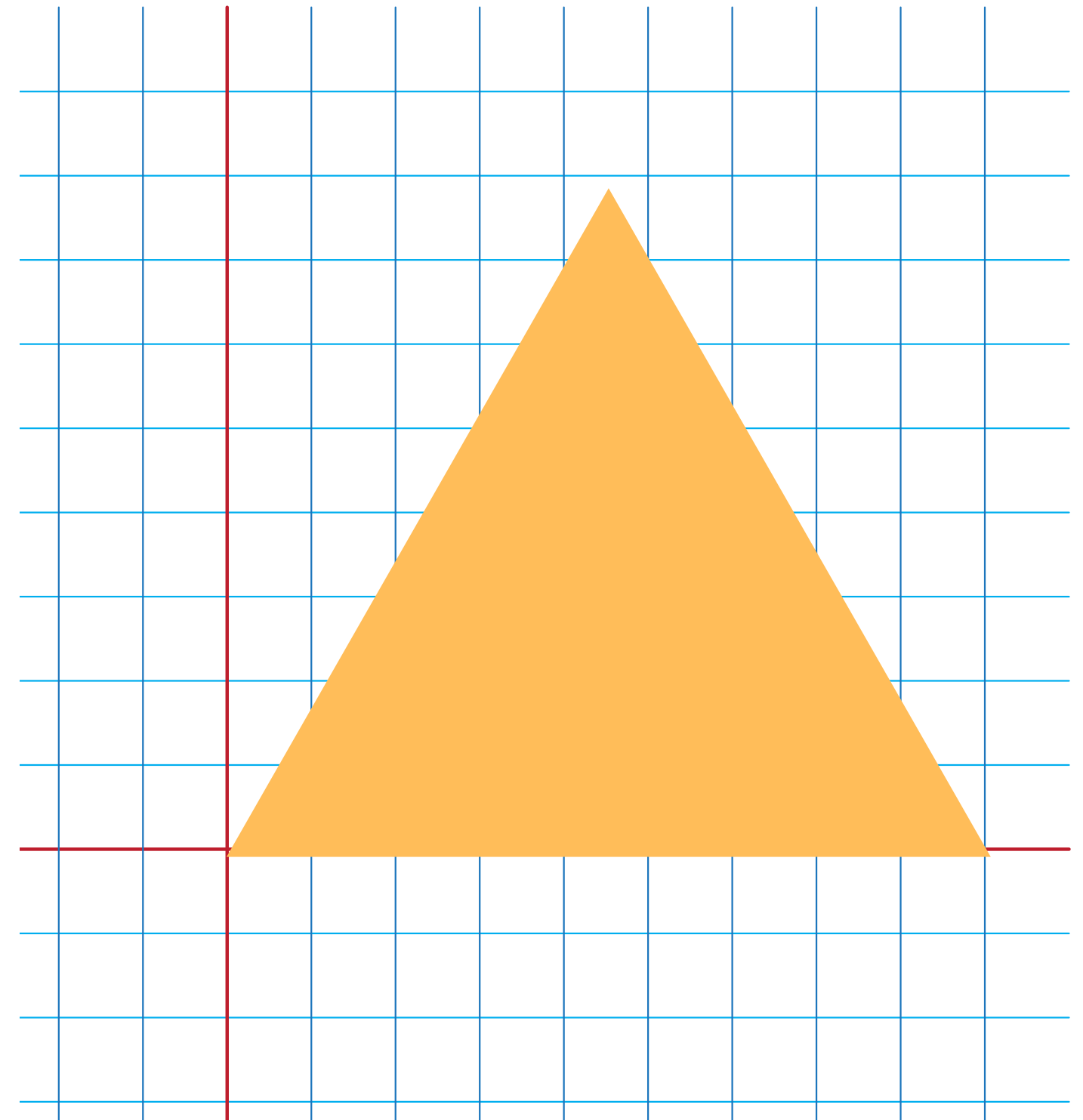
Triángulo equilátero

Se trata de un polígono regular que se caracteriza por:

Tener sus tres lados de igual longitud.

Sus tres ángulos interiores de igual medida, cada uno de 60 grados.

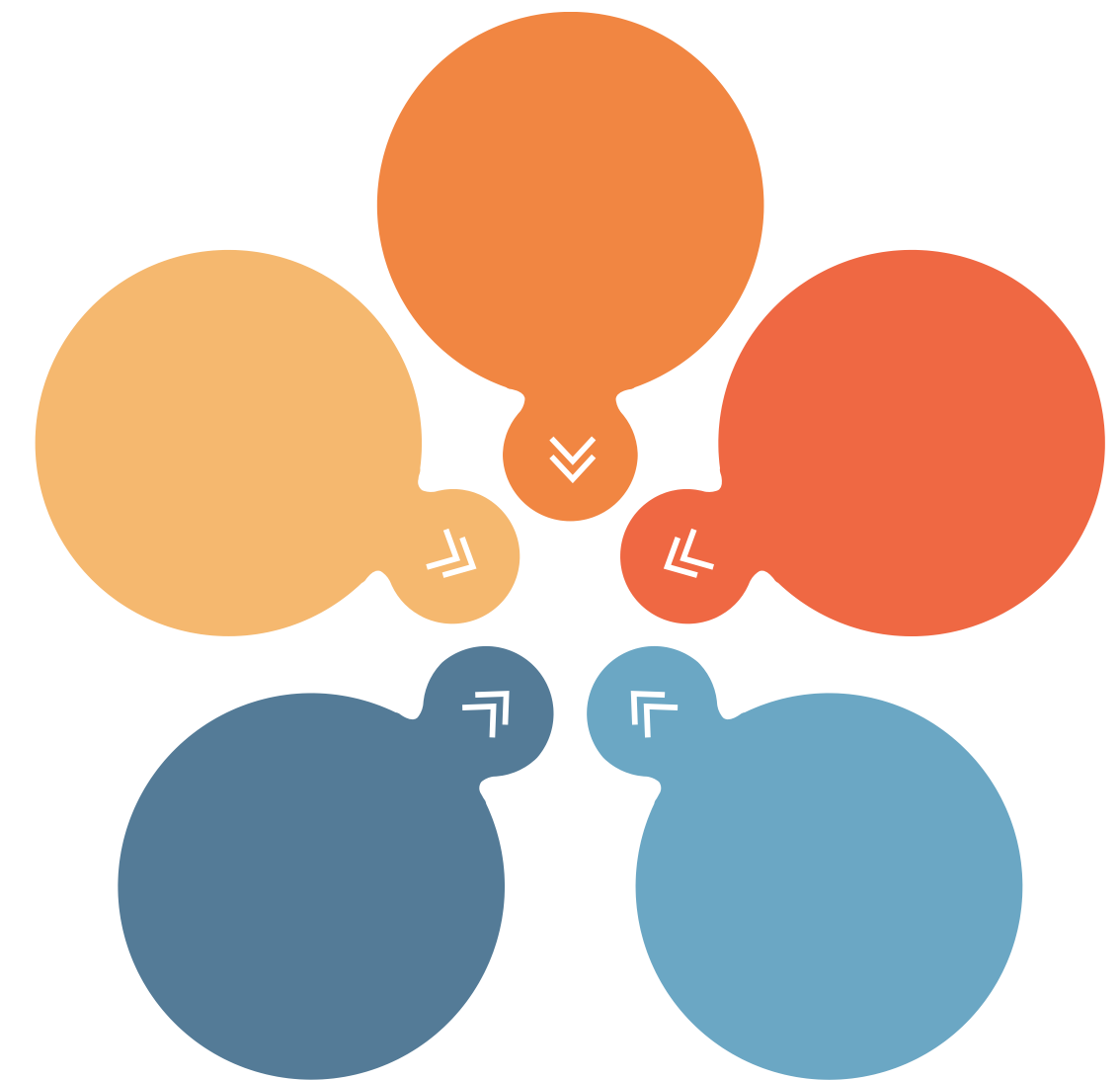
Por ello para determinarlo un buen apoyo es el plano cartesiano, generar puntos en sus esquinas y calcular la distancia que hay entre ellos.



¿Cómo hacerlo?

Para implementar una solución para este reto fue necesario organizar en cinco apartados diferentes.

- Main, centro de comunicaciones.
- Punto, dedicada a generar nuestros puntos y distancias.
- Igualdad, para ver si nuestras distancias son iguales.
- Ventana, dedicada al apartado gráfico
- Mensaje, encargado de informar al usuario de los resultados.



Prueba de escritorio exprés

Entonces ¿Qué es lo que esperaríamos de dos entradas diferentes?

Entrada 1: Mandamos como argumentos de main a la cadena **"0 0 7 5 10 2"**, estos se volverán tipo double que pasaran a formar puntos.

P1(0,0)

P2(7,5)

P3(10,2)

Se calcula la distancia entre ellos con Math.hypot

D1=9.2195

D2= 5.00

D3=10.1980

Se redondea el resultado

Se compara si son iguales: **D1 != D2, D2 != D3, D3 != D1**

Como son diferentes nos arroja False y por ende no es equilátero

Prueba de escritorio exprés

Entrada 2: Mandamos como argumentos de main a la cadena **"0 0 2 0 1 1.732"**, estos se volverán tipo double que pasaran a formar puntos.

P1(0,0)

P2(2,0)

P3(1,1.732)

Se calcula la distancia entre ellos con `Math.hypot`

D1=2.00

D2= 1.9999

D3=1.9999

Se redondea el resultado

Se compara si son iguales: **D1 == D2, D2 == D3, D3 == D1**

Como son iguales nos arroja True y por ende es equilátero

Clase TrianguloEq

```
1  public class TrianguloEq {  
    Run | Debug  
2      public static void main(String[] args){  
3  
4          double x1 = Double.parseDouble(args[0]);  
5          double y1 = Double.parseDouble(args[1]);  
6          double x2 = Double.parseDouble(args[2]);  
7          double y2 = Double.parseDouble(args[3]);  
8          double x3 = Double.parseDouble(args[4]);  
9          double y3 = Double.parseDouble(args[5]);  
10  
11          Punto P1 = new Punto(x1, y1);  
12          Punto P2 = new Punto(x2, y2);  
13          Punto P3 = new Punto(x3, y3);  
14  
15          double distancia1 = P1.distancia(P2);  
16          double distancia2 = P2.distancia(P3);  
17          double distancia3 = P3.distancia(P1);  
18  
19          boolean Equilatero = Igualdad.comparar(distancia1, distancia2, distancia3);  
20  
21          Mensaje controlador = new Mensaje(Equilatero, P1, P2, P3, distancia1, distancia2, distancia3);  
22          Ventana ventana = new Ventana(controlador);  
23          ventana.setVisible(true);  
24      }  
25  }
```

- Método main
- Comunicación con las demás clases

Clase Punto

```
1  public class Punto {  
2      double x;  
3      double y;  
4  
5      public Punto(double x, double y){  
6          this.x = x;  
7          this.y = y;  
8      }  
9  
10     public double distancia(Punto apoyo){  
11         return Math.hypot(apoyo.x - this.x, apoyo.y - this.y);  
12     }  
13  
14     public String toString(){  
15         return "(" + x + "," + y + ")";  
16     }  
17 }
```

- Método constructor, distancia y toString

Clase Igualdad

```
1 public class Igualdad {  
2     public static boolean comparar(double distancia1, double distancia2, double distancia3) {  
3  
4         double d1Redondeado = Math.round(distancia1 * 10.0) / 10.0;  
5         double d2Redondeado = Math.round(distancia2 * 10.0) / 10.0;  
6         double d3Redondeado = Math.round(distancia3 * 10.0) / 10.0;  
7  
8         return (d1Redondeado == d2Redondeado) && (d2Redondeado == d3Redondeado);  
9     }  
10 }
```

- Método comparar

- Redondeo

Clase Ventana

```
1  import javax.swing.*;
2  import java.awt.event.*;
3
4  public class Ventana extends JFrame {
5
6      JButton boton;
7      Mensaje controlador;
8
9      public Ventana(Mensaje controlador){
10         this.controlador = controlador;
11
12         setTitle(title:"Triangulo equilatero");
13         setSize(width:400, height:200);
14         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
15         setLocationRelativeTo(c:null);
16
17         boton = new JButton(text:"Haga clic para ver si el triangulo es equilatero");
18         boton.addActionListener(new ActionListener() {
19             @Override
20             public void actionPerformed(ActionEvent e) {
21                 String mensaje = controlador.mensaje();
22                 JOptionPane.showMessageDialog(parentComponent:null, mensaje);
23             }
24         });
25         add(boton);
26     }
27
28 }
```

- Apartado para generar la ventana gráfica

Clase Mensaje

- Constructor del mensaje
- Formato del mensaje

```
1 public class Mensaje {
2     public boolean Equilatero;
3     public Punto p1, p2, p3;
4     public double distancia1, distancia2, distancia3;
5
6     public Mensaje(boolean Equilatero, Punto p1, Punto p2, Punto p3,
7         double distancia1, double distancia2, double distancia3){
8         this.Equilatero = Equilatero;
9         this.p1 = p1;
10        this.p2 = p2;
11        this.p3 = p3;
12        this.distancia1 = distancia1;
13        this.distancia2 = distancia2;
14        this.distancia3 = distancia3;
15    }
16
17    public String mensaje(){
18        String mensaje = "Puntos: \n";
19        mensaje += "P1: " + p1 + "\n";
20        mensaje += "P2: " + p2 + "\n";
21        mensaje += "P3: " + p3 + "\n\n";
22        mensaje += "Distancias: \n";
23        mensaje += "Distancia entre P1 y P2: " + distancia1 + "\n";
24        mensaje += "Distancia entre P2 y P3: " + distancia2 + "\n";
25        mensaje += "Distancia entre P1 y P3: " + distancia3 + "\n\n";
26
27        if(Equilatero){
28            mensaje += "El triangulo es equilatero";
29        } else {
30            mensaje += "El triangulo no es equilatero";
31        }
32        return mensaje;
33    }
34 }
```

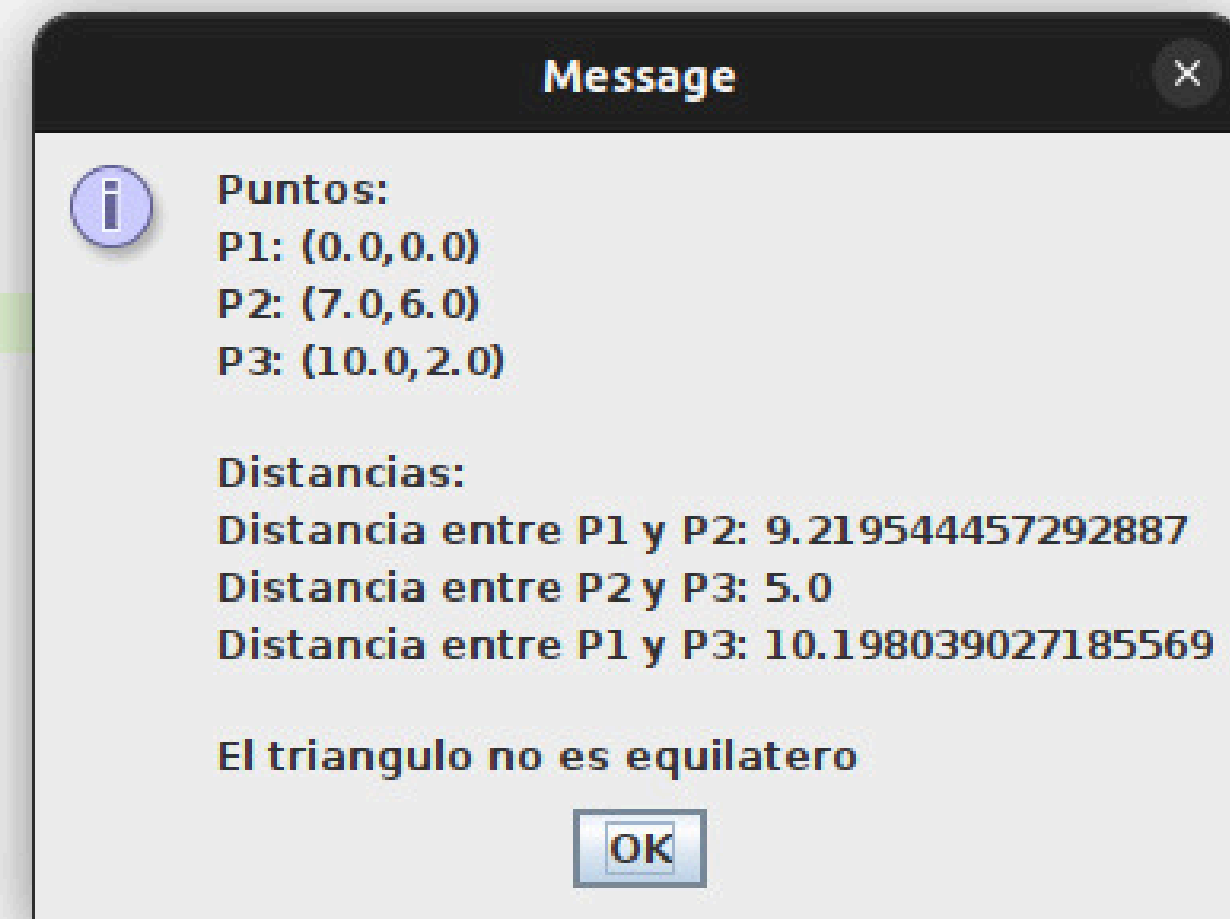
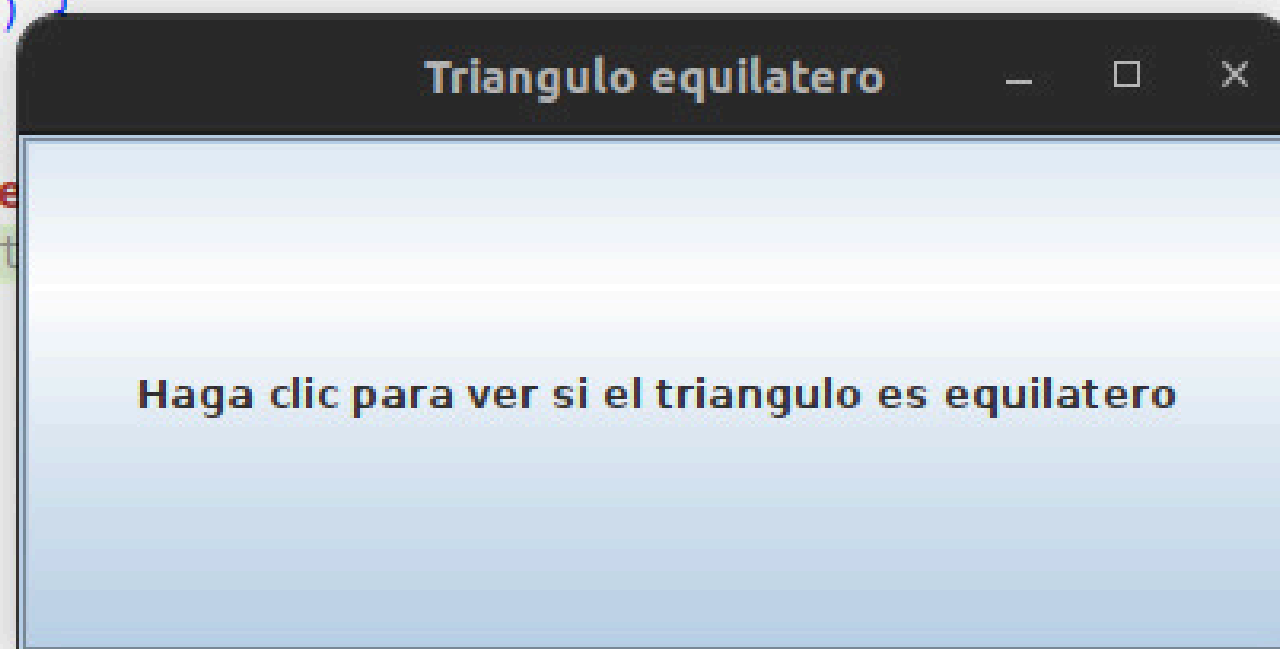
Para cuando no es equilátero

PORTS

```
~/proyectosJava/Reto04.java$ java TrianguloEq 0 0 7 6 10 2
```

```
ver si el triangulo es equilatero");  
er() {
```

```
ent  
saje  
rent
```



Para cuando si es equilátero

PORTS

```
00s:~/proyectosJava/Reto04.java$ java TrianguloEq 0 0 2 0 1 1.732
```

