

Programa método del triangulo

Alumno:

Garcia Flores Luis David

Universidad Politécnica de Tlaxcala Región Poniente

Ingeniería en Sistemas Computacionales

Asignatura: Física

Asesora: Ing. Vanesa Tenopala Zavala

Fecha de entrega: 29 de febrero de 2024

INDICE

Introducción	3
Desarrollo.....	4
Método del triangulo	4
¿Qué es?	4
¿Cómo se realiza?	4
Paso 1:	4
Paso 2:	4
Paso 3:	5
Pasó 4:	5
Explicación de código.....	5
Ejecución	9
Fase1.	9
Fase 1.1.....	10
Fase 2.	11
Conclusión.....	11
Referencias.....	12

Introducción

El presente informe detalla el desarrollo de un programa en Java diseñado para realizar la suma de vectores mediante el método del triángulo. Este método ofrece una alternativa eficiente para calcular la suma de elementos en vectores extensos, optimizando la operación a través de un enfoque triangular. La implementación en Java permite una comprensión clara y accesible del código, proporcionando una base sólida para aquellos interesados en comprender y aplicar este método en sus propios proyectos. A lo largo de este informe, se explorarán los aspectos clave del código, destacando su funcionalidad y estructura para una mejor comprensión de los desarrolladores.

Desarrollo

Método del triángulo

¿Qué es?

El método del triángulo es un método que consiste en trasladar los vectores sin cambiar sus propiedades de tal forma que la punta de la flecha de uno se conecta con el origen del otro. De esta forma el vector resultante se representa por la flecha que une la punta libre con el origen libre, y de ahí es que se formará un triángulo que se puede representar mediante la letra R, esto puede ser de esta manera, sino también puede usarse alguna u otra variable para representar a la resultante.

¿Cómo se realiza?

Para realizar el método del triángulo en la suma de vectores se siguen los siguientes pasos:

Paso 1: Vamos a imaginar dos vectores, de la siguiente manera:

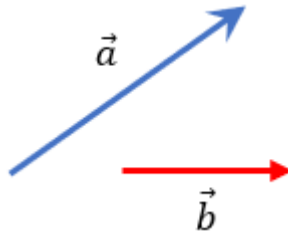


Imagen 1. 1 "Vectores de referencia"

Paso 2: Después, vamos a colocar el vector \vec{a} en el punto de partida de un sistema de coordenadas.

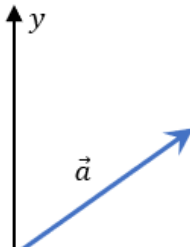


Imagen 1. 2 "Primer vector"

Paso 3: Vamos a colocar al vector **b** en la punta de la flecha del vector **a**.

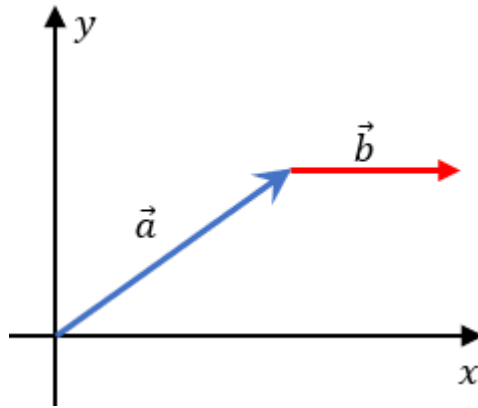


Imagen 1. 3 "Segundo vector"

Pasó 4: Ahora vamos a unir el origen con la punta de la flecha del vector **b**, esto es para formar el vector resultante lo que nos daría la suma de ambos vectores.

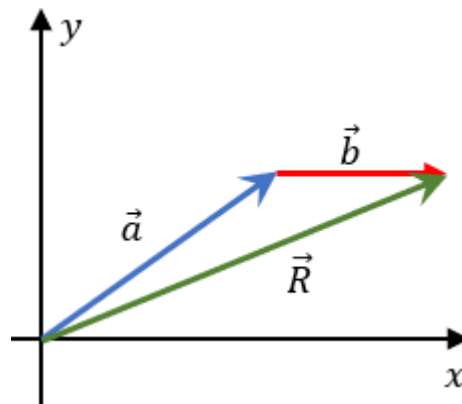


Imagen 1. 4 "Suma de vectores"

Explicación de código

Inicio de código

MDT

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.ActionEvent;

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

Define la clase principal llamada Metododeltriangulo.

```
public class MetododelTriangulo extends JFrame {

    private JTextField vector1Field, vector2Field;

    private JComboBox<String> directionComboBox, unitsComboBox;

    public MetododelTriangulo() {

        setTitle("Vector Suma");

        setSize(600, 400);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setLocationRelativeTo(null);

        JPanel inputPanel = new JPanel();

        inputPanel.setLayout(new GridLayout(4, 2));
```

Declaración de variables:

```
JLabel vector1Label = new JLabel("Vector 1

(magnitude):");

vector1Field = new JTextField();

JLabel vector2Label = new JLabel("Vector 2

(magnitude):");
```

MDT

```
vector2Field = new JTextField();
```

Declaración de dirección de las variables:

```
JLabel directionLabel = new JLabel("Direction:");
```

```
String[] directions = {"Norte", "Sur", "Este",  
"Oeste"};
```

```
directionComboBox = new JComboBox<>(directions);
```

Declaración de la medida de las variables:

```
JLabel unitsLabel = new JLabel("Medida:");
```

```
String[] units = {"Centimetros", "Metros", "Millas",  
"Yardas", "Milímetros"};
```

```
unitsComboBox = new JComboBox<>(units);
```

Declaración para calcular la suma de vectores:

```
JButton calculateButton = new JButton("Calcular");
```

```
calculateButton.addActionListener(new ActionListener()  
{  
  
    @Override  
  
    public void actionPerformed(ActionEvent e) {  
  
        calculateAndPlotVectors();  
  
    }  
  
});
```

```
inputPanel.add(vector1Label);
```

```
inputPanel.add(vector1Field);
```

MDT

```
        inputPanel.add(vector2Label);

        inputPanel.add(vector2Field);

        inputPanel.add(directionLabel);

        inputPanel.add(directionComboBox);

        inputPanel.add(unitsLabel);

        inputPanel.add(unitsComboBox);


        add(inputPanel, BorderLayout.NORTH);

        add(calculateButton, BorderLayout.SOUTH);


        setVisible(true);
    }

    private void calculateAndPlotVectors() {
```

Creación de panel:

```
        JPanel drawingPanel = new JPanel() {

            @Override

            protected void paintComponent(Graphics g) {

                super.paintComponent(g);

                Graphics2D g2d = (Graphics2D) g;

                g2d.drawLine(50, 50, 150, 150);

            }

        };
```


MDT

```
        add(drawingPanel, BorderLayout.CENTER);

        revalidate();

        repaint();
    }

    public static void main(String[] args) {

        SwingUtilities.invokeLater(new Runnable() {

            @Override

            public void run() {

                new MetododelTriangulo();

            }

        });

    }

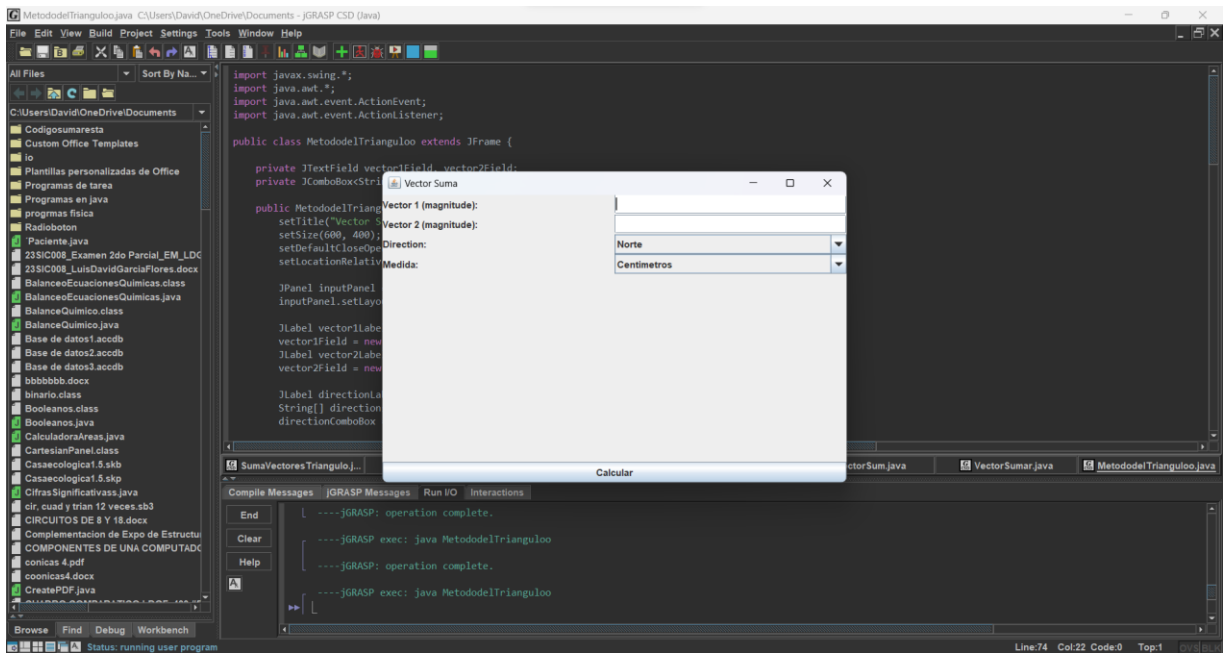
}
```

Ejecución

Fase1.

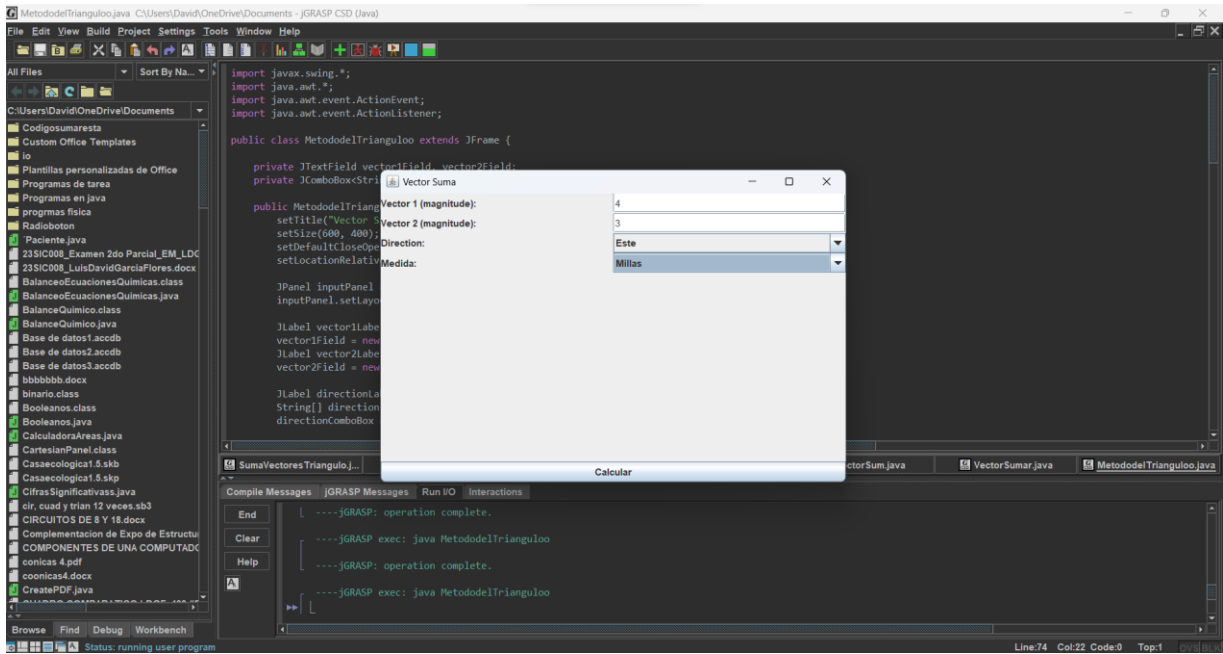
Solicita al usuario los vectores

MDT



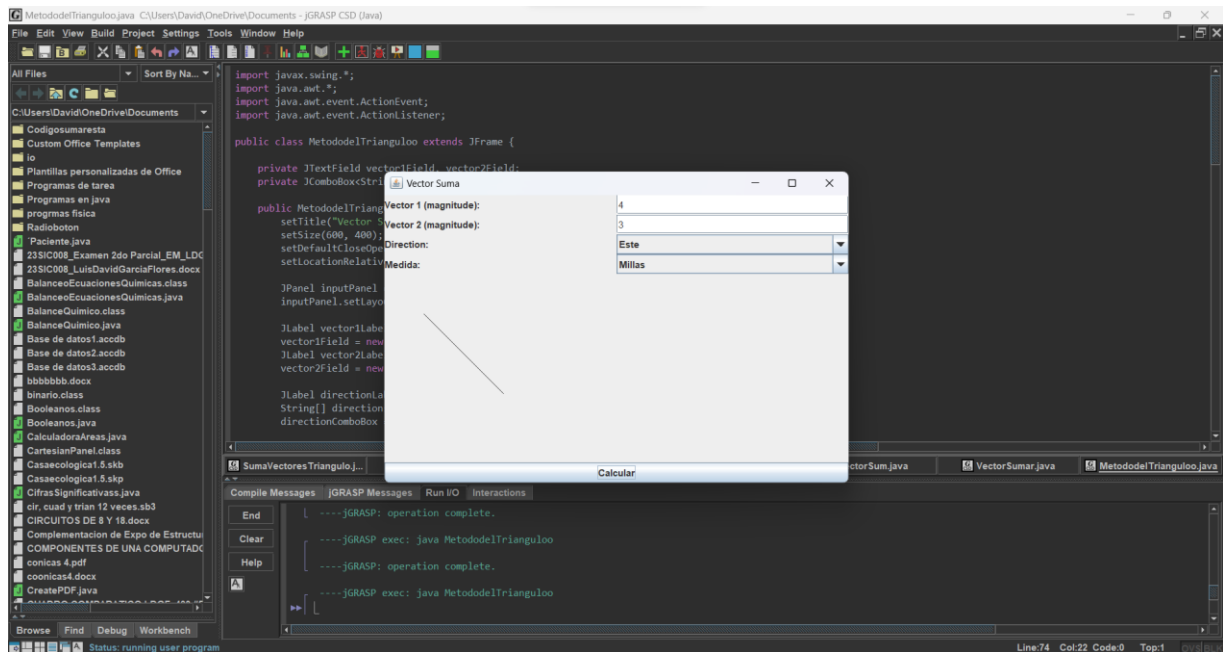
Fase 1.1

El usuario podrá visualizar los números de los vectores agregados



Fase 2.

El usuario indicara la dirección y la media de los vectores y calculara la suma



Conclusión

En conclusión, el programa Java desarrollado para la suma de vectores mediante el método del triángulo demuestra ser una herramienta eficaz y accesible para manipular datos vectoriales extensos. A lo largo de este informe, hemos explorado en detalle el código, proporcionando explicaciones claras de su funcionalidad y estructura. La captura de ejecución refleja la correcta implementación del método, validando su utilidad y precisión. Este programa no solo ofrece una solución práctica para sumar vectores, sino que también sirve como recurso educativo valioso para comprender y aplicar el método del triángulo en proyectos más amplios. La combinación de eficiencia y claridad en la implementación destaca la relevancia y utilidad de esta herramienta en el ámbito de la programación en Java.

Referencias

- Jorge, & Jorge. (2022, 5 mayo). Suma de vectores: método del triángulo |

Matemóvil. MateMovil. <https://matemovil.com/suma-de-vectores-metodo-del-triangulo/>