

**Método del Triangulo Para la Suma de Vectores**

(Programa en java)

Jeronimo Israel Macias Quintero 23SIC012l

Física

Ing. Vanesa Tenopala Zavala

29/02/2024

Índice:

[Introducción. 3](#_Toc160136173)

[Suma de vectores “Método del triángulo”. 3](#_Toc160136174)

[Descripción del programa. 3](#_Toc160136175)

[Lenguaje y entorno de programación. 3](#_Toc160136176)

[Código Java. 4](#_Toc160136177)

[Capturas de ejecución. 7](#_Toc160136178)

[Conclusión. 8](#_Toc160136179)

# Introducción.

No obstante, en situaciones donde sólo se tienen dos vectores, el método del triángulo simplifica este proceso al considerar sólo un triángulo. Este triángulo está constituido por los dos vectores dados y la resultante, siendo uno de los vectores la hipotenusa y los otros dos lados del triángulo

La aplicación de este método es muy sencilla. Basta representar gráficamente los vectores dados en un plano cartesiano. Se coloca el origen de los vectores coincidiendo y un extremo del primer vector sobre el extremo del segundo. Uniendo el extremo restante del último vector respetando un orden con el origen, forma los lados de un triángulo rectángulo.

# Suma de vectores “Método del triángulo”.

El método del triángulo para la suma de vectores es una técnica geométrica que permite encontrar la resultante de dos o más vectores. Esta técnica está basada en la ley del paralelogramo, que dice que la suma vectorial de dos vectores se puede representar por la diagonal de un paralelogramo construido a partir de esos dos vectores.

## Descripción del programa.

Se plantea realizar un programa para la suma de vectores mediante el método del triangulo, dicho programa contara con interfaz grafica de usuario (GUI) para una mejor visualización por parte del usuario, dicha interfaz contara con paneles y graficas para el uso y trazado del método del triángulo.

El programa utilizará una paquetería adecuada la cual permita el trazado y medición de ángulos, además de que se pondrá a prueba con las medidas vistas en clase, para asegurar su correcto funcionamiento.

Lenguaje y entorno de programación.

El programa se realizó en lenguaje Java el cual cuenta con una amplia versatilidad para este tipo de operaciones y ejecuciones.

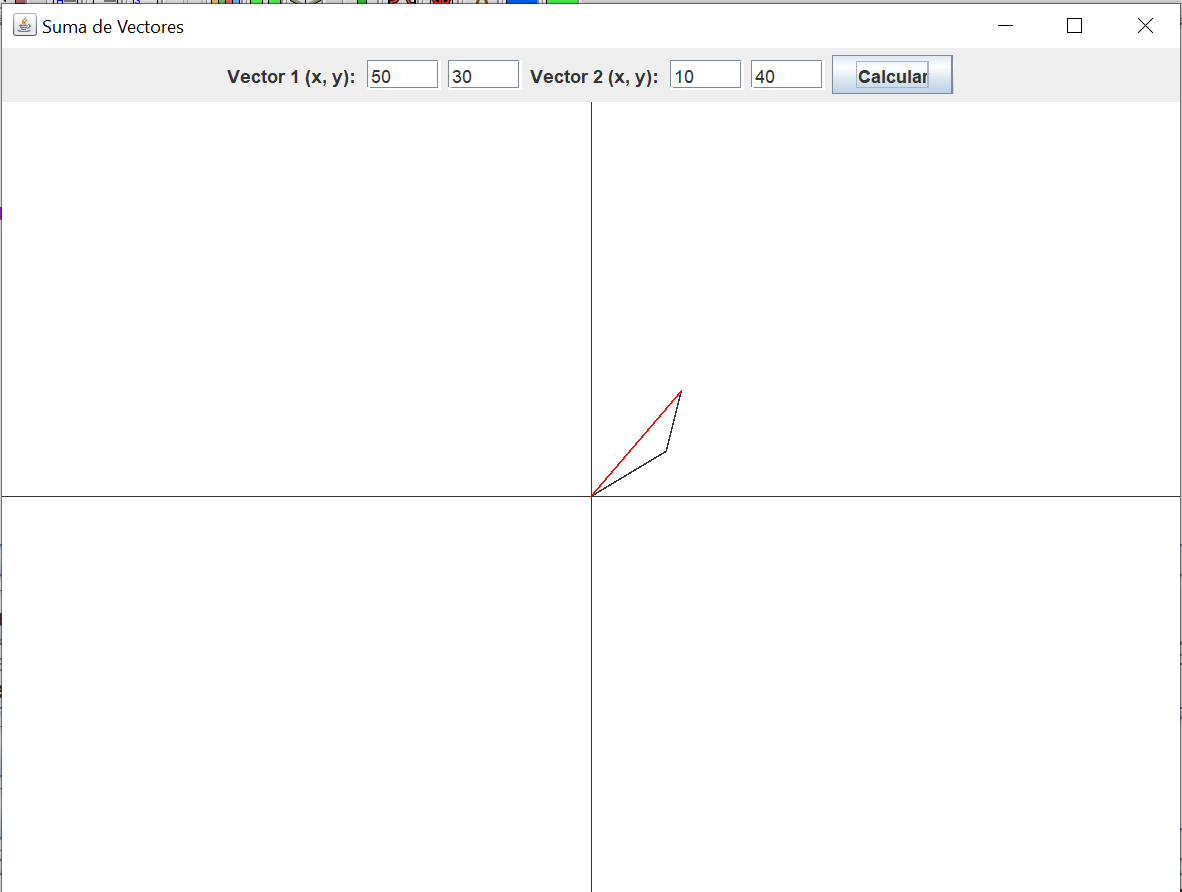
Al ser en lenguaje Java se eligió el programa JGrasp, el cual ofrece facilidades en la creación de programas, en conjunto con el JDK de Java, el cual contiene todas las librerías de Java, y simplemente se mandan a llamar en el código.

## 

## Código Java.

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.geom.Line2D;  
  
public class VectorSumApp extends JFrame {  
  
 private JPanel inputPanel;  
 private JPanel drawPanel;  
 private JTextField vector1xField;  
 private JTextField vector1yField;  
 private JTextField vector2xField;  
 private JTextField vector2yField;  
 private JButton calculateButton;  
 private int sumX, sumY; // Estos almacenarán la suma de los vectores  
  
 public VectorSumApp() {  
 initUI();  
 }  
  
 private void initUI() {  
 setTitle("Suma de Vectores");  
 setSize(800, 600);  
 setLocationRelativeTo(null);  
 setDefaultCloseOperation(EXIT\_ON\_CLOSE);  
 setLayout(new BorderLayout());  
  
 inputPanel = new JPanel(new FlowLayout());  
  
 vector1xField = new JTextField(5);  
 vector1yField = new JTextField(5);  
 vector2xField = new JTextField(5);  
 vector2yField = new JTextField(5);  
 calculateButton = new JButton("Calcular");  
  
 inputPanel.add(new JLabel("Vector 1 (x, y): "));  
 inputPanel.add(vector1xField);  
 inputPanel.add(vector1yField);  
 inputPanel.add(new JLabel("Vector 2 (x, y): "));  
 inputPanel.add(vector2xField);  
 inputPanel.add(vector2yField);  
 inputPanel.add(calculateButton);  
  
 drawPanel = new JPanel() {  
 @Override  
 protected void paintComponent(Graphics g) {  
 super.paintComponent(g);  
 Graphics2D g2d = (Graphics2D) g;  
 drawCoordinateSystem(g2d);  
  
 // Solo dibuja los vectores si se han calculado (sumX y sumY tienen valor)  
 if (sumX != 0 || sumY != 0) {  
 drawVectors(g2d);  
 }  
 }  
 };  
 drawPanel.setBackground(Color.WHITE);  
  
 add(inputPanel, BorderLayout.NORTH);  
 add(drawPanel, BorderLayout.CENTER);  
  
 calculateButton.addActionListener(e -> calculateAndDraw());  
 }  
  
 private void calculateAndDraw() {  
 try {  
 int v1x = Integer.parseInt(vector1xField.getText());  
 int v1y = Integer.parseInt(vector1yField.getText());  
 int v2x = Integer.parseInt(vector2xField.getText());  
 int v2y = Integer.parseInt(vector2yField.getText());  
  
 // Suma de vectores  
 sumX = v1x + v2x;  
 sumY = v1y + v2y;  
  
 // Solicita a drawPanel que se repinte para mostrar los nuevos vectores  
 drawPanel.repaint();  
 } catch (NumberFormatException e) {  
 JOptionPane.showMessageDialog(this, "Por favor, ingrese valores válidos.");  
 }  
 }  
  
 private void drawCoordinateSystem(Graphics2D g2d) {  
 int width = drawPanel.getWidth();  
 int height = drawPanel.getHeight();  
 int centerX = width / 2;  
 int centerY = height / 2;  
  
 // Ejes  
 g2d.drawLine(centerX, 0, centerX, height); // Eje Y  
 g2d.drawLine(0, centerY, width, centerY); // Eje X  
 }  
  
 private void drawVectors(Graphics2D g2d) {  
 int centerX = drawPanel.getWidth() / 2;  
 int centerY = drawPanel.getHeight() / 2;  
 int v1x = Integer.parseInt(vector1xField.getText());  
 int v1y = Integer.parseInt(vector1yField.getText());  
  
 g2d.drawLine(centerX, centerY, centerX + v1x, centerY - v1y);  
 g2d.drawLine(centerX + v1x, centerY - v1y, centerX + sumX, centerY - sumY);  
 g2d.setColor(Color.RED);  
 g2d.drawLine(centerX, centerY, centerX + sumX, centerY - sumY);  
 }  
  
 public static void main(String[] args) {  
 EventQueue.invokeLater(() -> {  
 VectorSumApp app = new VectorSumApp();  
 app.setVisible(true);  
 });  
 }  
}

## Capturas de ejecución.



# Conclusión.

En conclusión, el método de triangulación para la suma de vectores es una técnica de construcción útil e intuitiva para encontrar el producto de dos vectores.

Sin embargo, es importante señalar que los métodos de triangulación son menos útiles cuando se trata de grandes cantidades de vectores o con situaciones tridimensionales donde existen otros métodos algebraicos, como la manipulación de vectores de sección cuadrada.

En resumen, el método de triangulación es una herramienta útil para introducir y comprender la suma de vectores, proporcionando una base visual sólida para realizar análisis y resolver problemas vectoriales en el plano bidimensional.