

UNIVERSIDAD POLITÉCNICA DE TLAXCALA  
REGIÓN PONIENTE

INGENIERÍA EN SISTEMAS COMPUTACIONALES

*Física*

*Reporte de Programa De Vectores con  
Magnitud escalar*

DOCENTE: ING. Vanesa Tenopala Zavala

ESTUDIANTE: Gaston Amisael Zavala Zavala

24/01/2024



## Índice

<b>Introducción</b>	<b>3</b>
<b>Desarrollo</b>	<b>4</b>
Código	4
Ejecución del programa	6
Características	7
<b>Conclusión</b>	<b>7</b>

## Índice de imágenes

<b>Imagen 1. Primeras líneas de código</b>	<b>6</b>
<b>Imagen 2. Más líneas de código</b>	<b>6</b>
<b>Imagen 3. Programa ejecutado</b>	<b>6</b>



## Introducción

La representación gráfica de vectores en un plano cartesiano desempeña un papel crucial. La capacidad de comprender y calcular la distancia y dirección de estos vectores no solo es esencial para los estudios académicos, sino que también encuentra aplicaciones prácticas en diversas disciplinas, desde la ingeniería hasta la informática.

Con el propósito de simplificar y potenciar esta tarea, se creó este programa diseñado para calcular vectores en un plano cartesiano, teniendo en cuenta tanto su magnitud como su dirección. Este programa no solo brinda una solución eficiente a los desafíos matemáticos asociados con los vectores, sino que también permite visualizar de manera intuitiva cómo interactúan en el espacio.

Este programa se convierte en una herramienta invaluable para estudiantes, profesionales y entusiastas de las ciencias, proporcionando una plataforma donde se pueden explorar y entender conceptos fundamentales como la distancia euclidiana y la orientación angular de los vectores en el plano. A través de una interfaz amigable e interactiva, los usuarios pueden ingresar datos específicos y obtener resultados precisos, facilitando así el proceso de aprendizaje y aplicación práctica de estos conceptos.

A continuación se explicará adecuadamente cada una de las funciones que realiza el programa.



## Desarrollo

A medida que profundizamos en sus funcionalidades, quedará claro que este programa no solo simplifica la manipulación de vectores en un plano cartesiano, sino que también abre nuevas puertas para la exploración y comprensión de las magnitudes y direcciones en el espacio bidimensional.

## Código

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

public class SumaVectores extends JFrame {
    private JPanel panel;
    private JTextField vector1XField, vector1YField, vector2XField, vector2YField;
    private JButton sumButton;

    private int vector1X, vector1Y, vector2X, vector2Y;

    public SumaVectores() {
        super("Suma de Vectores");

        panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                Graphics2D g2d = (Graphics2D) g;

                // Dibujar los ejes coordenados
                g2d.setColor(Color.BLACK);
                g2d.drawLine(100, 300, 500, 300); // eje x
                g2d.drawLine(300, 100, 300, 500); // eje y

                // Dibujar los vectores
                drawVector(g2d, 300, 300, vector1X, vector1Y, "A");
                drawVector(g2d, 300, 300, vector2X, vector2Y, "B");
                drawVector(g2d, 300, 300, vector1X + vector2X, vector1Y + vector2Y, "C");
            }
        };

        private void drawVector(Graphics2D g2d, int startX, int startY, int dx, int dy, String label) {
            int endX = startX + dx;
            int endY = startY - dy; // Invertimos la coordenada y porque en pantalla aumenta hacia
            // abajo

            g2d.setColor(Color.BLUE);
            g2d.drawLine(startX, startY, endX, endY); // Dibujar el vector
            g2d.fill(new Ellipse2D.Double(endX - 3, endY - 3, 6, 6)); // Dibujar el punto final del vector
            g2d.drawString(label, endX + 5, endY - 5); // Etiqueta del vector
        }
    }
}
```



```
};

vector1XField = new JTextField(5);
vector1YField = new JTextField(5);
vector2XField = new JTextField(5);
vector2YField = new JTextField(5);
sumButton = new JButton("Sumar");

sumButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            vector1X = Integer.parseInt(vector1XField.getText());
            vector1Y = Integer.parseInt(vector1YField.getText());
            vector2X = Integer.parseInt(vector2XField.getText());
            vector2Y = Integer.parseInt(vector2YField.getText());
            panel.repaint();
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(null, "Ingrese valores numéricos válidos para los
vectores.");
        }
    }
});

JPanel inputPanel = new JPanel();
inputPanel.add(new JLabel("Vector 1 (x, y):"));
inputPanel.add(vector1XField);
inputPanel.add(new JLabel(", "));
inputPanel.add(vector1YField);
inputPanel.add(new JLabel("Vector 2 (x, y):"));
inputPanel.add(vector2XField);
inputPanel.add(new JLabel(", "));
inputPanel.add(vector2YField);
inputPanel.add(sumButton);

getContentPane().setLayout(new BorderLayout());
getContentPane().add(panel, BorderLayout.CENTER);
getContentPane().add(inputPanel, BorderLayout.SOUTH);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(600, 600);
setLocationRelativeTo(null);
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        SumaVectores sumaVectores = new SumaVectores();
        sumaVectores.setVisible(true);
    });
}
```



## Ejecución del programa

```
SumaVectores.java | X
C:\Users\52241> OneDrive\Documents\Física> SumaVectores.java SumaVectores
1 import javax.swing.*; SumaVectores.java is a non-project file, only syntax errors are reported
2 import java.awt.*;
3 import java.awt.event.*;
4 import java.awt.geom.*;
5
6 public class SumaVectores extends JFrame {
7     private JPanel panel;
8     private JTextField vector1XField, vector1YField, vector2XField, vector2YField;
9     private JButton sumButton;
10
11     private int vector1X, vector1Y, vector2X, vector2Y;
12
13     public SumaVectores() {
14         super(title:"Suma de Vectores");
15
16         panel = new JPanel() {
17             @Override
18             protected void paintComponent(Graphics g) {
```

Imagen 1. Primeras líneas de código

```
SumaVectores.java | X
C:\Users\52241> OneDrive\Documents\Física> SumaVectores.java SumaVectores
6 public class SumaVectores extends JFrame {
13     public SumaVectores() {
16         panel = new JPanel() {
18             protected void paintComponent(Graphics g) {
19
22                 // Dibujar los ejes de coordenadas
23                 g2d.setColor(Color.BLACK);
24                 g2d.drawLine(x1:100, y1:300, x2:500, y2:300); // eje x
25                 g2d.drawLine(x1:300, y1:100, x2:300, y2:500); // eje y
26
27                 // Dibujar los vectores
28                 drawVector(g2d, startX:300, startY:300, vector1X, vector1Y, label:"A");
29                 drawVector(g2d, startX:300, startY:300, vector2X, vector2Y, label:"B");
30                 drawVector(g2d, startX:300, startY:300, vector1X + vector2X, vector1Y + vector2Y, label:"C");
31             }
32
33             private void drawVector(Graphics2D g2d, int startX, int startY, int dx, int dy, String label) {
34                 int endX = startX + dx;
35                 int endY = startY - dy; // Invertimos la coordenada y porque en pantalla aumenta hacia abajo
```

Imagen 2. Más líneas de código

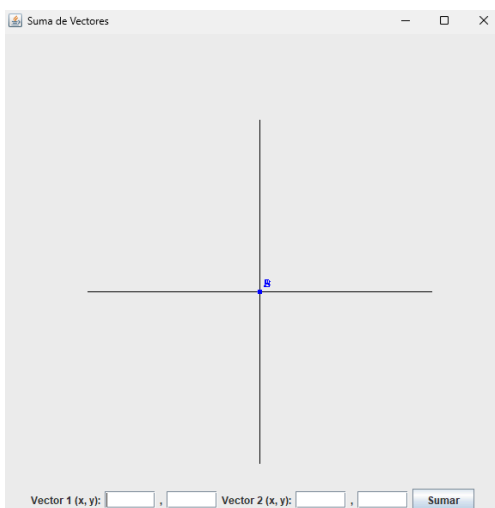


Imagen 3. Programa ejecutado



## Características

Este código Java crea una interfaz gráfica de usuario (GUI) para sumar vectores en un plano cartesiano.

- ★ Clase principal (suma de vectores)
- ★ Componentes de entrada
- ★ Manejador de eventos
- ★ Metodo main
- ★ Librerías

## Conclusión

Este programa se convierte en una herramienta invaluable para estudiantes, profesionales y entusiastas de las ciencias, proporcionando una plataforma donde se pueden explorar y entender conceptos fundamentales como la distancia euclidiana y la orientación angular de los vectores en el plano. A través de una interfaz amigable e interactiva, los usuarios pueden ingresar datos específicos y obtener resultados precisos, facilitando así el proceso de aprendizaje y aplicación práctica de estos conceptos.

## Referencias

*Vectores: qué son, características, tipos - Ferrovial.* (2024, 29 febrero). Ferrovial.  
<https://www.ferrovial.com/es/stem/vectores/>

Equipo editorial, Etecé. (2024, 29 febrero). *Vector - Qué es, definición, características, tipos y ejemplos.* Concepto. <https://concepto.de/vector/>

*Vector.* (s. f.). Concepto De - Definición. (2024, 29 febrero).  
<https://conceptodefinicion.de/vector/>