



UNIVERSIDAD POLITÉCNICA DE TLAXCALA REGIÓN PONIENTE

INGENIERÍA EN SISTEMAS COMPUTACIONALES

*FISICA*

REPORTE DE CÓDIGO

DOCENTE: ING. Vanesa Tenopala Zavala

ALUMNO: JUAN PABLO GONZALEZ ARAUZ

MATRÍCULA:23SIC023

CUATRIMESTRE: 2° GRUPO: “A”

FECHA: 17/01/2024



# INTRODUCCIÓN

La suma de vectores es una operación fundamental en matemáticas y física que nos permite combinar las magnitudes y direcciones de diferentes vectores para obtener un vector resultante. Uno de los métodos más comunes para realizar esta suma es el método del triángulo, que es especialmente útil cuando trabajamos con vectores representados en un plano cartesiano.

En el método del triángulo, primero ubicamos los vectores en el plano cartesiano, asignándoles coordenadas en función de sus componentes en los ejes x e y. Cada vector se representa como una flecha que parte del origen (0,0) y llega hasta el punto definido por sus coordenadas. Para sumar dos vectores utilizando este método, colocamos el origen del segundo vector en el extremo del primero, de manera que formen dos lados consecutivos de un triángulo.

El tercer lado del triángulo, que va desde el origen del primer vector hasta el extremo del segundo, representa el vector resultante de la suma. Luego, podemos encontrar las coordenadas del vector resultante utilizando la geometría del triángulo formado. Esto se puede hacer aplicando el teorema de Pitágoras para encontrar la magnitud del vector resultante y luego utilizando trigonometría para determinar su dirección. Finalmente, una vez que hemos encontrado las coordenadas del vector resultante, podemos interpretar su magnitud como la longitud y su dirección como el ángulo que forma con el eje x positivo.

Este método proporciona una forma visual e intuitiva de sumar vectores en un plano cartesiano, lo que facilita su comprensión y aplicación en una variedad de contextos, desde la física hasta la ingeniería y más allá.

# ÍNDICE

[INTRODUCCIÓN 1](#_gjdgxs)

[ÍNDICE 2](#_30j0zll)

[DESARROLLO 3](#_1fob9te)

[CÓDIGO 4](#_3znysh7)

# DESARROLLO

* Representación de vectores en el plano cartesiano: Antes de sumar los vectores, es crucial entender cómo se representan en el plano cartesiano. Cada vector se puede describir por dos componentes: una en la dirección del eje x (horizontal) y otra en la dirección del eje y (vertical). Estos componentes determinan las coordenadas del vector en el plano.
* Visualización de los vectores: Dibujar los vectores en un plano cartesiano ayuda a visualizar su magnitud y dirección. Cada vector se traza desde el origen hasta su punto final, utilizando sus componentes para determinar su posición.
* Suma de vectores utilizando el método del triángulo: Para sumar dos vectores, colocamos el origen del segundo vector en el extremo del primero. Esto crea un triángulo en el plano cartesiano, donde los dos vectores originales son dos lados del triángulo, y el vector resultante es el tercer lado.
* Cálculo del vector resultante: Una vez que tenemos el triángulo formado por los vectores originales, podemos usar el teorema de Pitágoras para encontrar la magnitud del vector resultante y trigonometría para encontrar su dirección.
* Interpretación de los resultados: Después de calcular las coordenadas, la magnitud y el ángulo del vector resultante, podemos interpretar estos resultados físicamente. La magnitud representa la longitud del vector resultante, mientras que el ángulo indica su dirección con respecto al eje x.

# CÓDIGO

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.canvas.Canvas;

import javafx.scene.canvas.GraphicsContext;

import javafx.scene.layout.StackPane;

import javafx.stage.Stage;

import java.util.Scanner;

public class MetodoDelTriangulo extends Application {

@Override

public void start(Stage primaryStage) {

// Solicitar al usuario las coordenadas y la unidad de medida

Scanner scanner = new Scanner(System.in);

System.out.println("Ingrese las coordenadas del vector en formato 'x y', donde x es la coordenada horizontal (oeste/negativo, este/positivo) y y es la coordenada vertical (sur/negativo, norte/positivo):");

double startX = scanner.nextDouble();

double startY = scanner.nextDouble();

System.out.println("Ingrese la dirección cardinal de la orientación del vector (norte, sur, este, oeste):");

String orientacion = scanner.next().toLowerCase();

System.out.println("Ingrese la longitud del vector:");

double longitud = scanner.nextDouble();

System.out.println("Ingrese la unidad de medida (metros, kilómetros, yardas, millas, pies):");

String unidad = scanner.next().toLowerCase();

// Convertir la unidad de medida a metros (sistema internacional)

switch (unidad) {

case "metros":

break;

case "kilómetros":

longitud \*= 1000; // 1 kilómetro = 1000 metros

break;

case "millas":

longitud \*= 1609.34; // 1 milla = 1609.34 metros

break;

case "yardas":

longitud \*= 0.9144; // 1 yarda = 0.9144 metros

break;

case "pies":

longitud \*= 0.3048; // 1 pie = 0.3048 metros

break;

default:

System.out.println("Unidad de medida no válida. Se asumirá que la longitud está en metros.");

break;

}

// Ajustar las coordenadas según la orientación del vector

switch (orientacion) {

case "norte":

startY -= longitud;

break;

case "sur":

startY += longitud;

break;

case "este":

startX += longitud;

break;

case "oeste":

startX -= longitud;

break;

default:

System.out.println("Orientación no válida. No se realizarán ajustes a las coordenadas.");

break;

}

// Definir el tamaño del lienzo

int width = 400;

int height = 400;

// Crear el lienzo

Canvas canvas = new Canvas(width, height);

GraphicsContext gc = canvas.getGraphicsContext2D();

// Dibujar los ejes cartesianos

drawAxes(gc, width, height);

// Dibujar el vector ingresado por el usuario

drawVector(gc, startX, startY, startX, startY + longitud);

// Crear la escena

StackPane root = new StackPane();

root.getChildren().add(canvas);

Scene scene = new Scene(root, width, height);

primaryStage.setTitle("Método del Triángulo para Suma de Vectores");

primaryStage.setScene(scene);

primaryStage.show();

}

// Método para dibujar los ejes cartesianos

private void drawAxes(GraphicsContext gc, int width, int height) {

gc.strokeLine(0, height / 2, width, height / 2); // Eje X

gc.strokeLine(width / 2, 0, width / 2, height); // Eje Y

}

// Método para dibujar un vector en el lienzo

private void drawVector(GraphicsContext gc, double startX, double startY, double endX, double endY) {

gc.strokeLine(startX, startY, endX, endY); // Dibujar el vector

gc.strokeOval(endX - 2, endY - 2, 4, 4); // Dibujar el extremo del vector

}

public static void main(String[] args) {

launch(args);

}

}

# Referencias

Hall, K. R., & Saxon, A. (2009). "Algebra 2: Solutions Key." Houghton Mifflin Harcourt.

Larson, R., & Boswell, L. (2017). "Calculus." Cengage Learning.

Young, H. D., & Freedman, R. A. (2016). "University Physics with Modern Physics." Pearson.

Stewart, J. (2015). "Calculus: Early Transcendentals." Cengage Learning.

Leithold, L. (1996). "The Calculus 7." HarperCollins College Publishers.