

*Física*

*Programa de Vectores*

Gastón Amisael Zavala Zavala

23SIC005

ING. Sistemas Computacionales

Ing. Vanesa Tenopala Zavala

Universidad Politécnica de Tlaxcala Región Poniente

Fecha: 16/02/2024

Índice

[Introducción 2](#_Toc1235014493)

[Desarrollo 3](#_Toc463100706)

[Objetivo 3](#_Toc1770106468)

[Características 3](#_Toc1499975878)

[Líneas del código 4](#_Toc1071247940)

[Ejecución del programa 6](#_Toc1651701903)

[Mejoras futuras 7](#_Toc642821987)

[Conclusión 7](#_Toc1778624465)

# Introducción

Un vector, en términos informáticos, se define como una secuencia ordenada de elementos que comparten un mismo tipo de datos. Los vectores son fundamentales en la representación y manipulación de información en muchos campos, desde la simulación numérica hasta la computación gráfica. El programa desarrollado en Java se enfoca en la implementación de algoritmos que permiten realizar operaciones fundamentales con vectores, como la suma y/o la resta.

A lo largo de esta documentación, exploraremos las funciones clave del programa, sus características de diseño, y proporcionaremos ejemplos prácticos de su implementación.

# Desarrollo

## Objetivo

Con este programa en Java, se pretende facilitar y potenciar la resolución de problemas matemáticos, consolidando a Java como una opción destacada en el desarrollo de aplicaciones robustas y eficientes para la manipulación de datos vectoriales.

## Características

**Tipos de datos para vectores:**

* En lugar de usar **String** para representar los elementos de los vectores, considera utilizar tipos numéricos (**int**, **double**, etc.) si el objetivo es realizar operaciones matemáticas. Esto permitirá realizar operaciones aritméticas directamente.

**Operación de resta:**

* En la operación de resta, estás concatenando los elementos del vector A y B con un guion ("-"). Para realizar la resta, debes restar los valores numéricos correspondientes.

**Entrada de datos:**

* Añadir mensajes descriptivos en la entrada de datos puede mejorar la experiencia del usuario. Por ejemplo, se puede indicar que se espera un número entero al solicitar el tamaño de los vectores.

**Manejo de errores:**

* Muestra el error al ingresar los datos para resolver la operación.

## Líneas del código

import java.util.Scanner;

import java.util.Scanner;

public class VectorOperations {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Tamaño de los vectores

System.out.print("Ingrese el tamaño de los vectores: ");

int n = scanner.nextInt();

// Declaración de vectores

String[] vectorA = new String[n];

String[] vectorB = new String[n];

// Ingreso de elementos para el vector A

System.out.println("Ingrese los elementos del vector A:");

for (int i = 0; i < n; i++) {

System.out.print("Elemento " + (i + 1) + ": ");

vectorA[i] = scanner.next();

}

// Ingreso de elementos para el vector B

System.out.println("Ingrese los elementos del vector B:");

for (int i = 0; i < n; i++) {

System.out.print("Elemento " + (i + 1) + ": ");

vectorB[i] = scanner.next();

}

// Suma de vectores

String[] suma = new String[n];

for (int i = 0; i < n; i++) {

suma[i] = vectorA[i] + vectorB[i];

}

// Resta de vectores

String[] resta = new String[n];

for (int i = 0; i < n; i++) {

resta[i] = vectorA[i] + "-" + vectorB[i];

}

// Mostrar resultados

System.out.println("\nResultado de la suma:");

mostrarVector(suma);

System.out.println("\nResultado de la resta:");

mostrarVector(resta);

scanner.close();

}

// Método para mostrar un vector

private static void mostrarVector(String[] vector) {

System.out.print("[ ");

for (int i = 0; i < vector.length; i++) {

System.out.print(vector[i]);

if (i < vector.length - 1) {

System.out.print(", ");

}

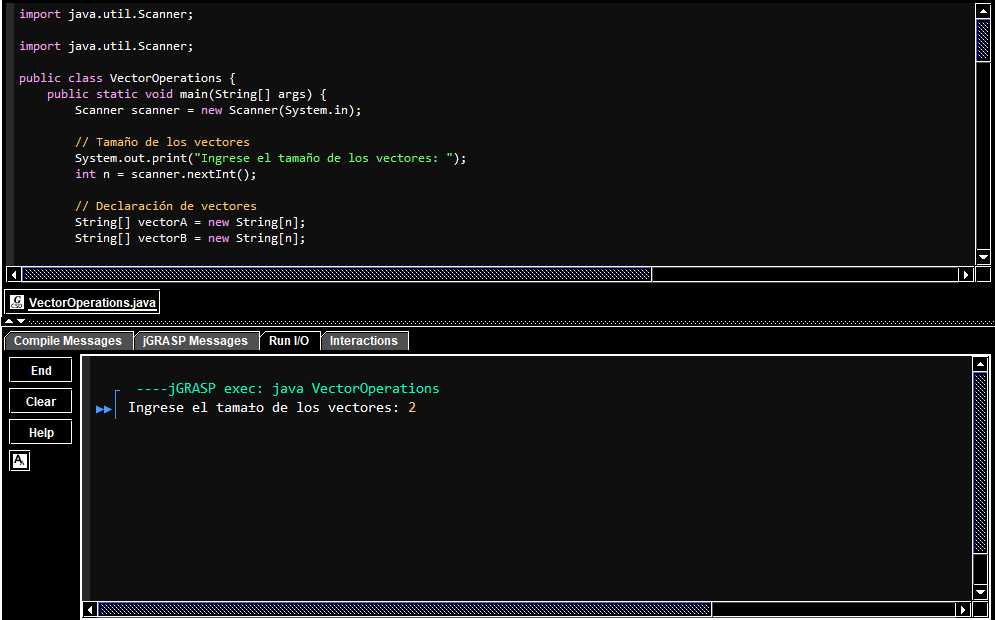
}

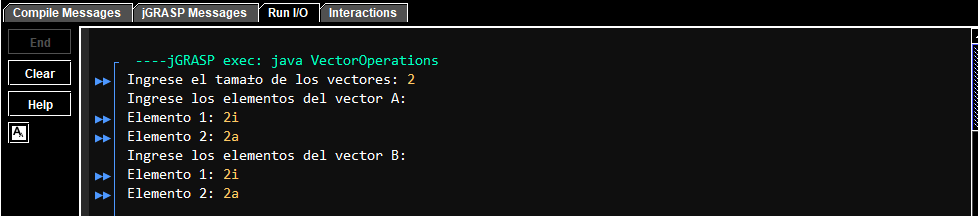
System.out.println(" ]");

}

}

## Ejecución del programa

El programa inicia preguntando, ¿Cuál es el tamaño para la cantidad de vectores?

Se tendrá que ingresar los elemento para la operación.

Al final mostrara el resultado de la suma y resta de vectores.

## Mejoras futuras

**Orientación a objetos:**

* Se considera organizar el código utilizando conceptos de programación orientada a objetos, creando una clase Vector con métodos para realizar las operaciones.

**Modularidad:**

* Diseño modular que permita fácilmente la extensión del programa para incluir operaciones adicionales en el futuro.

**Interfaz de Usuario:**

* Una interfaz gráfica más avanzada que permita al usuario ingresar los vectores y seleccionar la operación deseada (suma, resta, multiplicación, división, etc.,).

# Conclusión

Un vector, en términos informáticos, se define como una secuencia ordenada de elementos que comparten un mismo tipo de datos. Los vectores son fundamentales en la representación y manipulación de información en muchos campos, desde la simulación numérica hasta la computación gráfica. El programa desarrollado en lenguaje Java se enfoca en la implementación de algoritmos que permiten realizar operaciones fundamentales con vectores, como la suma y/o resta, pero más adelante se busca hacer la inclusión de la multiplicación, división y otras operaciones específicas según los requisitos del usuario.