

FACULDADE SENAI FATESG
BACHARELADO EM ENGENHARIA DE SOFTWARE - TURMA II
COMPONENTE CURRICULAR: FUNDAMENTOS MATEMÁTICOS I
PROFESSOR: UJEVERSON TAVARES SAMPAIO.
ALUNA: GABRIELLA PIO CORREA.

SITUAÇÃO DE APRENDIZAGEM: SIMULADOR DE VETORES

GOIÂNIA
2024

SUMÁRIO

01. Introdução.....	3
02. Requisitos do Sistema.....	4
03. Tutorial de Uso.....	5
04. Apêndice.....	9

01. Introdução

Seja bem-vindo ao manual de uso da Calculadora de Vetores, um software desenvolvido para facilitar a compreensão e a aplicação de operações vetoriais complexas. Este projeto originou-se da necessidade de integrar habilidades de programação com conceitos matemáticos fundamentais, desafio enfrentado por estudantes do segundo período do curso de Engenharia de Software da Faculdade SENAI FATESG.

No contexto acadêmico, a aplicação prática de conceitos teóricos de matemática a problemas reais de engenharia e computação apresenta-se como um desafio significativo. O objetivo primordial deste software é auxiliar estudantes e profissionais a entenderem e visualizarem operações vetoriais, essenciais em diversas áreas, como simulação física, gráficos computacionais e inteligência artificial.

Por meio da Calculadora de Vetores, será possível realizar e visualizar operações importantes, tais como cálculo de magnitude, produto escalar, ângulo entre vetores, ortogonalidade e produto vetorial. Desenvolvida em Java, a ferramenta foi projetada para ser intuitiva e de fácil utilização, permitindo a inserção de vetores e a exibição dos resultados das operações.

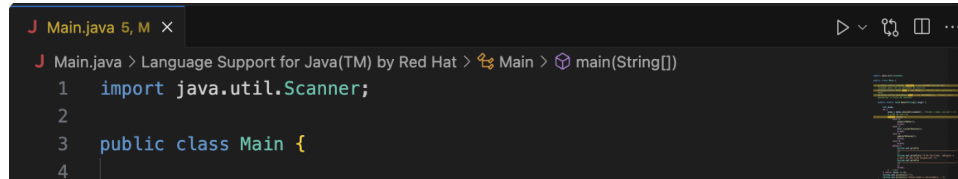
02. Requisitos do Sistema

- Sistema Operacional: Windows 10 ou superior, macOS 10.13 ou superior, ou qualquer distribuição Linux recente.
- Java: JDK 11 ou superior

03. Tutorial de Uso

- **Executando a Calculadora**

Para iniciar a Calculadora de Vetores, execute o programa clicando no símbolo de 'play' localizado no canto superior direito da interface.



```
J Main.java 5, M x
J Main.java > Language Support for Java(TM) by Red Hat > Main > main(String[])
1 import java.util.Scanner;
2
3 public class Main {
4
```

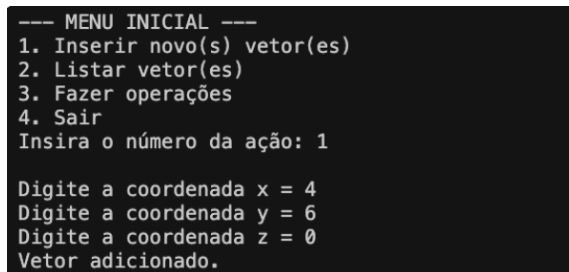
- **Menu Inicial**

Assim que o programa for executado, o Menu Inicial será exibido. Nele, digite o número da ação que você deseja realizar, pressione Enter e aguarde a próxima solicitação.

1. **Ação 1: Inserir novo(s) vetor(es)**

Para inserir um vetor:

1. Digite o valor da coordenada X e pressione Enter.
2. Repita o processo para as coordenadas Y e Z.



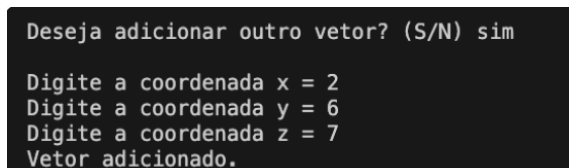
```
--- MENU INICIAL ---
1. Inserir novo(s) vetor(es)
2. Listar vetor(es)
3. Fazer operações
4. Sair
Insira o número da ação: 1

Digite a coordenada x = 4
Digite a coordenada y = 6
Digite a coordenada z = 0
Vetor adicionado.
```

Pronto, seu vetor foi adicionado com sucesso na calculadora!

Se quiser adicionar outro vetor:

1. Digite 'sim' e pressione Enter.
2. Insira as coordenadas do novo vetor.



```
Deseja adicionar outro vetor? (S/N) sim

Digite a coordenada x = 2
Digite a coordenada y = 6
Digite a coordenada z = 7
Vetor adicionado.
```

Se não quiser adicionar outro vetor:

1. Digite 'não' e pressione Enter.

Você será direcionado novamente ao menu inicial para optar por outra ação.

Observação: Se você quiser adicionar um vetor bidimensional (com apenas valores de X e Y), insira o valor 0 quando for solicitado o valor de Z.

2. Ação 2: Listar vetor(es)

Caso a ação escolhida seja a Ação 2, digite o número 2 e aperte Enter. Os vetores adicionados no software aparecerão listados para conferência.

```
Insira o número da ação: 2
Vetor 1: (4.0, 6.0, 0.0)
Vetor 2: (2.0, 6.0, 7.0)
```

3. Ação 3: Fazer Operações

Caso a ação escolhida seja a Ação 3, digite o número 3 e aperte Enter. Será exibido o Menu de Operações. Digite o número da operação que você deseja calcular:

```
Insira o número da ação: 3
--- MENU DE OPERAÇÕES ---
1. Calcular Módulo
2. Calcular Produto Escalar
3. Verificar Ortogonalidade
4. Calcular Ângulo entre Vetores
5. Calcular Produto Vetorial
Insira o número da operação: □
```

1. Operação 1: Módulo

Para calcular o módulo, digite 1 e aperte Enter.

Serão listados os vetores armazenados na calculadora, anteriormente inseridos na Ação 1.

Por fim, digite o número do vetor desejado.

```
Insira o número da operação: 1
Vetor 1: (4.0, 6.0, 0.0)
Vetor 2: (2.0, 6.0, 7.0)
Digite o número do vetor: 1
Módulo do vetor (4.0, 6.0, 0.0) é: 7.211102550927978
```

Pronto, será exibido o módulo do vetor selecionado.

2. Operação 2: Produto Escalar

Para calcular o produto escalar, digite 2 e aperte Enter.

Serão listados os vetores armazenados na calculadora, anteriormente inseridos por você na Ação 1.

Digite o número do primeiro vetor.

Digite o número do segundo vetor.

```
Insira o número da operação: 2
Vetor 1: (4.0, 6.0, 0.0)
Vetor 2: (2.0, 6.0, 7.0)
Digite o número do vetor: 1
Vetor 1: (4.0, 6.0, 0.0)
Vetor 2: (2.0, 6.0, 7.0)
Digite o número do vetor: 2
Produto escalar entre (4.0, 6.0, 0.0) e (2.0, 6.0, 7.0) é: 44.0
```

Pronto, será exibido o produto escalar entre os vetores selecionados.

3. Operação 3: Ortogonalidade

Para calcular a ortogonalidade, digite 3 e aperte Enter.

Serão listados os vetores armazenados na calculadora, anteriormente inseridos por você na Ação 1.

Digite o número do primeiro vetor e aperte Enter.

Digite o número do segundo vetor e aperte Enter.

```
Insira o número da operação: 3
Vetor 1: (4.0, 6.0, 0.0)
Vetor 2: (2.0, 6.0, 7.0)
Digite o número do vetor: 1
Vetor 1: (4.0, 6.0, 0.0)
Vetor 2: (2.0, 6.0, 7.0)
Digite o número do vetor: 2
Os vetores (4.0, 6.0, 0.0) e (2.0, 6.0, 7.0) não são ortogonais.
```

Pronto, será exibido se os vetores selecionados são ou não ortogonais.

4. Operação 4: Ângulo entre Vetores

Para calcular o ângulo entre dois vetores, digite 4 e aperte Enter.

Serão listados os vetores armazenados na calculadora, anteriormente inseridos por você na Ação 1.

Digite o número do primeiro vetor e aperte Enter.

Digite o número do segundo vetor e aperte Enter.

```
Insira o número da operação: 4
Vetor 1: (4.0, 6.0, 0.0)
Vetor 2: (2.0, 6.0, 7.0)
Digite o número do vetor: 1
Vetor 1: (4.0, 6.0, 0.0)
Vetor 2: (2.0, 6.0, 7.0)
Digite o número do vetor: 2
Ângulo entre os vetores (4.0, 6.0, 0.0) e (2.0, 6.0, 7.0) é: 49.70079965044623
```

Pronto, será exibido o ângulo entre os vetores selecionados.

5. Operação 5: Produto Vetorial

Para calcular o produto vetorial de dois vetores, digite 5 e aperte Enter.

Serão listados os vetores armazenados na calculadora, anteriormente inseridos por você na Ação 1.

Digite o número do primeiro vetor e aperte Enter.

Digite o número do segundo vetor e aperte Enter.

```
Insira o número da operação: 5
Vetor 1: (4.0, 6.0, 0.0)
Vetor 2: (2.0, 6.0, 7.0)
Digite o número do vetor: 1
Vetor 1: (4.0, 6.0, 0.0)
Vetor 2: (2.0, 6.0, 7.0)
Digite o número do vetor: 2
Produto vetorial entre (4.0, 6.0, 0.0) e (2.0, 6.0, 7.0): (42.0, -11.0, 24.0)
```

Pronto, será exibido o produto vetorial dos vetores selecionados.

4. Ação 4: Sair

Para encerrar a calculadora e limpar todos os dados, digite 4 e pressione Enter.

```
Insira o número da ação: 4
Encerrando a calculadora...
```


04. Apêndice

- **Conceitos Matemáticos Básicos:**

Definição de Vetor: Um vetor é um segmento de reta orientado no espaço, caracterizado por um módulo (comprimento), uma direção e um sentido.

Representação: Graficamente, um vetor é representado por uma seta. Algebricamente, pode ser representado por uma n-upla ordenada de números (coordenadas).

Operações:

- **Módulo:** O módulo de um vetor é uma medida de seu comprimento e pode ser calculada utilizando a fórmula:

Para um vetor v com as coordenadas (x, y) , seu módulo é: $|v| = \sqrt{x^2 + y^2}$

- **Produto Escalar:** O produto escalar de dois vetores é uma operação que resulta em um escalar, calculada pela fórmula:

O produto escalar entre o vetor $u = (x_1, y_1)$ e o vetor $v = (x_2, y_2)$, é: $u \cdot v$

$$u \cdot v = x_1x_2 + y_1y_2$$

- **Ortogonalidade:** Dois vetores são ortogonais se o ângulo entre eles é 90° , o que acontece quando o produto escalar é zero.

Dois vetores u e v são ortogonais se seu produto escalar é zero: $u \cdot v = 0$.

- **Ângulo entre vetores:** O ângulo entre dois vetores pode ser encontrado usando a fórmula:

O ângulo entre os vetores $u = (x_1, y_1)$ e $v = (x_2, y_2)$ é:

$$\theta = \arccos((u \cdot v) / (|u| |v|))$$

- **Produto Vetorial:** O produto vetorial de dois vetores em três dimensões resulta em um vetor que é perpendicular aos dois vetores originais:

Vetor A = (a_1, a_2, a_3) e Vetor B = (b_1, b_2, b_3) ;

$$x = (a_2 \cdot b_3) - (a_3 \cdot b_2)$$

$$y = (a_3 \cdot b_1) - (a_1 \cdot b_3)$$

$$z = (a_1 \cdot b_2) - (a_2 \cdot b_1)$$

$$\text{Vetor C} = (x, y, z)$$

- **Código fonte:**

Nesta seção, está descrita uma visão geral das classes utilizadas na Calculadora de Vetores. No código fonte, as ações são comentadas para facilitar o entendimento do funcionamento do software.

1. Classe Main

A classe `Main` é responsável pela execução do programa, apresentando o menu inicial, lendo as entradas do usuário e chamando os métodos apropriados para inserir vetores e realizar operações.

```
1  import java.util.Scanner;
2
3  public class Main {
4
5      private static Scanner scanner = new Scanner(System.in); //Scanner para ler as entradas do usuário
6      private static Menu menu = new Menu(); //Objeto para gerenciar o menu
7      private static utilVetor util = new utilVetor(); //Objeto para gerenciar a lista de vetores
8
9      Run | Debug | Run main | Debug main
10     public static void main(String[] args) {
11
12         int acao;
13         do {
14             acao = menu.inicial(scanner); //Exibe o menu inicial e lê a ação do usuário
15             switch (acao) {
16                 case 1:
17                     inserirVetor();
18                     break;
19                 case 2:
20                     util.listarVetores();
21                     break;
22                 case 3:
23                     operarVetores();
24                     break;
25                 case 4:
26                     break;
27                 default:
28                     System.out.println(x:"-----");
29                     System.out.println(x:"Ação Inválida. \nDigite o número de uma ação disponível.");
30                     System.out.println(x:"-----");
31                     break;
32             } //sair
33         } while (acao != 4);
34         System.out.println(x:"");
35         System.out.println(x:"Encerrando a calculadora...");
36     }
```

2. Classe Vetor

A classe `Vetor` representa um vetor com coordenadas X, Y e Z, e inclui métodos que definem o cálculo do módulo, produto escalar, ortogonalidade, ângulo entre vetores e produto vetorial.

```
1 public class Vetor {
2     //atributos x, y e z do vetor
3     private double x;
4     private double y;
5     private double z;
6
7 > public Vetor(double x, double y, double z) { //constructor: inicializa os atributos do vetor...
12
13     //getters; para acessar os atributos (privados)
14 > public double getX() {...
17 > public double getY() {...
20 > public double getZ() {...
23
24     public double modulo() { //define o cálculo do módulo
25         return Math.sqrt(this.x * this.x + this.y * this.y + this.z * this.z);
26     }
27
28     public double produtoEscalar(Vetor outro) { //define o cálculo do produto escalar
29         return this.x * outro.x + this.y * outro.y + this.z * outro.z;
30     }
31
32     public boolean ortogonalidade(Vetor outro){ //verifica a ortogonalidade
33         return this.produtoEscalar(outro) == 0;
34     }
35
36     public double anguloEntreVetores(Vetor outro){ //define o cálculo do ângulo entre vetores
37         double prodEscalar = this.produtoEscalar(outro);
38         double modulos = this.modulo() * outro.modulo();
39         return Math.toDegrees(Math.acos(prodEscalar / modulos)); //retorna o resultado em graus
40     }
41
42     public Vetor produtoVetorial(Vetor outro) { //define o cálculo do Produto vetorial
43         double pX = (this.y * outro.z - this.z * outro.y);
44         double pY = (this.z * outro.x - this.x * outro.z);
45         double pZ = (this.x * outro.y - this.y * outro.x);
46
47         return new Vetor(pX, pY, pZ);
48     }
49
50     @Override
51     public String toString() { //representação visual do vetor
52         return "(" + x + ", " + y + ", " + z + ")";
53     }
54 }
```

3. Classe Menu

A classe `Menu` exibe os menus de ações e operações disponíveis para o usuário e lê as escolhas do usuário.

```
1  import java.util.Scanner;
2
3  public class Menu {
4
5      public int inicial(Scanner scanner) { //mostra as ações iniciais disponíveis
6          System.out.println(x:"\n--- MENU INICIAL ---");
7          System.out.println(x:"1. Inserir novo(s) vetor(es)");
8          System.out.println(x:"2. Listar vetor(es)");
9          System.out.println(x:"3. Fazer operações");
10         System.out.println(x:"4. Sair");
11
12         System.out.print(s:"Insira o número da ação: ");
13         return scanner.nextInt(); //Retorna o valor da ação
14     }
15
16     public int operacoes(Scanner scanner) { //mostra as operações disponíveis
17         System.out.println(x:"\n--- MENU DE OPERAÇÕES ---");
18         System.out.println(x:"1. Calcular Módulo");
19         System.out.println(x:"2. Calcular Produto Escalar");
20         System.out.println(x:"3. Verificar Ortogonalidade");
21         System.out.println(x:"4. Calcular Ângulo entre Vetores");
22         System.out.println(x:"5. Calcular Produto Vetorial");
23
24         System.out.print(s:"Insira o número da operação: ");
25         return scanner.nextInt(); //Retorna o valor da operação
26     }
27 }
```

4. Classe utilVetor

A classe `utilVetor` gerencia a lista de vetores, permitindo adicionar novos vetores, recuperar vetores específicos e listar todos os vetores armazenados.

```
1  import java.util.ArrayList;
2  import java.util.List;
3
4  public class utilVetor {
5      private List<Vetor> vetores; //lista para armazenar os vetores
6
7      public utilVetor() { //constructor: inicializa a lista de vetores
8          vetores = new ArrayList<>();
9      }
10
11     public void addVetor(Vetor vetor) { //adiciona um novo vetor à lista
12         vetores.add(vetor);
13     }
14
15     public Vetor getVetor(int index) {
16         if (index >= 0 && index < vetores.size()) {
17             return vetores.get(index); //retorna o vetor da posição indicada
18         } else {
19             throw new IndexOutOfBoundsException(s:"Índice inválido."); //lança uma exceção caso o índice seja inválido
20         } //utilizado em: selecionarVetor();
21     }
22
23     public void listarVetores() { //lista todos os vetores armazenados
24         System.out.println(x:" ");
25         for (int i = 0; i < vetores.size(); i++) {
26             System.out.println("Vetor " + (i + 1) + ": " + vetores.get(i));
27         }
28     }
29 }
```