

## ADO 01 – Estruturas de Dados

**Nome:** Vitor Santana Sousa

**Descrição:** Uma loja de eletrônicos deseja criar um sistema em Java para gerenciar o estoque de seus produtos. O objetivo é manipular os dados dos produtos (marca, modelo e valor) armazenados em um vetor, permitindo ao usuário adicionar novos produtos, remover produtos existentes, buscar produtos por diferentes critérios, além de verificar a quantidade de produtos em estoque.

### Classe Vetor:

```
public class Vetor {

    private Produto[] elementos;

    private int tamanho;

    public Vetor(int capacidade) {
        this.elementos = new Produto[capacidade];
        this.tamanho = 0;
    }

    public void adiciona(Produto elemento) throws Exception {
        aumentaCapacidade();
        if(this.tamanho < this.elementos.length) {
            this.elementos[this.tamanho] = elemento;
            this.tamanho++;
        }
    }

    public int tamanho() {
        return this.tamanho;
    }

    @Override
    public String toString() {
```

```

        StringBuilder s = new StringBuilder();

        s.append("[");

        for(int i = 0; i < this.tamanho - 1; i++) {
            s.append(this.elementos[i]);
            s.append(", ");
        }

        if(this.tamanho > 0) {
            s.append(this.elementos[this.tamanho - 1]);
        }

        s.append("]");
        return s.toString();
    }

    public Object busca(int posicao) throws Exception {
        if(posicao >= 0 && posicao < tamanho) {
            return elementos[posicao];
        }
        else {
            throw new Exception("Posição Inválida");
        }
    }

    public int busca(Produto elemento) {
        for(int i = 0; i < tamanho; i++) {
            if(this.elementos[i].comparar(elemento)) {
                return i;
            }
        }
    }

```

```

        return -1;
    }

    public boolean adicionalInicio(int posicao, Produto elemento) throws Exception {
        if(posicao >= 0 && posicao < tamanho) {
            for(int i = this.tamanho - 1; i >= posicao; i--) {
                this.elementos[i + 1] = this.elementos[i];
            }
            this.elementos[posicao] = elemento;
            this.tamanho++;
        }
        else {
            throw new Exception("Posição Inválida");
        }
        return true;
    }

```

```

    public void aumentaCapacidade() {
        if(this.tamanho == this.elementos.length) {
            Produto[] elementosNovos = new Produto[this.elementos.length *
2];

            for(int i = 0; i < this.elementos.length; i++) {
                elementosNovos[i] = this.elementos[i];
            }
            this.elementos = elementosNovos;
        }
    }

```

```

    public void remove(int posicao) throws Exception {
        if(posicao >= 0 && posicao < tamanho) {
            for(int i = posicao; i < this.tamanho - 1; i++) {

```

```
                this.elementos[i] = this.elementos[i + 1];
            }
            this.tamanho--;
        }
        else {
            throw new Exception("Posição Inválida");
        }
    }
}
```

### **Classe Produto:**

```
public class Produto {

    private String marca;
    private String modelo;
    private double valor;

    public Produto() {}

    public Produto(String marca, String modelo, double valor) {
        this.marca = marca;
        this.modelo = modelo;
        this.valor = valor;
    }

    public String getMarca() {
        return marca;
    }

    public void setMarca(String marca) {
        this.marca = marca;
    }
}
```

```
}
```

```
public String getModelo() {  
    return modelo;  
}
```

```
public void setModelo(String modelo) {  
    this.modelo = modelo;  
}
```

```
public double getValor() {  
    return valor;  
}
```

```
public void setValor(double valor) {  
    this.valor = valor;  
}
```

```
public String toString() {  
    return "[" + marca + ", " + modelo + ", R$ " + valor + "];"  
}
```

```
public void menu() {  
    System.out.println("\nMENU DE OPÇÕES");  
    System.out.println("1 - Cadastrar Produto");  
    System.out.println("2 - Remover Produto");  
    System.out.println("3 - Exibir Estoque");  
    System.out.println("4 - Exibir Tamanho do Estoque");  
    System.out.println("5 - Adicionar Produto em uma Posição Específica");  
    System.out.println("6 - Buscar Produto por Posição");  
    System.out.println("7 - Buscar Produto por Objeto");  
}
```

```

        System.out.println("0 - Finalizar");

        System.out.printf("Opção: ");

    }

    public boolean comparar(Produto outro) {

        return this.marca.equals(outro.getMarca()) &&
this.modelo.equals(outro.getModelo()) && this.valor == outro.getValor();

    }

}

```

#### **Classe Teste:**

```

import java.util.Scanner;

public class Teste {

    public static void main(String[] args) {

        Scanner ler = new Scanner(System.in);

        Produto p = new Produto() {};

        Vetor estoque = new Vetor(5);

        int op;

        do {

            p.menu();

            op = ler.nextInt();

            if(op < 0 || op > 7) {

                System.err.println("\nOpção Inválida!");

                continue;

            }

        }
    }
}

```

```

if(op == 1) {
    ler.nextLine();
    System.out.printf("\nMarca: ");
    String marca = ler.nextLine();
    System.out.printf("Modelo: ");
    String modelo = ler.nextLine();
    System.out.printf("Valor: ");
    double valor = ler.nextDouble();
    Produto produto = new Produto(marca, modelo, valor);
    try {
        estoque.adiciona(produto);
        System.out.println("Produto      Adicionado      com
Sucesso!");
    } catch (Exception e){
        System.err.println("Erro: " + e.getMessage());
    }
}
else if(op == 2) {
    System.out.printf("\nInforme a Posição: ");
    int posicao = ler.nextInt();
    try {
        estoque.remove(posicao);
        System.out.println("Produto      Removido      com
Sucesso!");
    } catch (Exception e) {
        System.err.println("Erro: " + e.getMessage());
    }
}
else if(op == 3) {
    System.out.println("\nEstoque: " + estoque);
}
else if(op == 4) {

```

```

        System.out.println("\nTamanho do Estoque: " +
estoque.tamanho() + " produto(s)");
    }
    else if(op == 5) {
        System.out.printf("\nInforme a Posição: ");
        int posicao = ler.nextInt();
        ler.nextLine();
        System.out.printf("Marca: ");
        String marca = ler.nextLine();
        System.out.printf("Modelo: ");
        String modelo = ler.nextLine();
        System.out.printf("Valor: ");
        double valor = ler.nextDouble();
        Produto produto = new Produto(marca, modelo, valor);
        try {
            estoque.adicionalInicio(posicao, produto);
            System.out.println("Produto Adicionado com
Sucesso!");
        } catch (Exception e){
            System.err.println("Erro: " + e.getMessage());
        }
    }
    else if(op == 6) {
        System.out.printf("\nInforme a Posição: ");
        int posicao = ler.nextInt();
        try {
            Produto prod = (Produto) estoque.busca(posicao);
            System.out.println("Produto na Posição " + posicao +
": " + prod);
        } catch (Exception e) {
            System.err.println("Erro: " + e.getMessage());
        }
    }

```



```

    }
    else if(op == 7) {
        ler.nextLine();
        System.out.printf("\nMarca: ");
        String marca = ler.nextLine();
        System.out.printf("Modelo: ");
        String modelo = ler.nextLine();
        System.out.printf("Valor: ");
        double valor = ler.nextDouble();
        Produto produto = new Produto(marca, modelo, valor);
        int posicao = estoque.busca(produto);
        if(posicao >= 0) {
            System.out.println("Produto encontrado na
Posição: " + posicao);
        }
        else {
            System.err.println("Produto Não Encontrado.");
        }
    }
    else if(op == 0) {
        System.err.println("\nPrograma Finalizado!");
    }

}while(op != 0);

}

}

```