

UNIVERSIDADE PARANAENSE  
CURSO ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

VICTOR FAGUNDES RAMOS - RA:00239823

**TRABALHO ESTRUTURA DE DADOS**  
RELATÓRIO DO TRABALHO

CASCABEL  
2023

**Menu:** Como main criei uma classe chamada Menu, para executar os métodos que vão manipular o vetor. Assim que executar o programa ele pede para inserir um vetor utilizando o método “inserirNumeros()” presente na classe “Vetor”, após inserir o vetor um “while” será executado mostrando as opções, juntamente com um switch para escolher quais métodos escolher, como forma de entrada de dados escolhi a biblioteca Scanner e para saída utilizei “System.out.print”.

```
public class Menu {

    public static void main(String[] args) {
        Scanner ler = new Scanner(System.in);
        Vetor vetor = new Vetor();

        System.out.println("\nINSIRA UM VETOR PARA CONTINUAR");
        vetor.inserirNumeros();

        while(true){
            System.out.println("\nEscolha usando um número inteiro:");
            System.out.println("(1) Inserir Vetor de Inteiros");
            System.out.println("(2) Ordenar por Inserção");
            System.out.println("(3) Ordenar por Seleção");
            System.out.println("(4) Ordenar por Bolha");
            System.out.println("(5) Pesquisa Binária");
            System.out.println("(6) Pesquisa Linear");
            System.out.println("(7) Exibir Vetor Original");
            System.out.println("(0) Sair");
            System.out.print("Opção: ");
            int opcao = ler.nextInt();

            switch (opcao) {
                case 1:
                    vetor.inserirNumeros();
                    break;

                case 2:
                    vetor.exibirVetorOriginal();
                    vetor.ordenarInsercao();
                    break;

                case 3:
                    vetor.exibirVetorOriginal();
                    vetor.ordenarSelecao();
                    break;

                case 4:
                    vetor.exibirVetorOriginal();
                    vetor.ordenarBolha();
                    break;

                case 5:
                    vetor.pesquisaLinear();
                    break;

                case 6:
                    vetor.pesquisaBinaria();
                    break;

                case 7:
                    vetor.exibirVetorOriginal();
                    break;

                case 0:
                    System.out.println("\nVocê escolheu sair!");
                    System.exit(status:0);

                default:
                    System.out.println("\nOpção Inválida! Escolha novamente");
            }
        }
    }
}
```

**Ordenações:** Para ordenar o vetor conforme solicitado, criei uma classe chamada "Ordenacoes", a qual possui três métodos, sendo "ordenarInsercao", "ordenarSelecao" e "ordenarBolha", sendo todas do tipo "void" e recebendo como parâmetro o vetor, os métodos foram implementados conforme as particularidades que ordenação por inserção, ordenação por seleção e ordenação por bolha possuem, após ordenar o vetor, o método irá imprimir a posição e os elementos do vetor já ordenado.

```
: class Ordenacoes {

public static void ordenarInsercao(int[] vetor){
    int aux;
    for(int i = 1; i < vetor.length; i++){
        aux = vetor[i];
        int j = i - 1;
        while(j >= 0 && vetor[j] > aux){
            vetor[j + 1] = vetor[j];
            j--;
        }
        vetor[j + 1] = aux;

        System.out.println(":\nVetor Ordenado por Inserção:");
        for(int i = 0; i < vetor.length; i++){
            System.out.println("Posição["+i+"] = "+vetor[i]);
        }

    }

    public static void ordenarSelecao(int[] vetor){
        for(int i = 0; i < vetor.length; i++){
            int posicaoMenor = i;

            for(int j = i+1; j < vetor.length; j++){
                if(vetor[j] < vetor[posicaoMenor]){
                    posicaoMenor = j;
                }
            }

            if(posicaoMenor != i){
                int aux = vetor[i];
                vetor [i] = vetor[posicaoMenor];
                vetor[posicaoMenor] = aux;
            }
        }
        System.out.println(":\nVetor Ordenado por Seleção:");
        for(int i = 0; i < vetor.length; i++){
            System.out.println("Posicao[" + i+ " ] = "+vetor[i]);
        }
    }

    public static void ordenarBolha(int[] vetor){
        boolean mudou = true;

        while(mudou){
            mudou = false;

            for (int i = 0; i < vetor.length - 1; i++){
                if(vetor[i] > vetor[i+1]){
                    int aux = vetor [i];
                    vetor [i] = vetor [i+1];
                    vetor [i+1] = aux;
                    mudou = true;
                }
            }
        }
        System.out.println(":\nVetor Ordenado por Bolha:");
        for(int i = 0; i < vetor.length; i++){
            System.out.println("Posicao["+i+"] = "+vetor[i]);
        }
    }
}
```

**Pesquisa Linear e Binária:** Para a função de pesquisa Linear e Binária foi criado uma classe chamada “Pesquisa”, contendo dois métodos “pesquisaLinear” e “pesquisaBinaria” os quais são do tipo inteiro, onde ambos recebem como parâmetro o vetor e o elemento a ser pesquisado, no caso de “pesquisaLinear” utiliza um “for” para verificar se o elemento se encontra no vetor, e o “if” para retornar a posição caso seja encontrado, e caso não seja retorna -1. Já em “pesquisaBinaria”, utiliza “if” e “else if” para verificar se o elemento se encontra no vetor, caso não seja retorna -1 e caso seja retorna o número da posição.

```
public class Pesquisa {

    public static int pesquisaLinear(int[] vetor, int elemento){
        for(int i = 0; i < vetor.length; i++){
            if (vetor[i] == elemento){
                return i;
            }
        }
        return -1;
    }

    public static int pesquisaBinaria(int[] vetor, int elemento, int esquerda, int direita){
        if (esquerda > direita) {
            return -1;
        }

        int meio = (esquerda + direita) / 2;

        if (vetor[meio] == elemento) {
            return meio;
        } else if (vetor[meio] < elemento) {
            return pesquisaBinaria(vetor, elemento, meio + 1, direita);
        } else {
            return pesquisaBinaria(vetor, elemento, esquerda, meio - 1);
        }
    }
}
```

**Vetor:** Para manipular o vetor criei uma classe chamada “Vetor”, a qual possui um método chamado “inserirNumeros”, que vai pedir ao usuário definir um tamanho do vetor e os elementos para adicionar, no caso número inteiros. Através dessa classe será enviado o vetor para as outras, além de implementar algumas funções, como nos métodos de ordenações, foi inserido um sistema que conta em nanosegundo o tempo de execução do método, através do “System.nanoTime()”, onde a duração é imprimido após o término da execução. Além disso a classe conta com um método chamado “exibirVetorOriginal”, que tem como função guardar o vetor originalmente inserido para ser exibido após a ordenação do mesmo, assim sendo possível mostrar o vetor sem alteração e o vetor ordenado.

```
public class Vetor {
    Scanner ler = new Scanner(System.in);
    private int[] numeros;
    private int[] vetorOriginal;
    private int tamanho;
    long duracao;

    public void inserirNumeros() {
        System.out.println(":\nDigite o Tamanho do vetor: ");
        tamanho = ler.nextInt();
        numeros = new int[tamanho];
        vetorOriginal = new int[tamanho];

        for(int i = 0; i < tamanho; i++){
            System.out.println("\nDigite um numero inteiro para posição "+i);
            int elemento = ler.nextInt();
            numeros[i] = elemento;
            vetorOriginal[i] = numeros[i];
        }
    }

    public void ordenarInsercao() {
        long comecar = System.nanoTime();
        Ordenacoes.ordenarInsercao(vetor: numeros);
        long finalizar = System.nanoTime();
        duracao = finalizar - comecar;
        System.out.println("\nTempo de execução(ordenarInsercao): " + duracao + " nanosegundo");
    }

    public void ordenarSelecao() {
        long comecar = System.nanoTime();
        Ordenacoes.ordenarSelecao(vetor: numeros);
        long finalizar = System.nanoTime();
        duracao = finalizar - comecar;
        System.out.println("\nTempo de execução(ordenarSelecao): " + duracao + " nanosegundo");
    }

    public void ordenarBolha() {
        long comecar = System.nanoTime();
        Ordenacoes.ordenarBolha(vetor: numeros);
        long finalizar = System.nanoTime();
        duracao = finalizar - comecar;
        System.out.println("\nTempo de execução(ordenarBolha): " + duracao + " nanosegundo");
    }
}
```

```

public void pesquisaLinear() {
    System.out.println(":\nDigite um numero para pesquisar: ");
    int numeroPesquisado = ler.nextInt();
    int resultado = Pesquisa.pesquisaLinear(vetor: numeros, elemento: numeroPesquisado);
    if(resultado != -1){
        System.out.println("\nO número '" + numeroPesquisado + "' está na posição " + resultado);
    }else{
        System.out.println("\nNúmero '" + numeroPesquisado + "' não encontrado");
    }
}

public void pesquisaBinaria() {
    System.out.println(":\nDigite um numero para pesquisar: ");
    int numeroPesquisado = ler.nextInt();
    int resultado = Pesquisa.pesquisaBinaria(vetor: numeros, elemento: numeroPesquisado, esquerda: 0, tamanho -1);
    if(resultado != -1){
        System.out.println("\nO número '" + numeroPesquisado + "' está na posição " + resultado);
    }else{
        System.out.println("\nNúmero '" + numeroPesquisado + "' não encontrado");
    }
}

public void exibirVetorOriginal() {
    System.out.println(":\nVetor Original: ");
    for(int i =0; i < vetorOriginal.length; i++){
        System.out.println("Posicao[" + i+ "] = " +vetorOriginal[i]);
    }
}

```