

## **TRABALHO ESTRUTURA DE DADOS 1º BIMESTRE – 3º ADS NOTURNO**

**Participantes:** Adriel Esteves Wilton Águila, RA: 1090482313051; Arthur César Rodrigues, RA: 1090482313033; Pedro Junior de Paula Silva, RA: 1090482313021.

O programa foi desenvolvido em JavaScript, tem suas funções: cadastrar alunos - essa função deve cadastrar: o nome, RA, idade, sexo – masculino ou feminino, resultado, média e resultado (aprovado ou reprovado) de cada aluno; gerar relatório em ordem crescente por nome, gerar relatório de alunos em ordem decrescente por RA; gerar relatório de alunos em ordem crescente por nome, apenas dos aprovados e encerrar o programa.

O programa usou ordenação com Selection-Sort, por conta de ser o método ensinado na matéria Estrutura de Dados mais eficiente em relação a tempo de processamento e uso de memória até o momento, comparando com o Bubble-Sort (outro método de ordenação, mais demorado e alto uso de memória). Porém a outros métodos de ordenação mais ágeis, mais o Selection-Sort foi escolhido por termos maior conhecimento.

Houve a implementação da função externa Readline Sync, encontrada no site: <https://www.npmjs.com/package/readline-sync> . Essa função permite entrada de dados do usuário através do prompt de comando e powershell no terminal do VS Code. A ideia principal era que a entrada de dados fosse através de um formulário e inputs no HTML, porém pela falta de conhecimento, optamos pelo uso do Readline Sync.

O software não tem limite de alunos para cadastrar, sua condição de parada é a opção 5.

Essa função permite a entrada de dados de 3 tipos de dados: `readline.question()` – recebe string, `readline.questionInt()` – recebe número inteiro e `readline.questionFloat()` – recebe número real.

Existe duas maneiras de rodar o programa: junto deste pdf, há uma pasta com nome de “trabalho-estrutura-01”, abra ela no VSCode e em seu terminal, pelo Powers Shell ou Command Prompt, copie o seguinte caminho “node estrutura-01.js” e pressione “enter”. A outra maneira: crie uma pasta, abra no VS Code, crie um arquivo .js e cole o código abaixo neste arquivo; dentro do terminal no VSCode copie e cole o seguinte texto “npm i readline-sync”, após essa etapa, ainda no terminal, digite “node arquivocriado.js” e pressione “enter”.

Código do programa:

```
const readline = require("readline-sync"); // foi criada essa variavel para receber os dados do usuário
```

```
function selectionSort (vetor, comparador) {(parametro do vetor e parametro da função de comparação)
```

```
    for (let posicaoSelecionada = 0; posicaoSelecionada < vetor.length - 1; posicaoSelecionada++) {(para posição selecionada = 0 a até numero de elementos do vetor - 1, passo 1
```

```
        let posicaoMenor = posicaoSelecionada + 1;{//posicao menor= posicao selecionada + 1
```

```
        for (let i = posicaoMenor + 1; i < vetor.length; i++) {
```

```
            //i = posição menor + 1 até numero de elemento
```

```
            if(comparador(vetor[posicaoMenor], vetor[i])) {(se função de comparação(vetor na posição menor, vetor na posição i )
```

```
                posicaoMenor = i;{//descobre a posição do menor valor
```

```
            }
```

```
        }
```

```
        if (comparador(vetor[posicaoSelecionada], vetor[posicaoMenor])) {
```

```
            [vetor[posicaoSelecionada], vetor[posicaoMenor]] = [vetor[posicaoMenor], vetor[posicaoSelecionada]];{//troca de valores
```

```
        }
```

```
    }
```

```
}
```

```
function comparaNome (primeiro, segundo) {
```

```
    return primeiro.nome > segundo.nome;{// função para ordenar em ordem crescente comparando pela propriedade nome
```

```
}
```

```
function comparaRa (primeiro, segundo) {
```

```
    return segundo.ra > primeiro.ra;{// função para ordenar em ordem decrescente comparando pela propriedade ra
```

```
}
```

```
function nomeAprovados (vetor) {(função pega apenas os alunos aprovados
```

```
for (let j = 0; j < vetor.length; j++){  
    if (vetor[j].resultado === "Aprovado") {  
        console.log(vetor[j]);  
    }  
}  
  
}
```

```
let aluno = {}; // inicialização do objeto aluno  
let posicao = []; // inicialização do vetor posição  
let aux2; // inicialização da variável aux2 que vai ser usada para parada do  
looping
```

```
do { //faça (continuação linha 157)
```

```
    aux2 = true; //atribuição ao aux2 de verdadeiro para rodar o código
```

```
    console.log(`Entre com uma das opções abaixo:
```

1. Cadastrar Alunos.
2. Relatório de Alunos em ordem crescente por Nome.
3. Relatório de Alunos em ordem decrescente por RA.
4. Relatório de Alunos em ordem crescente por Nome, apenas dos Aprovados.
5. Encerre a execução do programa.`); //menu de opções ao usuário na tela

```
    let opc = readline.questionInt("Opcao: "); //leitura da opção digitada pelo  
usuário, que é um número inteiro de 1 a 5
```

```
    console.log(""); //quebra de linha
```

```

switch(opc) { //opções de 1 a 5

    case 1: // caso o usuário digitou a opção 1

        aluno = { //objeto aluno

            nome: readline.question("Digite o nome do aluno(a): "), //lendo o valor
            //que o usuário digitar e atribuir direto a propriedade nome do objeto aluno;
            //readline.question = recebe string

            ra: readline.questionInt("RA (apenas os numeros): "), //lendo o valor
            //que o usuário digitar e atribuir direto a propriedade ra do objeto aluno;
            //readline.questionInt = recebe número inteiro

            idade: readline.questionInt("Idade: "), //lendo o valor que o usuário
            //digitar e atribuir direto a propriedade idade do objeto aluno;

            media: readline.questionFloat("Media: `"), //lendo o valor que o usuário
            //digitar e atribuir direto a propriedade media do objeto aluno;
            //readline.questionFloat = recebe número real

        }

        let aux; //inicialização da variavel aux para parada do looping

        do { //faça (continuação na linha 94 )

            aux = true; //aux recebe valor verdadeiro como padrão

            aluno.sexo = readline.question(("Sexo ("m" - para masculino e "f" -
            //para feminino): `")); //lendo o valor que o usuário digitar e atribuir direto a
            //propriedade sexo do objeto aluno;

            aluno.sexo.toLowerCase(); //transformando a string digitada em letra
            //minuscula

            if (aluno.sexo === "m" || aluno.sexo === "f") { // se valor digitado for a
            //letra "m"(masculino) ou a letra "f"(feminino) que são as opções válidas no
            //cadastro campo sexo

                aux = false; //aux recebe falso para sair do looping

            }

```

```

    } while(aux === true); //enquanto aux = true (verdadeiro)
    //quando o programa sai do "do while", então aluno.sexo é = "m" ou "f"

    if (aluno.sexo === "m"){ //se aluno.sexo for = m
        aluno.sexo = "Masculino"; //aluno.sexo recebe "Masculino"
    } else { //senao
        aluno.sexo = "Feminino"; //aluno.sexo recebe "Feminino"
    }

    do { //faça (continuação na linha 113)
        aux = true; //aux recebe valor verdadeiro como padrão

        aluno.resultado = readline.question(`Resultado ("a" - para aprovado
e "r" - para reprovado): `); //lendo o valor que o usuário digitar e atribuir direto a
propriedade resultado do objeto aluno;

        aluno.resultado = aluno.resultado.toLowerCase(); //transformando a
string digitada em letra minuscula

        if (aluno.resultado === "a" || aluno.resultado === "r"){ // se valor
digitado for a letra "a"(aprovado) ou a letra "r"(reprovado) que são as opções
válidas no cadastro campo resultado

            aux = false; //aux recebe falso para sair do looping
        }
    } while(aux === true); //enquanto aux = true (verdadeiro)

    //quando o programa sai do "do while", então aluno.resultado é = "a" ou
    "r"

    if (aluno.resultado === "a"){ //se aluno.resultado for = a
        aluno.resultado = "Aprovado"; //aluno.resultado recebe "Aprovado"
    } else { //senao
        aluno.resultado = "Reprovado"; //aluno.resultado recebe "Reprovado"
    }

```

```
}
```

```
posicao.push(aluno);//o objeto aluno digitado entra dentro do vetor  
aluno
```

```
console.log("");//quebra de linha
```

```
break;//depois de executar os comando da opção 1 pare
```

```
case 2:// caso o usuário digitou a opção 2
```

```
console.log("Relatório de Alunos em ordem crescente por  
Nome:");//titulo do relatorio
```

```
console.log("");//quebra de linha
```

```
selectionSort(posicao, comparaNome);//chamada da função  
selectionSort com os valores posição (vetor), e comparaNome (função de  
comparação)
```

```
console.log(posicao);// depois de ordenado o vetor de objeto e ordem  
crescente por nome, a saída de dados para o usuário na tela
```

```
console.log("");//quebra de linha
```

```
break;//depois de executar os comando da opção 2 pare
```

```
case 3:// caso o usuário digitou a opção 3
```

```
console.log("Relatório de Alunos em ordem decrescente por  
RA:");//titulo do relatorio
```

```
console.log("");//quebra de linha
```

```
selectionSort(posicao, comparaRa);//chamada da função selectionSort  
com os valores posição (vetor), e comparaRa (função de comparação)
```

```
console.log(posicao);// depois de ordenado o vetor de objeto e ordem  
decrescente por RA, a saída de dados para o usuário na tela
```

```
console.log("");//quebra de linha
```

```
break;//depois de executar os comando da opção 3 pare
```

```
case 4:// caso o usuário digitou a opção 4
```

```
console.log("Relatório de Alunos em ordem crescente por Nome,  
apenas dos Aprovados:");//titulo do relatorio
```

```
console.log("");//quebra de linha
```

selectionSort(posicao, comparaNome);//chamada da função  
selectionSort com os valores posição (vetor), e comparaNome (função de  
comparação)

nomeAprovados(posicao);//depois de ordenado em ordem crescente  
por nome, é chamado a função nomeAprovado, que verifica e retorna para o  
usuário apenas os alunos aprovados

console.log("");// quebra de linha

break;//depois de executar os comando da opção 4 pare

case 5:// caso o usuário digitou a opção 5

console.log("Encerrando o programa...");//mensagem ao usuário

aux2 = false;//aux recebe falso para sair do looping

break;//depois de executar os comando da opção 5 pare

default:// caso o usuário digitou nenhuma das opções acima

console.log("Opção inválida!!");//mensagem ao usuário

console.log("");// quebra de linha

}

} while(aux2 === true); // enquanto aux2 for igual a true (verdadeiro)