Católica
Universidade Católica de Brasília

Pró-Reitoria Acadêmica
Cursos de Ciência da Computação
e Análise e Desenvolvimento de Sistemas
Trabalho Disciplinar de Estrutura de Dados

ATIVIDADE AVALIATIVA: ESTRUTURA DE DADOS

Autores: Gabriel Victor Nasulicz Martins,
Gabriel Morais Resplandes,
Giovana Pacheco Velasco,
Guilherme Vaz Gomes,
Maria Luiza Barbosa Ferreira da Silva,
Raíssa Herculano de Matos

Orientador: Zoé Roberto Magalhães Junior

Brasília - DF
2025

GABRIEL VICTOR NASULICZ MARTINS - UC24200813

GIOVANA PACHECO VELASCO - UC24200330

GUILHERME VAZ GOMES – UC24100762

MARIA LUIZA BARBOSA FERREIRA DA SILVA - UC23200096

ATIVIDADE AVALIATIVA:

ESTRUTURA DE DADOS.

Documento apresentado aos Cursos de graduação de Ciência da Computação da Universidade Católica de Brasília, como requisito parcial para obtenção da aprovação na disciplina de Estrutura de Dados.

Orientador: Prof. Zoé Roberto Magalhães Junior

Brasília
2025

1. OBJETIVO DA SOLUÇÃO:

O objetivo deste programa é realizar a ordenação de elementos contidos em um arquivo CSV, utilizando diferentes métodos de ordenação estudados em sala de aula, como o **Processamento de strings** e **Selection Sort**.
Esses métodos foram aplicados para ordenar dados de diferentes tipos: inteiros, floats e strings.

O projeto foi desenvolvido na linguagem C e utiliza como base um sistema de cadastro de produtos, contendo as seguintes informações:

- Nome do produto (string)
- Quantidade em estoque (inteiro)
- Valor do produto (float)
- Código do produto (inteiro)

Além disso, o programa permite que o usuário escolha o modo de ordenação desejado, podendo organizar os dados em **ordem crescente**, **ordem decrescente ou ordem alfabética**.

2. DESCRIÇÃO DA ENTRADA E SAÍDA DE DADOS:

A **entrada de dado**s do projeto consiste em um arquivo CSV contendo o cadastro de produtos.
Cada linha do arquivo representa um produto, com os seguintes campos, separados pelo caractere ";".

- Nome do produto (string).
- Preço do produto (float).
- Quantidade em estoque (inteiro).
- Código do produto (inteiro)

Exemplo de linhas no arquivo CSV:

BiscoitoTrakinas;4.49;6582;374890;
Toddynho;2.50;4343;738918;

A **saída de dados** consiste na geração de um novo arquivo CSV, contendo os mesmos registros, porém ordenados conforme a escolha do usuário.

Exemplos de organização:

Ordem alfabética (nome do produto):

BiscoitoTrakinas;4.49;6582;374890;
Toddynho;2.50;4343;738918;

Ordem crescente (por preço):

Toddynho;2.50;4343;738918;
BiscoitoTrakinas;4.49;6582;374890

3. ALGORITMO DE ORDENAÇÃO UTILIZADO:

O algoritmo de ordenação escolhido para a confecção do projeto foi o Selection Sort, Esse algoritmo percorre o vetor de produtos, selecionando o menor (ou maior) valor a cada passo e trocando de posição para organizá-lo.

Além disso, foram usadas funções como:

- **strtok()** para separar a linha em partes menores baseadas no caractere ;.
- **strcpy()** para copiar o conteúdo de uma string para outra variável do tipo string
- **atof()** para converter strings em números decimais (preço).
- **atoi()** para converter strings em números inteiros (quantidade e código).
- **strcspn()** para remover caracteres indesejados de uma string, como \n ou ;.

4. ESTRUTURA DO CÓDIGO E PRINCIPAIS FUNÇÕES IMPLEMENTADAS:

O código foi dividido em várias partes para melhorar sua organização:

- **copiarArquivo():** lê todo o conteúdo do arquivo e armazena em uma string.
- **obterNumeroLinhas():** Função que percorre o arquivo e obtém um inteiro igual ao número de linhas.
- **obterLinhas():** Função que divide a string obtida com a função anterior em strings menores e armazena seus valores em vários itens do tipo string de uma struct, depois converte os valores dessa struct para outra com tipos de dados diferentes e retorna a struct.

- **salvarCSV()** - Função responsável por criar um arquivo com o nome digitado pelo usuário e a ordenação escolhida no menu.
- **exibirMenu()** - Função responsável por exibir um menu ao usuário

Função main():

Coordena a execução, chamando as outras funções na sequência correta. Essa divisão de funções torna o código mais modular e fácil de entender.

Funções de Ordenação:

- **ordenarPorNome() -** Função responsável por ordenar a lista de itens por ordem alfabética.
- **ordenarPorItemCrescente() -** Grupo de funções com nome similar que organiza os produtos do menor para o maior, de acordo com o item especificado.
- **ordenarPorItemDecrescente() -** Grupo de funções com nome similar que organiza os produtos do maior para o menor, de acordo com o item especificado.

5. PROBLEMAS ENCONTRADOS E SOLUÇÕES ADOTADAS:

- **NULL Pointers:** Diversas vezes o programa falhava em executar do jeito esperado pois alguma das várias strings manipuladas continham um ponteiro NULL. Isso acontecia principalmente ao usar a função strtok(), mas algo parecido também ocorreu ao manusear inputs do usuário com fgets().
    - **Soluções:** As soluções variaram dependendo da ocasião em que o ponteiro era referenciado, mas uma solução comum foi usar a função strcspn() para percorrer a string e encontrar o caractere que causava o problema.
- **Manipulação de ponteiros.** Ao utilizar strings em programas C, é necessário ter bom domínio de como funcionam os ponteiros. Algumas vezes o uso incorreto de ponteiros causou problemas na execução do programa que tomaram várias horas de reflexão e revisão.

    - **Solução:** A solução foi comparativamente muito simples se comparada ao problema causado, já que normalmente era apenas uma questão de inserir um * no local correto.
- **Comparação de String:** Obtivemos um problema para ordenar strings, do qual não era possível somente com os sinais comuns (“<” e “>”).
    - **Solução:** A solução encontrada foi utilizar a função “strcmp()” do qual é responsável por comparar duas strings**.**
- **Transformar os campos char em tipos diferentes, como int, float:** Um dos requisitos da atividade foi tratar os campos como string e o processamento deveria considerar os diferentes tipos.
    - **Solução:** A solução encontrada foi desenvolver uma struct que pegasse todos os campos como char e depois desenvolver uma outra struct que consideraria os diferentes tipos de campo. Essa mudança foi realizada com o auxílio de uma função struct “obterLinhas”.

6. CONCLUSÃO:

O desenvolvimento deste projeto permitiu a aplicação prática dos conhecimentos adquiridos sobre leitura de arquivos CSV, processamento de strings e implementação do algoritmo Selection Sort na linguagem C.
O programa atendeu aos requisitos estabelecidos, sendo capaz de importar os dados corretamente, permitir a escolha do critério de ordenação pelo usuário e gerar um novo arquivo CSV com os dados organizados conforme a seleção.

A divisão do código em funções específicas facilitou a organização, a manutenção e a compreensão do programa. Além disso, o uso de funções auxiliares como strtok(), atof() e atoi() foi essencial para o tratamento correto dos diferentes tipos de dados.

Com este projeto, foi possível aprofundar o entendimento sobre manipulação de arquivos, conversão de dados e ordenação de estruturas em C, reforçando a importância de boas práticas de programação.

Como melhorias futuras, destaca-se:

- A diminuição da quantidade de funções para tornar o fluxo do código mais direto;

- A verificação de possíveis situações de overflow de memória;
- A redução do número de variáveis utilizadas;
- E a simplificação geral do código para deixá-lo mais eficiente e fácil de manter.

7. REFERÊNCIAS:

C READ FILES. **W3Schools**, 2025. Disponível em: https://www.w3schools.com/c/c_files_read.php. Acesso em: 24/04/2025.

C STRUCTURES (STRUCTS). **W3Schools**, 2025. Disponível em: https://www.w3schools.com/c/c_structs.php Acesso em: 24/04/2025.

C STDIO FGETS() FUNCTION. **W3Schools**, 2025. Disponível em: https://www.w3schools.com/c/ref_stdio_fgets.php. Acesso em: 24/04/2025.

C FILES. **W3Schools**, 2025. Disponível em: https://www.w3schools.com/c/c_files.php. Acesso em: 24/04/2025.

C STRING STRCSPN() FUNCTION. **W3Schools**, 2025. Disponível em: https://www.w3schools.com/c/ref_string_strcspn.php. Acesso em: 24/04/2025.

STRCSPN() IN C FUNCTION. **GeeksForGeeks**, 2025. Disponível em: https://www.geeksforgeeks.org/strcspn-in-c/. Acesso em: 24/04/2025.

IBM Security QRadar Suite - Log Insights . Disponível em: https://www.ibm.com/docs/pt-br/uax?topic=functions-strcmp>. Acesso em: 27 abr. 2025

Selection Sort . Disponível em: https://deinfo.uepg.br/~alunoso/2022/AEP/SELECTION_SORT/>. Acesso em: 27 abr. 2025.