

Projeto – Fase 2

Entrega da Primeira Versão Operacional: implementação de estruturas de dados e algoritmo geral, interface e funcionalidades possivelmente simplificadas

Atividade avaliativa - Projeto

Objetivos:

- Finalizar uma primeira versão do projeto, operacional, com a implementação das estruturas de dados, e do algoritmo geral, com funcionalidade e interface possivelmente simplificadas;
- Propiciar interação e feedback sobre a primeira versão operacional da implementação;
- Avançar o desenvolvimento do Projeto.

Atividade individual ou em grupo? Em grupo.

Prazo para entrega: até a data indicada no ambiente de interação da disciplina no Google Classroom.

Onde/como entregar: No ambiente de interação da disciplina no Google Classroom, no link indicado.

Quanto vale: 20% da nota do Projeto.

1- Equipe

Nome	O Que Fez no Projeto
Matheus Goulart Ranzani	Criação dos TADs e funções primitivas.
Rodrigo Pavão Coffani Nunes	Criação das funções não primitivas.
Vinícius Matheus Romualdo Santos	Implementação da interface gráfica.

2- Qual é a aplicação?

A aplicação se trata de um estoque de produtos de um supermercado. Nele é possível ter o controle dos produtos que estão à venda neste supermercado. Os produtos possuem informações básicas tais como: nome, preço e quantidade em estoque. Essas informações são armazenadas e gerenciadas em uma estrutura de dados do tipo Lista Cadastral.

3- Qual(is) estrutura(s) de dados estão sendo utilizada(s) e de que forma?

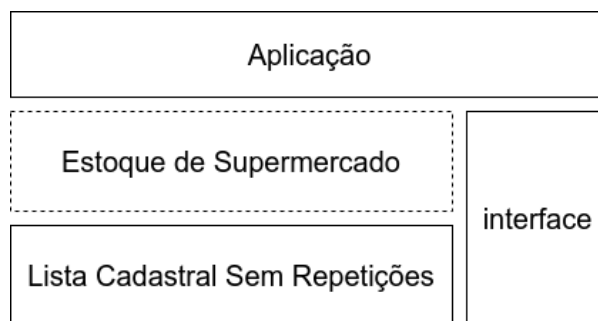
A estrutura de dados que será utilizada é uma Lista Cadastral. Ela será implementada com alocação encadeada e dinâmica de memória. A princípio terá ordenação com base na ordem de inserção dos produtos. Dessa maneira, será possível realizar todas as operações essenciais para gerenciar o estoque do supermercado. Alguns exemplos de operações essenciais são: inserção de

produtos, remoção de produtos, verificar se um produto está no estoque e alteração da quantidade.

4- Quais são as principais unidades de software?

Serão utilizadas duas estruturas principais, um TAD “Estoque” e um TAD “Produto”.

- TAD “Estoque”: Servirá como nossa Lista Cadastral, possuindo o endereço de memória dos elementos presentes no início e final da Lista.
- TAD “Produto”: Representando os elementos da Lista Cadastral, o Produto possuirá os campos “nome” (string), “preço” (double), “quantidade” (int) e “proximo” (Produto *).



5- Alguns prints demonstrando a execução / testes da Versão 1

Compilando e executando o programa / Menu inicial

```
ranzani in AED1/Projeto on ʘ main
→ g++ main.cpp -o main
ranzani in AED1/Projeto on ʘ main [!]
→ ./main
Controle de estoque

Escolha uma opcao:
1 - Adicionar produto
2 - Alterar quantidade de um produto
3 - Remover produto
4 - Exibir produto
5 - Exibir estoque
6 - Encerrar execução
Digite a opcao desejada: █
```

Adicionando produtos ao estoque

```
Controle de estoque

Escolha uma opcao:
1 - Adicionar produto
2 - Alterar quantidade de um produto
3 - Remover produto
4 - Exibir produto
5 - Exibir estoque
6 - Encerrar execucao
Digite a opcao desejada: 1

Informe os dados do produto:
Nome: Arroz Branco 1kg
Preço: 8.99
Quantidade: 25

Produto adicionado com sucesso!

Pressione qualquer tecla para continuar. . . █
```

Exibindo estoque após as adições dos produtos

```
Controle de estoque

Escolha uma opcao:
1 - Adicionar produto
2 - Alterar quantidade de um produto
3 - Remover produto
4 - Exibir produto
5 - Exibir estoque
6 - Encerrar execucao
Digite a opcao desejada: 5

Nome: Arroz Branco 1kg
Preço: 8.99
Quantidade: 25

Nome: Feijao Carioca 500 g
Preço: 5.5
Quantidade: 50

Pressione qualquer tecla para continuar. . . █
```

Tentando alterar a quantidade de um produto não cadastrado

Controle de estoque

Escolha uma opção:

- 1 - Adicionar produto
- 2 - Alterar quantidade de um produto
- 3 - Remover produto
- 4 - Exibir produto
- 5 - Exibir estoque
- 6 - Encerrar execução

Digite a opção desejada: 2

Informe o nome do produto que deseja alterar a quantidade: Feijao Preto

Produto não encontrado no estoque

Pressione qualquer tecla para continuar. . . █

Alterando a quantidade de um produto do estoque

Controle de estoque

Escolha uma opção:

- 1 - Adicionar produto
- 2 - Alterar quantidade de um produto
- 3 - Remover produto
- 4 - Exibir produto
- 5 - Exibir estoque
- 6 - Encerrar execução

Digite a opção desejada: 2

Informe o nome do produto que deseja alterar a quantidade: Feijao Carioca 500 g

Produto encontrado no estoque:

Nome: Feijao Carioca 500 g

Preço: 5.5

Quantidade: 50

Informe a nova quantidade do produto:

Controle de estoque

Escolha uma opção:

- 1 - Adicionar produto
- 2 - Alterar quantidade de um produto
- 3 - Remover produto
- 4 - Exibir produto
- 5 - Exibir estoque
- 6 - Encerrar execução

Digite a opção desejada: 2

Informe o nome do produto que deseja alterar a quantidade: Feijao Carioca 500 g

Produto encontrado no estoque:

Nome: Feijao Carioca 500 g

Preço: 5.5

Quantidade: 50

Informe a nova quantidade do produto: 30

Quantidade alterada com sucesso!

Pressione qualquer tecla para continuar. . .

Procurando e mostrando um produto do estoque

```
Controle de estoque

Escolha uma opcao:
1 - Adicionar produto
2 - Alterar quantidade de um produto
3 - Remover produto
4 - Exibir produto
5 - Exibir estoque
6 - Encerrar execucao
Digite a opcao desejada: 4

Informe o nome do produto que deseja exibir: Feijao Carioca 500 g

Produto encontrado no estoque:
Nome: Feijao Carioca 500 g
Preço: 5.5
Quantidade: 30

Pressione qualquer tecla para continuar. . .
```

Tentando excluir um produto que não está no estoque

```
Controle de estoque

Escolha uma opcao:
1 - Adicionar produto
2 - Alterar quantidade de um produto
3 - Remover produto
4 - Exibir produto
5 - Exibir estoque
6 - Encerrar execucao
Digite a opcao desejada: 3

Informe o nome do produto que deseja remover: Batat frita

Falha ao remover o produto

Pressione qualquer tecla para continuar. . .
```

Removendo um produto do estoque

```
Controle de estoque

Escolha uma opcao:
1 - Adicionar produto
2 - Alterar quantidade de um produto
3 - Remover produto
4 - Exibir produto
5 - Exibir estoque
6 - Encerrar execucao
Digite a opcao desejada: 3

Informe o nome do produto que deseja remover: Arroz Branco 1kg

Produto removido com sucesso!

Pressione qualquer tecla para continuar. . . █
```

Exibindo o estoque após a remoção

```
Controle de estoque

Escolha uma opcao:
1 - Adicionar produto
2 - Alterar quantidade de um produto
3 - Remover produto
4 - Exibir produto
5 - Exibir estoque
6 - Encerrar execucao
Digite a opcao desejada: 5

Nome: Feijao Carioca 500 g
Preço: 5.5
Quantidade: 30

Pressione qualquer tecla para continuar. . . █
```

Removendo outro produto do estoque e exibindo o estoque atualizado

Controle de estoque

Escolha uma opcao:

- 1 - Adicionar produto
- 2 - Alterar quantidade de um produto
- 3 - Remover produto
- 4 - Exibir produto
- 5 - Exibir estoque
- 6 - Encerrar execucao

Digite a opcao desejada: 3

Informe o nome do produto que deseja remover: Feijao Carioca 500 g

Produto removido com sucesso!

Pressione qualquer tecla para continuar. . . █

Controle de estoque

Escolha uma opcao:

- 1 - Adicionar produto
- 2 - Alterar quantidade de um produto
- 3 - Remover produto
- 4 - Exibir produto
- 5 - Exibir estoque
- 6 - Encerrar execucao

Digite a opcao desejada: 5

Estoque vazio

Pressione qualquer tecla para continuar. . . █

Tentando inserir um opção inválida no menu

```
Controle de estoque

Escolha uma opção:
1 - Adicionar produto
2 - Alterar quantidade de um produto
3 - Remover produto
4 - Exibir produto
5 - Exibir estoque
6 - Encerrar execução
Digite a opção desejada: 7

Opção inválida

Pressione qualquer tecla para continuar. . .
```

Encerrando o programa

```
Controle de estoque

Escolha uma opcao:
1 - Adicionar produto
2 - Alterar quantidade de um produto
3 - Remover produto
4 - Exibir produto
5 - Exibir estoque
6 - Encerrar execução
Digite a opcao desejada: 6

Programa finalizado!
```

6- O que falta fazer

- Aprimoramento da interface gráfica (RAD Studio);
- Criação de novas funções como: contagem total do estoque, filtros (produto com maior quantidade em estoque, produto com maior preço...), etc.

7- Código da Versão 1

Arquivo estoque.hpp

```
#include <iostream>

typedef struct Produto {
    std::string nome;
    double preco;
    int quantidade;
    struct Produto* proximo;
} Produto;

typedef Produto* ProdutoPtr;

typedef struct Estoque {
    ProdutoPtr primeiro;
    ProdutoPtr ultimo;
} Estoque;

void cria(Estoque* e) {
    e->primeiro = NULL;
    e->ultimo = NULL;
}

bool vazio(Estoque* e) {
    return e->primeiro == NULL;
}

bool esta_no_estoque(Estoque* e, std::string nome) {
    ProdutoPtr p_aux = e->primeiro;

    while (p_aux != NULL) {
        if (p_aux->nome == nome) {
            return true;
        }
        p_aux = p_aux->proximo;
    }

    return false;
}

ProdutoPtr retorna_produto(Estoque* e, std::string nome) {
    ProdutoPtr p_aux = e->primeiro;
```

```

while (p_aux != NULL) {
    if (p_aux->nome == nome) {
        return p_aux;
    }

    p_aux = p_aux->proximo;
}

return NULL;
}

void insere(Estoque* e, Produto p, bool* ok) {
    ProdutoPtr p_novo = new Produto;

    p_novo->nome = p.nome;
    p_novo->preco = p.preco;
    p_novo->quantidade = p.quantidade;
    p_novo->proximo = NULL;

    if (vazio(e)) {
        e->primeiro = p_novo;
    } else {
        if (esta_no_estoque(e, p_novo->nome)) {
            *ok = false;
            return;
        }

        e->ultimo->proximo = p_novo;
    }

    e->ultimo = p_novo;
    *ok = true;
}

void retira(Estoque* e, std::string nome, bool* ok) {
    if (!esta_no_estoque(e, nome)) {
        *ok = false;
        return;
    }

    ProdutoPtr p = retorna_produto(e, nome);
    ProdutoPtr p_aux = e->primeiro;

```

```

    ProdutoPtr anterior = NULL;

    while (p_aux != NULL) {
        if (p_aux->nome == p->nome) {
            if (anterior == NULL) { // Começo da lista
                e->primeiro = e->primeiro->proximo;
                delete p_aux;
            } else if (p_aux->proximo == NULL) { // Fim da lista
                e->ultimo = anterior;
                anterior->proximo = NULL;
                delete p_aux;
            } else { // Meio da lista
                anterior->proximo = p_aux->proximo;
                delete p_aux;
            }

            *ok = 1;
            return;
        } else {
            anterior = p_aux;
            p_aux = p_aux->proximo;
        }
    }

    *ok = false;
}

void altera_quantidade(Estoque* e, std::string nome, int
quantidade, bool* ok) {
    if (!esta_no_estoque(e, nome) || quantidade < 0) {
        *ok = false;
        return;
    }

    ProdutoPtr p_aux = e->primeiro;

    while (p_aux != NULL && p_aux->nome != nome) {
        p_aux = p_aux->proximo;
    }

    p_aux->quantidade = quantidade;
    *ok = true;
}

```

Arquivo main.cpp

```
#include "estoque.hpp"

void limpar_tela() {
    #if defined _WIN32
        system("cls");
    #elif defined (__LINUX__) || defined(__gnu_linux__) ||
defined(__linux__)
        system("clear");
    #elif defined (__APPLE__)
        system("clear");
    #endif
}

void esperar() {
    std::cout << "Pressione qualquer tecla para continuar. . . ";
    std::cin.get();
    std::cout << std::endl;
}

void adicionar_produto(Estoque* e) {
    Produto produto;
    bool ok;

    std::cout << "\nInforme os dados do produto:" << std::endl;
    std::cout << "Nome: ";
    std::cin.ignore();
    getline(std::cin, produto.nome);
    std::cout << "Preço: ";
    std::cin >> produto.preco;
    std::cout << "Quantidade: ";
    std::cin >> produto.quantidade;

    insere(e, produto, &ok);

    if (ok) {
        std::cout << "\nProduto adicionado com sucesso!\n" <<
std::endl;
        std::cin.ignore();
        esperar();
    } else {
```

```

        std::cout << "\nFalha ao adicionar o produto\n" <<
std::endl;
        std::cin.ignore();
        esperar();
    }
}

void retirar_produto(Estoque* e) {
    std::string nome;
    bool ok;

    std::cout << "\nInforme o nome do produto que deseja remover: ";
    std::cin.ignore();
    getline(std::cin, nome);

    retira(e, nome, &ok);

    if (ok) {
        std::cout << "\nProduto removido com sucesso!\n" <<
std::endl;
        esperar();
    } else {
        std::cout << "\nFalha ao remover o produto\n" << std::endl;
        esperar();
    }
}

void imprime_produto(Estoque* e, std::string nome, bool* ok) {
    ProdutoPtr produto = retorna_produto(e, nome);
    if (produto == NULL) {
        *ok = false;
        std::cout << "\nProduto não encontrado no estoque\n" <<
std::endl;
        esperar();
        return;
    }

    std::cout << "\nProduto encontrado no estoque: ";

    std::cout << "\nNome: " << produto->nome << std::endl;
    std::cout << "Preço: " << produto->preco << std::endl;
    std::cout << "Quantidade: " << produto->quantidade << std::endl;

```

```

        std::cout << std::endl;

        *ok = true;
    }

void imprimir_produto(Estoque* e) {
    std::string nome;
    bool ok;

    std::cout << "\nInforme o nome do produto que deseja exibir: ";
    std::cin.ignore();
    getline(std::cin, nome);
    imprime_produto(e, nome, &ok);

    if (ok) {
        esperar();
    }
}

void alterar_quantidade(Estoque* e) {
    std::string nome;
    int quantidade;
    bool ok;

    std::cout << "\nInforme o nome do produto que deseja alterar a
quantidade: ";
    std::cin.ignore();
    getline(std::cin, nome);
    imprime_produto(e, nome, &ok);
    if (!ok) {
        return;
    }

    std::cout << "Informe a nova quantidade do produto: ";
    std::cin >> quantidade;

    altera_quantidade(e, nome, quantidade, &ok);

    if (ok) {
        std::cout << "\nQuantidade alterada com sucesso!\n" <<
std::endl;
        std::cin.ignore();
        esperar();
    } else {

```

```

        std::cout << "\nFalha ao alterar a quantidade\n" <<
std::endl;
        std::cin.ignore();
        esperar();
    }
}

void imprimir_estoque(Estoque* e) {
    if (vazio(e)) {
        std::cout << "\nEstoque vazio\n" << std::endl;
        std::cin.ignore();
        esperar();
        return;
    }

    ProdutoPtr p_aux = e->primeiro;

    do {
        std::cout << "\nNome: " << p_aux->nome << std::endl;
        std::cout << "Preço: " << p_aux->preco << std::endl;
        std::cout << "Quantidade: " << p_aux->quantidade <<
std::endl;

        p_aux = p_aux->proximo;
    } while (p_aux != NULL);

    std::cout << std::endl;
    std::cin.ignore();
    esperar();
}

int main() {
    setlocale(LC_ALL, "pt_BR.UTF-8");

    Estoque e;
    int opcao;

    cria(&e);

    do {
        std::cout << "Controle de estoque" << std::endl;
        std::cout << "\nEscolha uma opcao:" << std::endl;
        std::cout << "1 - Adicionar produto" << std::endl;

```



```

        std::cout << "2 - Alterar quantidade de um produto" <<
std::endl;
        std::cout << "3 - Remover produto" << std::endl;
        std::cout << "4 - Exibir produto" << std::endl;
        std::cout << "5 - Exibir estoque" << std::endl;
        std::cout << "6 - Encerrar execução" << std::endl;
        std::cout << "Digite a opcao desejada: ";
        std::cin >> opcao;

        switch (opcao) {
            case 1:
                adicionar_produto(&e);
                limpar_tela();
                break;
            case 2:
                alterar_quantidade(&e);
                limpar_tela();
                break;
            case 3:
                retirar_produto(&e);
                limpar_tela();
                break;
            case 4:
                imprimir_produto(&e);
                limpar_tela();
                break;
            case 5:
                imprimir_estoque(&e);
                limpar_tela();
                break;
            case 6:
                std::cout << "\nPrograma finalizado!" << std::endl;
                break;
            default:
                std::cout << "\nOpção inválida\n" << std::endl;
                std::cin.ignore();
                esperar();
                limpar_tela();
        }
    } while (opcao != 6);

    return 0;
}

```