Relatório - Sistema de Receitas e Ingredientes

Disciplina: Algoritmos e Estruturas de Dados I
Curso: Ciência da Computação - 3º Período

Introdução

O problema proposto consistia na criação de um sistema para cadastro e gerenciamento de receitas, onde cada receita possui uma lista de ingredientes. Foi solicitado que o sistema utilizasse uma **lista simplesmente encadeada** para as receitas e uma **lista duplamente encadeada** para os ingredientes, com funcionalidades extras como marcar ingredientes essenciais e receitas favoritas.

A solução foi modelada com foco na modularidade e clareza, separando o código em diferentes arquivos (`main.c`, `menu.c`, `receita.c`, `ingrediente.c`, `receita.h`, `ingrediente.h`, `menu.h`) e utilizando estruturas de dados dinâmicas para permitir o crescimento e a reorganização das listas.

Para as **receitas**, foi usada uma estrutura com ponteiro para a próxima receita (lista simplesmente encadeada). Para os **ingredientes**, utilizamos uma estrutura com ponteiros para o anterior e o próximo (lista duplamente encadeada), permitindo fácil inserção, remoção e reordenação dos ingredientes.

Documentação do Código

Estruturas:

- `Receita`: possui nome, status de favorita, ponteiro para ingredientes e ponteiro para a próxima receita.
- `Ingrediente`: possui nome, status de essencial, e ponteiros duplos para anterior e próximo.

Funções principais:

- `criar_receita()`: aloca dinamicamente uma nova receita, solicita nome do usuário e inicializa seus campos.
- `inserir_receita()`: insere uma nova receita na lista de receitas.
- `listar_receitas()`: exibe todas as receitas cadastradas, indicando se são favoritas.
- `adicionar_ingrediente()`: insere um ingrediente em uma receita específica.
- `listar_ingredientes()`: lista os ingredientes de uma receita, marcando os essenciais.

- `marcar_favorita()` e `marcar_essencial()`: atualizam os status das receitas e ingredientes.
- `liberar_receitas()`: libera toda a memória alocada dinamicamente no final do programa.

As funções estão organizadas por responsabilidade e separadas em arquivos para manter a modularização e facilitar a manutenção do código.

Exemplos de Uso

Cadastro de Receita:

Entrada:

1 (inserir nova receita)
Digite o nome da receita: Bolo de Chocolate

Saída:

Receita 'Bolo de Chocolate' adicionada com sucesso.

Adição de Ingrediente:

Entrada:

3 (adicionar ingrediente)
Nome da receita: Bolo de Chocolate
Digite o nome do ingrediente: Ovo
O ingrediente é essencial? (1 = sim, 0 = não): 1

Saída:

Ingrediente 'Ovo' adicionado com sucesso.

Listar Receitas:
2 (listar receitas)

1. Bolo de Chocolate (favorita: nao)

Marcar como favorita:

Entrada:

4 (marcar como favorita)
Nome da receita: Bolo de Chocolate

Saída:

Receita marcada como favorita.

Conclusão

Durante o desenvolvimento deste projeto, foi essencial entender como listas encadeadas funcionam na prática. Implementar uma lista duplamente encadeada dentro de outra lista (simplesmente encadeada) apresentou desafios, principalmente no gerenciamento de memória e na manipulação dos ponteiros.

Aprendi a importância da modularização e do uso de nomes claros para variáveis e funções. Também ficou claro como pequenos erros em ponteiros podem causar falhas graves, exigindo testes constantes.

Acredito que este projeto consolidou meus conhecimentos sobre listas encadeadas e reforçou minha confiança na manipulação de estruturas dinâmicas em C.
