Relatório sobre o Sistema de Gerenciamento de Receitas e Ingredientes

Disciplina: Algoritmos e Estrutura de Dados
Alunos: Bryan Mendes Monteiro, Júlia Marques Santos

---

1. Introdução

O problema proposto foi o desenvolvimento de um sistema de gerenciamento de receitas e ingredientes, no qual o sistema deve permitir a adição, listagem e modificação de receitas, além de permitir a marcação de receitas favoritas e o gerenciamento de ingredientes essenciais para cada receita. A solução foi desenhada de forma modular, utilizando listas encadeadas para armazenar as receitas e os ingredientes.

Foram utilizadas duas estruturas de listas encadeadas para representar os dados:
1. Lista simplesmente encadeada de receitas: Cada receita é armazenada como um nó nesta lista, e cada nó contém um ponteiro para o próximo nó e informações sobre a receita (nome, lista de ingredientes e flag indicando se é favorita).

2. Lista duplamente encadeada de ingredientes: Para cada receita, foi criada uma lista duplamente encadeada de ingredientes, onde cada ingrediente tem um ponteiro para o próximo e anterior ingrediente, facilitando a remoção ou alteração da ordem dos ingredientes. A solução também envolveu a criação de funções para salvar e carregar dados em um arquivo, permitindo que as receitas e ingredientes sejam persistidos e carregados ao iniciar o programa.

2.Documentação do código

O código é dividido em vários arquivos, cada um com funções e responsabilidades específicas. A seguir, descrevo as principais funções implementadas:
1. adicionarReceita:
- Propósito: Adiciona uma nova receita à lista de receitas.

- Implementação: Cria um novo nó (uma nova receita) e a insere no início da lista de receitas
- Parâmetros: Recebe o ponteiro para a lista de receitas e o nome da receita a ser adicionada.

2. buscarReceita:
- Propósito: Busca uma receita na lista de receitas, comparando os nomes.
- Implementação: Percorre a lista de receitas e retorna o ponteiro para a receita encontrada ou NULL caso não seja encontrada.
- Parâmetros: Recebe a lista de receitas e o nome da receita a ser buscada.

3. listarReceitas:
- Propósito: Exibe todas as receitas armazenadas na lista.
- Implementação: Percorre a lista de receitas, imprimindo o nome da receita e, se for o caso, marca se a receita é favorita. Também chama a função listarIngredientes para listar os ingredientes de cada receita.
- Parâmetros: Recebe a lista de receitas.

4. marcarFavorita:
- Propósito: Marca uma receita como favorita.
- Implementação: A função simplesmente altera o valor da flag favorita da receita para 1, indicando que ela é favorita.
- Parâmetros: Recebe um ponteiro para a receita a ser marcada como favorita.

5. adicionarIngrediente:
- Propósito: Adiciona um ingrediente à lista de ingredientes de uma receita.
- Implementação: Cria um novo nó de ingrediente e o insere no início da lista de ingredientes da receita.
- Parâmetros: Recebe o ponteiro para a lista de ingredientes da receita, o nome do ingrediente e um valor que indica se o ingrediente é essencial ou não.

6. listarIngredientes:
- Propósito: Exibe todos os ingredientes de uma receita.

- Implementação: Percorre a lista de ingredientes da receita e imprime o nome do ingrediente, além de indicar se o ingrediente é essencial.
- Parâmetros: Recebe a lista de ingredientes de uma receita.

7. salvarDados:
- Propósito: Salva as receitas e seus ingredientes em um arquivo.
- Implementação: Percorre a lista de receitas e seus ingredientes, salvando essas informações no arquivo salvar_dados.txt. O formato de armazenamento é legível e permite a reconstrução dos dados.
- Parâmetros: Recebe a lista de receitas a ser salva.

8. carregarDados:
- Propósito: Carrega as receitas e ingredientes a partir de um arquivo.
- Implementação: Abre o arquivo salvar_dados.txt e lê as receitas e seus ingredientes, reconstruindo as listas de receitas e ingredientes de acordo com os dados no arquivo.
- Parâmetros: Recebe o ponteiro para a lista de receitas onde os dados carregados serão armazenados.

3.Exemplos de Uso

A seguir, apresento alguns exemplos de operações realizadas pelo sistema, com a entrada de dados e as saídas obtidas.
Exemplo 1: Adicionar Receita e Ingredientes
Entrada:
Nome da receita: Bolo de Chocolate
Nome do ingrediente: Farinha
É essencial? (1 = sim, 0 = não): 1
Nome do ingrediente: Açúcar
É essencial? (1 = sim, 0 = não): 1
Nome do ingrediente: Cacau
É essencial? (1 = sim, 0 = não): 0

Saída:

Receita: Bolo de Chocolate (favorita)

Ingredientes:

- Farinha (essencial)
- Açúcar (essencial)
- Cacau

Exemplo 2:

Marcar Receita como Favorita

Entrada:

Nome da receita a marcar como favorita: Bolo de Chocolate

Saída:

Receita: Bolo de Chocolate (favorita)

Ingredientes:

- Farinha (essencial)
- Açúcar (essencial)
- Cacau

Exemplo 3:

Salvar Dados em Arquivo Após adicionar receitas e ingredientes, ao escolher a opção de salvar os dados, o programa cria o arquivo salvar_dados.txt com o seguinte conteúdo:

Receita: Bolo de Chocolate (favorita)

Ingrediente: Farinha (essencial)

Ingrediente: Açúcar (essencial)

Ingrediente: Cacau

Receita: Macarrão

Ingrediente: Macarrão (essencial)

Ingrediente: Molho de Tomate

Ingrediente: Queijo Ralado

Exemplo 4: Carregar Dados do Arquivo Quando o programa é iniciado novamente e o arquivo salvar_dados.txt está presente, as receitas e ingredientes são carregados automaticamente.

Saída:

Receita: Bolo de Chocolate (favorita)

Ingredientes:

- Farinha (essencial)
- Açúcar (essencial)
- Cacau

Receita: Macarrão Ingredientes:

- Macarrão (essencial)
- Molho de Tomate
- Queijo Ralado

4.Conclusão

A implementação do sistema de gerenciamento de receitas e ingredientes foi bem-sucedida e permitiu que as principais funcionalidades fossem implementadas de maneira eficiente. Durante o processo, alguns desafios foram enfrentados, como a gestão de memória e a manipulação de listas encadeadas, mas esses problemas foram resolvidos com a utilização adequada de ponteiros. As principais lições aprendidas incluem a importância de uma boa organização de dados, a utilização de listas encadeadas para representações dinâmicas e a necessidade de persistir os dados para permitir que o usuário continue a interação com o sistema mesmo após o encerramento do programa. A funcionalidade de salvar e carregar dados foi particularmente importante, pois proporciona uma maneira de manter os dados entre execuções do programa, o que é um requisito essencial para o sistema ser útil em um contexto real.