Documentação Sistema de Gerenciamento do Hortifruti

**Visão Geral**

Este sistema é um software de controle de estoque de hortifruti, permitindo o cadastro, edição, venda, e listagem de produtos divididos em três grupos: Frutas, Verduras e Legumes. Ele armazena os dados de cada produto em um arquivo de texto e oferece funcionalidades para gerenciar as transações de vendas com controle de troco e múltiplas vendas.

**Como Usar**

Ao iniciar o programa, o usuário verá um menu principal com as seguintes opções:

1. **Cadastrar Produto** - Adiciona um novo produto ao sistema, associando-o a um dos grupos (Frutas, Verduras ou Legumes).
2. **Listar Produtos** - Exibe todos os produtos cadastrados e suas informações.
3. **Vender Produto** - Realiza uma venda de produtos, permitindo adicionar múltiplos itens em uma mesma operação e calcular o troco.
4. **Editar Produto** - Modifica as informações de um produto específico.
5. **Sair** - Fecha o programa.

**Variáveis Globais e Estruturas de Dados**

* **Produto (struct)**: Estrutura que armazena informações sobre um produto com os seguintes campos:
  + id: ID único do produto.
  + nome: Nome do produto.
  + preco: Preço unitário do produto.
  + quantidade: Quantidade disponível em estoque.
  + grupo: Grupo ao qual o produto pertence, armazenado como string ("Frutas", "Verduras" ou "Legumes").

**Funções Principais**

1. **void cadastrarProduto(Produto \*\*produtos, int \*n)**
   * Solicita informações para cadastrar um novo produto.
   * Valida o nome, o preço e a quantidade para garantir que sejam válidos.
   * Verifica se o produto já existe no sistema para evitar duplicatas.
   * Armazena o grupo do produto e exibe uma confirmação de cadastro.
2. **void listarProdutos(Produto \*produtos, int n)**
   * Exibe uma tabela com todos os produtos cadastrados, incluindo ID, Nome, Preço (formatado com "R$"), Quantidade e Grupo.
3. **void venderProduto(Produto \*produtos, int n)**
   * Realiza uma venda, permitindo ao usuário adicionar múltiplos produtos.
   * Calcula o valor total da compra e solicita o valor em dinheiro recebido do cliente para calcular o troco.
   * Faz verificações para garantir a entrada válida dos IDs, quantidades e valores.
4. **void editarProduto(Produto \*produtos, int n)**
   * Permite ao usuário alterar a quantidade ou o preço de um produto específico.
   * Solicita o ID do produto e valida a nova quantidade para garantir que seja um número inteiro positivo.
5. **void salvarDados(Produto \*produtos, int n, const char \*arquivo)**
   * Salva todas as informações dos produtos no arquivo especificado.
   * Atualiza o arquivo após cada operação que modifica o estoque ou dados dos produtos.
6. **void carregarDados(Produto \*\*produtos, int \*n, const char \*arquivo)**
   * Carrega os dados dos produtos armazenados no arquivo ao iniciar o programa.
   * Cria as estruturas de dados correspondentes na memória para que sejam manipuladas.
7. **int produtoJaExiste(Produto \*produtos, int n, char nome[])**
   * Verifica se já existe um produto com o mesmo nome, ignorando maiúsculas/minúsculas.
   * Usada para impedir o cadastro de produtos duplicados.
8. **int obterInteiroPositivo()**
   * Função auxiliar que força o usuário a inserir um valor inteiro positivo.
9. **float obterFloatPositivo()**
   * Função auxiliar que força o usuário a inserir um valor float positivo.
10. **char obterCaractereSN()**

* Valida a entrada do usuário para aceitar apenas 's' ou 'n' em perguntas de confirmação.

**Fluxo do Programa**

1. **Carregamento de Dados**: Ao iniciar, o programa lê os dados do arquivo "hortifruti.txt" e armazena na memória.
2. **Menu Principal**: Exibe as opções do menu, cada uma associada a uma funcionalidade.
3. **Processamento de Funções**: Dependendo da escolha do usuário, o programa executa a função apropriada (cadastrar, listar, vender, editar).
4. **Atualização do Arquivo**: Após operações de cadastro, venda ou edição, o programa salva os dados no arquivo.
5. **Saída**: Quando o usuário escolhe sair, o programa libera a memória e encerra.

**Detalhamento de Algumas Funções**

* **Validação das Entradas**: A função obterInteiroPositivo garante que valores como ID, quantidade e opção no menu aceitem apenas números inteiros positivos, evitando entradas inválidas.
* **Simulação de Venda Completa**: Em venderProduto, o usuário pode adicionar múltiplos itens à venda, calcular o valor total e o troco, proporcionando uma simulação próxima de uma venda real.
* **Grupos de Produtos**: Cada produto é associado a um grupo. No listarProdutos, é exibida a categoria (grupo) do produto, facilitando a visualização por tipo.

**Formatação e Exibições**

* **Tabela de Produtos**: A função listarProdutos exibe os produtos com preços formatados com "R$" antes do valor.
* **Entradas Estritamente Validadas**: As funções obterInteiroPositivo e obterFloatPositivo tratam erros de entrada, como letras e caracteres especiais.

**Arquivo de Dados**

O arquivo hortifruti.txt contém uma lista de produtos, armazenando informações como ID, Nome, Preço, Quantidade e Grupo em formato legível. Cada linha representa um produto.

**Considerações Finais**

Este sistema foi projetado para ser extensível. A adição de novos grupos de produtos seria simples e manteria a organização atual do código. Cada função é modular e facilmente adaptável para futuras funcionalidades, como relatórios de vendas ou controle de reabastecimento.