FATEC IPIRANGA – PASTOR ENÉAS TOGNINI
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ISABELLA SANAE KIYATAKE

DISCIPLINA: PROGRAMAÇÃO ESTRUTURADA E MODULAR

PROF. CARLOS VERÍSSIMO

Atividade - N2 - 7: ANÁLISE CRÍTICA DE CÓDIGO

SÃO PAULO

2024

1. Modularização

O código apresenta boa modularização, organizada em funções específicas que desempenham tarefas bem definidas. Essa abordagem melhora a legibilidade, facilita a manutenção e promove a reutilização de código. Exemplos:

- **Funções de Cadastro e Manipulação**: `cadastrarProduto`, `alterarProduto`, `consultarProduto`, `excluirProduto`, etc.
- **Funções Auxiliares**: `imprimirdados`, `imprimirLista`, `descontoProduto`.
- A função principal (`main`) está bem estruturada, funcionando como ponto de entrada, direcionando ações com base nas escolhas do usuário.

2. Elementos Conceituais

O código aborda os seguintes conceitos:

- **Estruturas de Dados**: Uso da `struct Produto` para representar os produtos, encapsulando informações relacionadas (ID, nome, quantidade, valor).
- **Vetores**: Armazena os produtos em um vetor de tamanho fixo (`MAXPRODUTOS`).
- **Validação de Entrada**: Existem controles para evitar entradas inválidas, como valores negativos para preços ou estoques.
- **Modularidade**: Separação lógica entre o processamento dos dados e a interface com o usuário (funções que manipulam a lista e a função `main` que apresenta o menu e obtém entradas).

3. Elementos de Negócio (Requisitos)

O sistema atende a requisitos de um gerenciamento básico de estoque em um ambiente comercial:

1. **Cadastrar produtos**: Inserir novos produtos com dados como nome, quantidade e preço.
2. **Alterar produtos**: Atualizar as informações de um produto existente.
3. **Consultar produtos**: Visualizar dados de um produto específico ou listar todos os produtos cadastrados.
4. **Excluir produtos**: Remover um produto do sistema.
5. **Vender produtos**: Reduzir o estoque com base nas vendas, calculando o valor total.
6. **Aplicar desconto**: Ajustar o preço de um produto com base em uma porcentagem de desconto.

4. Regras Explícitas

As regras explícitas estão diretamente implementadas e facilmente identificáveis no código:

- **ID do Produto**: Gerado automaticamente com base no índice no vetor.
- **Estoque e Preço**: Não podem ser negativos. Há validações específicas que impedem a entrada de valores inválidos.

- **Quantidade para Venda**: Deve ser maior que zero, e o estoque precisa ser suficiente.
- **Desconto**: Deve ser aplicado em forma de porcentagem e calculado corretamente.

5. Regras Implícitas

Regras não declaradas diretamente, mas que emergem do comportamento ou design do sistema:

- **Limite de Produtos**: O sistema suporta no máximo 50 produtos, pois o vetor `listaProdutos` tem tamanho fixo (`MAXPRODUTOS`). Não há tratamento explícito para quando esse limite é atingido.
- **Persistência de Dados**: Não há mecanismo para salvar ou carregar os dados entre execuções, implicando que o uso do sistema é exclusivamente em memória e os dados são descartados ao encerrar o programa.
- **Identificação Única**: O ID do produto é gerado sequencialmente e depende da posição no vetor, o que evita duplicações, mas é frágil frente a exclusões, já que IDs podem mudar.

Sugestões de Melhoria

1. **Persistência de Dados**:
    - Incluir funcionalidades para salvar e carregar produtos de um arquivo, garantindo que os dados sejam preservados entre sessões.
2. **Mensagens de Erro Mais Informativas**:
    - Algumas mensagens poderiam ser mais claras para o usuário final.
3. **Limitação do Número de Produtos**:
    - Implementar validação explícita para avisar o usuário caso o limite de 50 produtos seja atingido.
4. **Estrutura Dinâmica**:
    - Utilizar estruturas dinâmicas (como listas encadeadas) para gerenciar produtos, eliminando o limite fixo do vetor.

Conclusão

O código atende bem aos requisitos básicos de um sistema de gerenciamento de produtos em um ambiente comercial. Ele demonstra boa modularização e uso de práticas como validação de entrada e encapsulamento de dados. No entanto, a ausência de persistência de dados e a limitação fixa de produtos podem restringir sua aplicabilidade em contextos mais amplos.