

ATIVIDADE N2-7- ANÁLISE CRÍTICA DO CÓDIGO

Letícia Sudan e Fernando Palmieri - 2º Semestre de ADS – Vespertino

Pontos Positivos:

- ☺ O código é modularizado.
- ☺ As funções do menu: **Cadastrar**, **Consultar**, **Alterar**, **Excluir**, **Vender**, **Dar Desconto**, **Consultar Lista** e **Encerrar o programa** estão funcionando perfeitamente.
- ☺ O código utiliza ponteiros para manipular toda sua estrutura.
- ☺ O código utiliza struct para realizar o cadastro de produtos.
- ☺ Os tipos de dados declarados no struct estão corretos:
 - ID (inteiro) - Automático
 - Nome do produto (string)
 - QuantidadeEmEstoque (int)
 - ValorDoProduto (double)
- ☺ O código possui uma função para cada CRUD.
- ☺ As funções Alterar, Consultar e Excluir Produto trabalham por ID.

Pontos Negativos:

- ☹ Menu não se encontra modularizado.
Foi criada uma função para isso:

```
// Função que exibe o menu e captura a opção
void exibirMenu(int *opcao) {
    printf("----- * MENU * -----\\n");
    printf("1. Cadastrar produto\\n");
    printf("2. Alterar dados do produto\\n");
    printf("3. Consultar produto\\n");
    printf("4. Excluir Produto\\n");
    printf("5. Consultar lista de produtos\\n");
    printf("6. Vender produto\\n");
    printf("7. Dar desconto a um produto\\n");
    printf("8. Encerrar o programa\\n");
    printf("Escolha uma opcao: ");
    scanf(" %d", opcao);
}
```

- ☹ Código não possui tratamento de erro para os inputs das funções.

Ao cadastrar um produto, o programa permite que o usuário insira caracteres alfabéticos nos campos de QuantidadeEmEstoque e ValorDoProduto, sendo que deveriam ser aceitos apenas valores numéricos:

```
Digite o nome do produto: videogame
Digite a quantidade em estoque do produto (Somente numeros naturais): g
Digite o valor do produto: Produto adicionado com sucesso!
```

Dessa forma, ao consultar o produto podemos ver que as informações são guardadas de forma indesejada:

```
Digite o ID do produto que deseja consultar: 1
ID: 1
Nome: videogame
Quantidade em estoque: 1590065232
Valor do produto: 0.00
```

Para corrigirmos isso vamos utilizar os seguintes trechos de código:

```
while (scanf("%d", &produto->quantidadeEmEstoque) != 1 || produto->quantidadeEmEstoque < 0) {
    while (getchar() != '\n'); // Limpar o buffer de entrada
    printf("Entrada invalida! A quantidade deve ser um numero inteiro positivo.\n");
    printf("Digite a quantidade em estoque do produto (Somente numeros naturais): ");
}
```

```
while (scanf("%lf", &produto->valorDoProduto) != 1 || produto->valorDoProduto < 0) {
    while (getchar() != '\n'); // Limpar o buffer de entrada
    printf("Entrada invalida! O valor deve ser um numero valido e positivo.\n");
    printf("Digite o valor do produto: ");
}
```

Essa parte do código ficará assim dentro da função cadastrarProduto:

```
// Garantir que a quantidade seja um número inteiro
printf("Digite a quantidade em estoque do produto (Somente numeros naturais): ");
while (scanf("%d", &produto->quantidadeEmEstoque) != 1 || produto->quantidadeEmEstoque < 0) {
    while (getchar() != '\n'); // Limpar o buffer de entrada
    printf("Entrada invalida! A quantidade deve ser um numero inteiro positivo.\n");
    printf("Digite a quantidade em estoque do produto (Somente numeros naturais): ");
}

// Garantir que o valor seja um número flutuante válido
printf("Digite o valor do produto: ");
while (scanf("%lf", &produto->valorDoProduto) != 1 || produto->valorDoProduto < 0) {
    while (getchar() != '\n'); // Limpar o buffer de entrada
    printf("Entrada invalida! O valor deve ser um numero valido e positivo.\n");
    printf("Digite o valor do produto: ");
}
```

Agora ao executar, o tratamento de erros funciona corretamente:

```
Digite o nome do produto: videogame
Digite a quantidade em estoque do produto (Somente numeros naturais): g
Entrada invalida! A quantidade deve ser um numero inteiro positivo.
Digite a quantidade em estoque do produto (Somente numeros naturais): 5
Digite o valor do produto: g
Entrada invalida! O valor deve ser um numero valido e positivo.
Digite o valor do produto: 2000
Produto adicionado com sucesso!
```

E ao fazer uma consulta vemos que está tudo certo:

```
Digite o ID do produto que deseja consultar: 1
ID: 1
Nome: videogame
Quantidade em estoque: 5
Valor do produto: 2000.00
```

Agora realizamos o mesmo procedimento nos inputs de outras funções.

- ☹ Código possui alguns erros de acentuação na exibição no console.
O problema foi corrigido removendo acentuação das palavras