

Projeto: Controle de Estoque - Algoritmos e Programação I

Integrantes:

Pedro Henrique Leite – 32336926

Gabriel Nobrega Neri – 32366851

Cauê Lemos Garcia – 32315104

Relatório de Desenvolvimento - Controle de Estoque

- **Introdução:**

O presente relatório descreve o desenvolvimento de um sistema de controle de estoque, realizado como parte de um trabalho em grupo na disciplina de Algoritmos e Programação I. Neste projeto, focamos na implementação de funcionalidades de cadastro, consulta, atualização e exclusão de produtos, além da geração de relatórios dos produtos cadastrados. Utilizamos a linguagem de programação Python para desenvolver o sistema.

Decisões de Implementação:

- **Estrutura de Dados:**

Decidimos utilizar uma lista de dicionários para armazenar os produtos. Cada dicionário representa um produto e possui os campos "nome", "código" e "quantidade". Essa escolha proporciona uma estrutura flexível e eficiente para armazenar múltiplos produtos, além de permitir um acesso conveniente às informações de cada produto.

- **Funções Modulares:**

Optamos por implementar funções separadas para cada operação do sistema, como cadastro, consulta, atualização e exclusão de produtos. Essa abordagem modulariza o código, facilitando sua manutenção, reutilização e compreensão. Cada função realiza uma tarefa específica, contribuindo para a organização e clareza do programa.

- **Menu Interativo:**

Decidimos implementar um menu interativo para permitir ao usuário escolher as ações a serem executadas. Utilizamos um loop contínuo para exibir o menu repetidamente até que o usuário escolha a opção de encerrar o programa. Essa abordagem proporciona uma interação amigável com o usuário e melhora a usabilidade do sistema.

- **Entrada e Validação de Dados:**

Utilizamos o comando `input()` para obter dados fornecidos pelo usuário, como nome, código e quantidade do produto. Implementamos também validações simples para garantir a integridade das entradas do usuário. Por exemplo, verificamos se um produto existe antes de realizar sua atualização ou exclusão, evitando erros e inconsistências nos dados.

- **Mensagens de Feedback:**

Incluimos mensagens informativas para fornecer feedback ao usuário durante a execução do programa. Ao cadastrar, atualizar ou excluir um produto com sucesso, exibimos mensagens de confirmação. Além disso, ao ocorrer a situação em que um produto não é encontrado, apresentamos mensagens de erro adequadas, melhorando a experiência do usuário.

- **Encerramento do Programa:**

Implementamos uma função para encerrar o programa de forma adequada. Ao solicitar o encerramento, exibimos uma mensagem de confirmação ao usuário. Em seguida, encerramos a execução do script utilizando a biblioteca `sys`, garantindo uma finalização adequada do programa.

- **Conclusão:**

Neste projeto de controle de estoque, aplicamos as decisões de implementação mencionadas anteriormente para criar um sistema funcional e interativo. A estrutura de dados, as funções modulares, o menu interativo, a entrada e validação de dados, as mensagens de feedback e o encerramento adequado do programa contribuíram para um sistema eficiente e amigável ao usuário. Com essas funcionalidades, o sistema permite o gerenciamento de produtos, facilitando o controle de estoque de forma simples e intuitiva.

No decorrer do trabalho em grupo, adquirimos conhecimentos em programação e aprimoramos nossa capacidade de trabalho colaborativo. Além disso, compreendemos a importância de uma boa estruturação do código e das decisões adequadas de implementação para o desenvolvimento de sistemas de software eficazes.

Printscreen com os testes de execução de todas as opções do Menu

Teste de execução da opção de cadastro do produto:

Como podemos ver na imagem, cadastramos um produto chamado 'Chiclete Trident Menta' com o código de '1' e a quantidade '10' com sucesso.

```
---- Menu - Controle de Estoque ----
1. Cadastrar produto
2. Consultar produto
3. Atualizar produto
4. Excluir produto
5. Relatório dos produtos
6. Encerrar o programa
Digite o número da opção desejada: 1
-----
Digite o nome do produto a ser cadastrado: Chiclete Trident Menta
Digite o código do produto a ser cadastrado: 1
Digite a quantidade deste produto a ser cadastrado: 10
Produto cadastrado com sucesso!
```

Teste de execução da opção de consulta do produto:

Como podemos ver na imagem, consultamos o produto cadastrado com o código '1', que é o 'Chiclete Trident Menta', sendo assim, mostrando o nome, código e quantidade do produto.

```
---- Menu - Controle de Estoque ----
1. Cadastrar produto
2. Consultar produto
3. Atualizar produto
4. Excluir produto
5. Relatório dos produtos
6. Encerrar o programa
Digite o número da opção desejada: 2
-----
Digite o código do produto a ser consultado: 1
Produto encontrado:
→ Nome: Chiclete Trident Menta
→ Código: 1
→ Quantidade: 10
```

Teste de execução da opção de atualizar o produto:

Como podemos ver na imagem, atualizamos o produto chamado 'Chiclete Trident Menta'. Mudamos seu nome para 'Chiclete Trident Morango' com uma nova quantidade de '15'.

```
---- Menu - Controle de Estoque ----
1. Cadastrar produto
2. Consultar produto
3. Atualizar produto
4. Excluir produto
5. Relatório dos produtos
6. Encerrar o programa
Digite o número da opção desejada: 3
-----
Digite o código do produto a ser atualizado: 1
Produto encontrado:
→ Nome: Chiclete Trident Menta
→ Código: 1
→ Quantidade: 10
Digite o novo nome do produto a ser atualizado: Chiclete Trident Morango
Digite a nova quantidade do produto a ser atualizado: 15
Produto atualizado com sucesso!
```

Teste de execução da opção do relatório dos produtos:

Como podemos ver na imagem, o relatório de produtos mostra pra gente o nosso único produto cadastrado, lembrando que, o programa suporta mais de um produto cadastrado. Nosso produto 'Chiclete Trident Morango' como um novo nome, e quantidade, graças a opção de atualização do produto.

```
---- Menu - Controle de Estoque ----
1. Cadastrar produto
2. Consultar produto
3. Atualizar produto
4. Excluir produto
5. Relatório dos produtos
6. Encerrar o programa
Digite o número da opção desejada: 5
-----
Relatório de Produtos:
→ Nome: Chiclete Trident Morango
→ Código: 1
→ Quantidade: 15
```

Teste de execução da opção de excluir o produto:

Como podemos ver na imagem, excluímos o nosso único produto chamado 'Chiclete Trident Morango'. E em seguida rodamos a opção do relatório de produtos para mostrar que o produto foi realmente excluído.

```
---- Menu - Controle de Estoque ----
1. Cadastrar produto
2. Consultar produto
3. Atualizar produto
4. Excluir produto
5. Relatório dos produtos
6. Encerrar o programa
Digite o número da opção desejada: 4
-----
Digite o código do produto a ser excluído: 1
Produto excluído com sucesso!

---- Menu - Controle de Estoque ----
1. Cadastrar produto
2. Consultar produto
3. Atualizar produto
4. Excluir produto
5. Relatório dos produtos
6. Encerrar o programa
Digite o número da opção desejada: 5
-----
Não há produtos cadastrados.
```

Teste de execução da opção de encerrar o programa:

Como podemos ver na imagem abaixo, utilizamos a opção de encerrar o programa para fechar o mesmo.

```
---- Menu - Controle de Estoque ----
1. Cadastrar produto
2. Consultar produto
3. Atualizar produto
4. Excluir produto
5. Relatório dos produtos
6. Encerrar o programa
Digite o número da opção desejada: 6
-----
Tem certeza de que deseja encerrar o programa? (S/N): Sim
Entrada inválida. Responda com 'Sim' ou 'Não': S
O programa está sendo encerrado...
```

Apêndice

```
# Projeto: Controle de Estoque - Algoritmos e Programação I
# Integrantes: Pedro Henrique Leite - Gabriel Nobrega Neri - Cauê Lemos Garcia

# Importa uma biblioteca que permite utilizar o recurso "sys.exit()", que foi
# usado para a criação da função de encerrar o programa.
import sys

# Função para cadastrar um novo produto.
def cadastrar_produto(lista_produtos):
    # Solicita o nome, código e quantidade do novo produto ao usuário.
    nome = input("Digite o nome do produto a ser cadastrado: ")
    codigo = input("Digite o código do produto a ser cadastrado: ")
    quantidade = int(input("Digite a quantidade deste produto a ser
cadastrado: "))

    # Cria um dicionário para armazenar as informações do produto.
    produto = {"nome": nome, "codigo": codigo, "quantidade": quantidade}

    # Adiciona o produto cadastrado à lista de produtos.
    lista_produtos.append(produto)

    # Exibe uma mensagem de sucesso.
    print("Produto cadastrado com sucesso!", "\n")

# Função para consultar um produto existente.
def consultar_produto(lista_produtos):
    # Solicita o código do produto a ser consultado.
    codigo = input("Digite o código do produto a ser consultado: ")

    # Variável para indicar se o produto foi encontrado ou não.
    encontrado = False

    # Percorre a lista de produtos em busca do produto com o código fornecido.
    for produto in lista_produtos:
        if produto["codigo"] == codigo:
            # Exibe as informações do produto encontrado.
            print("Produto encontrado:")
            print("→ Nome:", produto["nome"])
            print("→ Código:", produto["codigo"])
            print("→ Quantidade:", produto["quantidade"], "\n")
            encontrado = True
            break

    # Se o produto não foi encontrado, exibe uma mensagem de erro e retorna ao
    # menu.
    if not encontrado:
        print("Produto não encontrado.", "\n")
```

```

# Função para atualizar as informações de um produto existente.
def atualizar_produto(lista_produtos):
    # Solicita o código do produto a ser atualizado.
    codigo = input("Digite o código do produto a ser atualizado: ")

    # Variável para indicar se o produto foi encontrado ou não.
    encontrado = False

    # Percorre a lista de produtos em busca do produto com o código fornecido.
    for produto in lista_produtos:
        if produto["codigo"] == codigo:
            # Exibe as informações do produto encontrado.
            print("Produto encontrado:")
            print("→ Nome:", produto["nome"])
            print("→ Código:", produto["codigo"])
            print("→ Quantidade:", produto["quantidade"])

            # Solicita as novas informações para atualizar o produto, no caso
            # o nome e a quantidade.
            novo_nome = input("Digite o novo nome do produto a ser atualizado: ")

            if novo_nome:
                produto["nome"] = novo_nome

            nova_quantidade = input("Digite a nova quantidade do produto a ser
atualizado: ")
            if nova_quantidade:
                produto["quantidade"] = int(nova_quantidade)

            # Exibe uma mensagem de sucesso.
            print("Produto atualizado com sucesso!", "\n")
            encontrado = True
            break

    # Se o produto não foi encontrado, exibe uma mensagem de erro e retorna ao
    # menu.
    if not encontrado:
        print("Produto não encontrado.", "\n")

# Função para excluir um produto do estoque.
def excluir_produto(lista_produtos):
    # Solicita o código do produto a ser excluído.
    codigo = input("Digite o código do produto a ser excluído: ")

    # Variável para indicar se o produto foi encontrado ou não.
    encontrado = False

    # Percorre a lista de produtos em busca do produto com o código fornecido.

```

```

for produto in lista_produtos:
    if produto["codigo"] == codigo:
        # Remove o produto da lista de produtos.
        lista_produtos.remove(produto)

        # Exibe uma mensagem de sucesso
        print("Produto excluído com sucesso!", "\n")
        encontrado = True
        break

# Se o produto não foi encontrado, exibe uma mensagem de erro e retorna ao
menu.
if not encontrado:
    print("Produto não encontrado.", "\n")

# Função para gerar um relatório dos produtos cadastrados.
def relatorio_produtos(lista_produtos): # Função para gerar um relatório dos
produtos cadastrados.
    # Verifica se há produtos cadastrados.
    if not lista_produtos:
        # Se não houver produtos cadastrados, exibe uma mensagem informando.
        print("Não há produtos cadastrados.", "\n")
    else:
        # Se houver produtos cadastrados, exibe o relatório de produtos.
        print("Relatório de Produtos:")
        for produto in lista_produtos:
            print("→ Nome:", produto["nome"])
            print("→ Código:", produto["codigo"])
            print("→ Quantidade:", produto["quantidade"], "\n")

# Função para encerrar o programa.
def encerrar_programa():
    # Solicita a confirmação do usuário para encerrar o programa.
    confirmacao = input("Tem certeza de que deseja encerrar o programa? (S/N):
").upper()

    # Verifica se a entrada do usuário é válida.
    while confirmacao not in ("S", "N"):
        confirmacao = input("Entrada inválida. Responda com 'Sim' ou 'Não':
").upper()

    # Se o usuário confirmar, encerra o programa. Caso contrário, retorna ao
menu principal.
    if confirmacao == "S":
        print("O programa está sendo encerrado...", "\n")
        sys.exit()
    elif confirmacao == "N":
        print("Certo. Voltando para o menu...", "\n")

```



```
# Função principal que exibe o menu e realiza as operações escolhidas pelo
usuário.
def menu(lista_produtos):
    while True:
        print("---- Menu - Controle de Estoque ----")
        print("1. Cadastrar produto")
        print("2. Consultar produto")
        print("3. Atualizar produto")
        print("4. Excluir produto")
        print("5. Relatório dos produtos")
        print("6. Encerrar o programa")

        # Solicita ao usuário a opção desejada.
        opcao = input("Digite o número da opção desejada: ")

        print("-----")

        # Com base na opção selecionada, chama a função correspondente.
        if opcao == "1":
            cadastrar_produto(lista_produtos)
        elif opcao == "2":
            consultar_produto(lista_produtos)
        elif opcao == "3":
            atualizar_produto(lista_produtos)
        elif opcao == "4":
            excluir_produto(lista_produtos)
        elif opcao == "5":
            relatorio_produtos(lista_produtos)
        elif opcao == "6":
            encerrar_programa()
        else:
            print("Opção inválida. Por favor, digite uma opção válida.", "\n")

# Cria uma lista vazia para armazenar os produtos.
lista_produtos = []

# Chama a função de menu para iniciar o programa.
menu(lista_produtos)
```