

```

from tkinter.messagebox import showinfo

# Produtos é a matriz responsável por conter meus produtos com relação a: Código, Nome, preço, e Estoque.
produtos = [[1,"coca-cola", 3.75,2],
             [2,"pepsi",3.67,5],
             [3,"monster",9.96,1],
             [4,"café",1.25,100],
             [5,"redbull",13.99,2]]

# Encontrar o código, recebe o código e verifica se existe esse produto apartir do código que o usuário
# forneceu.

def encontrar_produto(codigo):
    for i in range(len(produtos)):
        if produtos[i][0] == codigo:
            return produtos[i]
    return None

"""Verificar estoque recebe o código do usuário e recebe a quantidade preferida em relação ao produto
escolhido.
fazendo a verificação de se o estoque está disponível ou não, se não for, ele me diz que o estoque não está
disponível
caso contrário, mostra o estoque, o preço do produto, e também o seu valor final, em relação a quantidade que
o usuário quiz.
retornando assim o preço e quantidade, caso não haja o produto ele retorna um break, e finaliza o código
retornando None, que não
foi encontrado o produto.
"""

def verificar_estoque(codigo, quantidade):
    preco = ""
    for i in range(len(produtos)):
        if produtos[i][0] == codigo:
            preco = produtos[i][2]
            valor_f = produtos[i][2]*quantidade
            if produtos[i][3] >= quantidade:
                print(f"O Estoque é {produtos[i][3]}, e está disponível!")
                print(f"Preço R$ {preco}, valor Final: R$ {valor_f}")
                return preco, quantidade
            elif produtos[i][3] < quantidade:
                break
    return None

```

```
"""
```

def compra recebe preço e quantidade, para verificar o valor final o preço final que o usuário deve pagar, também é verificado se

```
"""
```

```
def compra(preco,quantidade):
    valor_final = preco * quantidade
    pagamento = float(input("Digite o valor a pagar: "))
    troco = ""
    while True:
        if pagamento > valor_final:
            troco = pagamento - valor_final
            troco = round(troco, 2)
            print(f"O valor final ficou R$: {valor_final}, valor Pago: R$ {pagamento}, Troco: {troco}")
            break
        else:
            print("Valor insuficiente!")
            pagamento = float(input("Digite o valor a pagar suficiente: "))
            continue

    return troco, valor_final
```

```
"""
```

Função troco, tem como objetivo calcular o troco, verificando quantas vezes cada cédula ou moeda, pode conter no troco final

se eu preciso dar o troco de 2,50 cabe uma nota de 2 e 1 moeda de 50 centavos, além disso os valores como float, são convertidos

int inicialmente, e depois dentro do laço são convertidas a cada interação.

```
"""
```

```
def troco_da_compra(troco,valor_final):
    notas_e_moedas = [100, 50, 20, 10, 5, 2, 1,0.50,0.25,0.10, 0.05, 0.01]

    resultado = {}
    troco = round(troco, 2)

    for valor in notas_e_moedas:
        qtd = int(troco// valor)
        if qtd > 0:
            resultado[valor] = qtd
            troco -= qtd * valor
```

```
troco = round(troco, 2)
print(f"{resultado[valor]} nota de, {valor} reais,")
```

```
elif qtd < 0:
    print("Não a notas e moedas disponíveis")
    print("Compra cancelada")
```

"""

A função atualizar estoque tem como objetivo atualizar o estoque em relação ao código e quantidade fornecidos da compra do cliente.

"""

```
def atualizar_estoque(codigo, quantidade):
    estoque = ""
    for i in range(len(produtos)):
        if produtos[i][0] == codigo:
            estoque = produtos[i][3]
            estoque -= quantidade
            produtos[i][3] = estoque
    print(f"Estoque Atualizado, {estoque}")
```

```
    return estoque, produtos
```

"""

A função adicionar produto, adiciona um novo produto, nome, preço, id, e estoque, e é adicionado numa lista, para ser adicionado na matriz produtos.

"""

```
def adicionar_produto():
    id_do_produto = int(input("Digite o id do produto: "))
    nome_do_produto = input("Digite o nome do produto: ")
    preco = float(input("Digite o valor do produto: "))
    estoque_do_produto = int(input("Digite o estoque do produto: "))
    novo_produto = [id_do_produto, nome_do_produto, preco, estoque_do_produto]
    produtos.append(novo_produto)
```

"""

Remover o produto, a função tem como objetivo fazer a remoção do produto por completo, em relação a sua posição linha na matriz dos produtos.

```
"""
```

```
def remover_produto():
    codigo = int(input("Digite o código do produto: "))
    for i in range(len(produtos)):
        if produtos[i][0] == codigo:
            print("Você está prestes a apagar um produto!")
            escolha = input("Deseja apagar o produto? [S/N] ").strip().lower()
            if escolha == "s":
                produtos.remove(produtos[i])
                break
            elif escolha == "n":
                print("Remoção cancelada!")
    else:
        print("Produto não encontrado!")
    return None
```

```
"""
```

A função editar produto, faz a edição do produto correspondente ao código dele fornecido e buscado na matriz dos produtos  
nisso é adicionado ou pode ser mantido, dependente da escolha do admin, e depois é salvo esses valores numa lista, e adicionado  
essa lista na minha matriz.

```
"""
```

```
def editar_produto():
    codigo = int(input("Digite o código do produto: "))
    for i in range(len(produtos)):
        if produtos[i][0] == codigo:
            print(f"Id : {produtos[i][0]}, Nome: {produtos[i][1]}, valor:R$ {produtos[i][2]}, Estoque: {produtos[i][3]}")
            escolha = input("Deseja editar o produto? [S/N] ").strip().lower()
            if escolha == "s":
                novo_nome = input("Digite o nome do produto: ")
                novo_preco = float(input("Digite o valor do produto: "))
                novo_estoque = int(input("Digite o novo estoque: "))
                produtos[i] = [produtos[i][0], novo_nome, novo_preco, novo_estoque]
                break
            else:
                print("Edição Negada!!!!!!")
    else:
        return None
```

"""

A função menu é o coração do meu código, que chamada funções como ver lista de produtos, a função executar máquina, que é

própria pra interação do usuário, chamada o mod admin, e a opção de sair, ou seja finalizar o programa.

"""

def menu():

while True:

print("\n===== MENU PRINCIPAL =====")

print("1 - Ver lista de produtos")

print("2 - Comprar produto")

print("3 - Acessar modo administrador")

print("4 - Sair")

print("=====")

escolha = input("Escolha uma opção: ")

if escolha == "1":

mostra\_lista\_produtos()

elif escolha == "2":

executar\_maquina() # sua função de venda

elif escolha == "3":

senha = input("Digite a senha do administrador: ")

if senha == "admin2033":

menu\_admin()

else:

print("Senha incorreta!")

elif escolha == "4":

print("Programa finalizado.")

return None

else:

print("Opção inválida. Tente novamente.")

"""

A função menu admin é responsável pelo menu de administrador, e só é chamado quando a função principal chamada menu,

a partir da escolha de N 3, entra no modo admin, que possui uma senha. No menu admin a cada escolha se for verdadeiro, será

chamada uma função própria do administrador.

"""

```

def menu_admin():
    while True:
        print("\n===== MENU PRINCIPAL =====")
        print("1 - Ver lista de produtos")
        print("2 - adicionar Produto")
        print("3 - Remover Produto")
        print("4 - editar_produto")
        print("5 - Sair do Modo Admin")
        print("=====")

        escolha = int(input("Digite uma opção: "))
        if escolha == 1:
            mostra_lista_produtos()
        elif escolha == 2:
            adicionar_produto()
        elif escolha == 3:
            remover_produto()
        elif escolha == 4:
            editar_produto()
        elif escolha == 5:
            print("Saindo do Modo Admin!!")
            menu()
        else:
            print("Opção inválida, tente novamente!")

```

"""Validar código, valida se meu código não é um valor que não seja string numérica, é usado para validar para caso o usuário

colocar um valor inválido, que não seja um número, que inicialmente é int, depois é convertido para int.

"""

```

def validar_codigo():
    codigo_produto = " "
    while not codigo_produto.isnumeric():
        codigo_produto = input("Digite o código do produto: ")

    return int(codigo_produto)

```

# Essa função abaixo dentro há um laço que após a finalização de um compra sempre retorna para o Menu.

```

def retorno_iniciar():
    while True:
        menu()

```

```
        continue
    """
```

A função executar máquina é responsável pela execução do meu programa do meu usuário recebendo as funções de validação de código, estoque, atualização de estoque.

Usando o método de desempacotamento de função, como:

produto, variável recebe o valor da minha função Verificar estoque, que retorna ou o produto, ou retorna None se não encontrar o produto.

além disso é armazenado os valores das funções de atualizar estoque, função da compra, do troco, toda parte que cabe

ao usuário é feito e chamado pela função executar máquina.

```
    """
```

```
def executar_maquina():
    while True:
        codigo = validar_codigo()
        produto = encontrar_produto(codigo)
        quantidade_de_produtos = int(input("Quantidade de produtos: "))
        resultado = verificar_estoque(codigo, quantidade_de_produtos)
        if resultado is not None:
            estoque = atualizar_estoque(codigo, quantidade_de_produtos)
        else:
            print("Estoque indisponível")
            continue
        if produto is None:
            print("Nenhum produto foi encontrado!")

        else:
            mostrar_produto(produto)
            preco, quantidade = resultado
            troco, valor_final = compra(preco, quantidade)
            troco_da_compra(troco, valor_final)
            mostra_lista_produtos()
            escolha = input("Deseja sair do ambiente de compra [S/N]?: ").strip().lower()
            if escolha == "s":
                return menu()
            elif escolha == "n":
                break

    return None
```

```
    """
```

A função mostrar produto mostra o produto bem formatado mostrando seu respectivo valor, em realção ao produto selecionado pelo usuário mostrando ele na tela.

```
"""
```

```
def mostrar_produto(produto):  
    print(f"Produto: {produto[1]}, Valor: R$ {produto[2]}, Estoque: {produto[3]}")
```

```
"""
```

A função mostrar lista de produtos mostra a lista de produtos na forma de uma matriz de forma encadeada.

```
"""
```

```
def mostra_lista_produtos():  
    print("Lista de produtos!")  
    print(" ")  
    for item in produtos:  
        print(item)  
    print(" ")
```

```
mostra_lista_produtos()  
retorno_iniciar()  
executar_maquina()
```