

Perfumaria

Herbarium

Isabela, Maria Clara Fonseca, Maria Clara
Sibernagel, Rebeca Mendes



E-commerce

Nossa loja e-commerce de perfumes é uma plataforma dedicada à venda de perfumes e produtos relacionados com cuidados com a pele, higiene e cuidados capilares. Ela permite que os clientes descubram, escolham e comprem os nossos produtos de forma prática e segura.



Login/logout - Vetor

- Utilizado para armazenar os dados de login/logout dos usuários;
- Permite acesso rápido por índice.

```
print(f"usuário logado: {usuario_logado.capitalize()}")
break

else:
    if cont >= 3:
        print("Número máximo de tentativas excedido. Saindo...")
        break

    print("\nLogin")
    print("1. Entrar")
    print("2. Criar novo usuário")
    escolha = input("Escolha uma opção: ")

    if escolha == '1':
        usuario_input = input('Digite seu usuário: ').lower()
        senha_input = input('Digite sua senha: ')

        if usuario_input in v_usuarios:
            idx = np.where(v_usuarios == usuario_input)[0][0]
            if v_senhas[idx] == senha_input:
                print(f"Acesso permitido. Seja bem-vindo(a), {usuario_input.capitalize()}!")
                usuario_logado = usuario_input
                cont = 0
            else:
                print("Senha incorreta.")
                cont += 1
        else:
            print("Usuário não encontrado.")
            cont += 1

    elif escolha == '2':
        novo_usuario = input("Digite o novo nome de usuário: ").lower()
        if novo_usuario in v_usuarios:
            print("Esse nome de usuário já existe. Tente outro.")
        else:
            nova_senha = input("Digite a nova senha: ")
            v_usuarios = np.append(v_usuarios, novo_usuario)
            v_senhas = np.append(v_senhas, nova_senha)
            usuario_logado = novo_usuario
            cont = 0
            print(f"Usuário {novo_usuario.capitalize()} criado com sucesso e logado!")
            break

    else:
        print("Opção inválida!")
```

Código

```
logout

while True:
    if usuario_logado:
        opcao = input(f"\nVocê está logado como {usuario_logado.capitalize()}. Deseja fazer logout? (s/n): ").lower()
        if opcao == 's':
            print(f"Usuário {usuario_logado.capitalize()} deslogado com sucesso!")
            usuario_logado = None
        elif opcao == 'n':
            print("Continuando logado.")
            break
        else:
            print("Resposta inválida. Digite 's' ou 'n'.")
    else:
        print("Nenhum usuário está logado.")
        break
```

Dados do usuário logado - Matriz

- Armazena informações dos usuários logados, organizadas em linhas e colunas.
- Cada linha representa um usuário, cada coluna uma informação (nome, e-mail, endereço, etc.).

```
#Dados de usuários logados
usuarios = [
    ["ana", "1234", True],
    ["bruno", "abcd", True],
    ["carla", "senha", True]
]

while True:
    nome = input("Digite o nome do usuário (ou 'sair' para encerrar): ")
    if nome.lower() == "sair":
        break

    senha = input("Digite a senha: ")
    logado_status = False
    for u in usuarios:
        if u[0].lower() == nome.lower():
            logado_status = True
            if u[1] == senha:
                if u[2]:
                    print(f"{u[0]} está logado.")

            else:
                print(f"{u[0]} não está logado.")
        else:
            print(f"{u[0]} Senha incorreta.")
            break

    if not logado_status:
        print("Usuário não encontrado.")
```

Lista de produtos



✿ O intuito desse programa é criar a lista de **produtos vendidos na loja**

✿ def mostrar categorias - mostra todas as categorias que existem

✿ def mostrar_produtos_da_categoria(categoria_escolhida):
Mostra todos os produtos que estão dentro da categoria selecionada

✿ A função para encerrar do programa é a main()

✿ O programa identifica se a categoria selecionada existe

```
#lista de produtos
produtos = [
    {"nome": "Perfume Floral", "categoria": "Perfumes", "preco": 129.90},
    {"nome": "Colônia Masculina", "categoria": "Perfumes", "preco": 99.50},
    {"nome": "Creme Hidratante", "categoria": "Cuidados com a Pele", "preco": 45.00},
    {"nome": "Sabonete Líquido", "categoria": "Higiene", "preco": 22.30},
    {"nome": "Loção Pós-Barba", "categoria": "Barbear", "preco": 38.90},
    {"nome": "Shampoo Reconstrutor", "categoria": "Cabelos", "preco": 29.90},
    {"nome": "Condicionador", "categoria": "Cabelos", "preco": 28.50},
    {"nome": "Óleo Corporal", "categoria": "Cuidados com a Pele", "preco": 58.00},
    {"nome": "Desodorante Roll-on", "categoria": "Higiene", "preco": 14.20},
    {"nome": "Máscara Capilar", "categoria": "Cabelos", "preco": 67.80}
]
print(produtos)

def mostrar_categorias():
    categorias = set()
    for produto in produtos:
        categorias.add(produto["categoria"])
    print("Categorias disponíveis:")
    for categoria in categorias:
        print(f"- {categoria}")

def mostrar_produtos_da_categoria(categoria_escolhida):
    encontrou = False
    print(f"\nProdutos da categoria '{categoria_escolhida}':")
    for produto in produtos:
        if produto["categoria"].lower() == categoria_escolhida.lower():
            print(f"- {produto['nome']} | Preço: R$ {produto['preco']:.2f}")
            encontrou = True
    return encontrou

def main():
    while True:
        mostrar_categorias()
        categoria = input("\nDigite o nome da categoria desejada (ou 'sair' para encerrar): ")
        if categoria.lower() == 'sair':
            print("Programa encerrado. Até logo!")
            break
        if not mostrar_produtos_da_categoria(categoria):
            print("Categoria inválida ou sem produtos. Tente novamente.")

main()
```

Carrinho de compras

Estrutura utilizada: Fila
(FIFO - First In, First Out)

Armazena os produtos adicionados pelo usuário na ordem em que foram escolhidos.

Primeiro a entrar é o primeiro a ser processado no checkout.

Mantém a ordem de compra dos perfumes.

Simples de implementar e entender.

Útil para sistemas que processam os pedidos sequencialmente.

```
#Carrinho de compras
from collections import deque

fila = []
cont = 0

while cont < 15:
    n_produto = input("Digite o produto que deseja adicionar ao carrinho ou se deseja sair: ")
    fila.append(n_produto)
    if n_produto != 'sair':
        print(f'Produto', n_produto, 'adicionado ao carrinho.')
    cont += 1
    if n_produto == 'sair':
        break
    print(fila)
print("1 - Realizar a compra")
print("2 - Continuar comprando")
compra = input("Digite a opção desejada: ")
if compra == '1':
    fila.clear()

else:
    print("Continuar comprando")
while cont < 15:
    n_produto = input("Digite o produto que deseja adicionar ao carrinho ou se deseja sair: ")
    fila.append(n_produto)
    if n_produto != 'sair':
        print(f'Produto', n_produto, 'adicionado ao carrinho.')
    cont += 1
    if n_produto == 'sair':
        break
    print(fila)

print(fila)
```

Formas de pagamento



Um conjunto de lista com pilha, permitindo controlar e reverter facilmente as últimas ações do usuário, como cancelar ou revisar o último pagamento feito. Essa estrutura garante flexibilidade, segurança e rastreamento de ações, sendo ideal para organizar dados temporários.

```
historico_pagamentos = []
```

```
historico_pagamentos.append(pagamento)
```

adiciona um pagamento

```
ultimo = historico_pagamentos.pop()
```

consulta pagamentos feitos





Obrigada!

