

Av. 60 pontos

F5 studios

E-COMMERCE

LOGIN E LOGOUT – VETOR

```
import numpy as np

v_usuario = np.array(["luis", "gaby", "maria"])
v_senhas = np.array(["1123", "1234", "12345"])

tentativas = 1

while tentativas <= 3:
    usuario_logado = input("Digite seu usuário: ")
    senha_logada = input("Digite sua senha: ")

    # Verifica se o usuário existe no vetor
    if usuario_logado in v_usuario:
        # Pega o índice do usuário
        indice = np.where(v_usuario == usuario_logado)[0][0]
        # Verifica se a senha no mesmo índice corresponde
        if v_senhas[indice] == senha_logada:
            print(f"Acesso permitido\nSeja bem-vindo(a), {usuario_logado}!")
            break
        else:
            print("Senha incorreta.")
    else:
        print("Usuário não encontrado.")

    tentativas += 1

if tentativas > 3:
    print("Número de tentativas excedido. Acesso bloqueado.")
```

```
Digite seu usuário: luis
Digite sua senha: 1123
Acesso permitido
Seja bem-vindo(a), luis!
```

LOGIN E LOGOUT

Bom, este formato de vetor foi escolhido devido a facilidade de manipulação do login , reduzindo assim o tempo do processo e aumentando a eficiência desta porção do código.

deste modo, organizando também uma visão geral de priorização de utilidade dos outros tipos de formato nas demais aplicações.

LISTA DE PRODUTOS – Lista

```
catalogo_diverso = [  
    {  
        "titulo": "A Origem",  
        "genero": "Ficção Científica/Suspense",  
        "classificacao": "14 anos"  
    },  
    {  
        "titulo": "Vingadores: Ultimato",  
        "genero": "Ação/Aventura",  
        "classificacao": "12 anos"  
    },  
    {  
        "titulo": "Forrest Gump",  
        "genero": "Drama/Comédia",  
        "classificacao": "12 anos"  
    },  
    {  
        "titulo": "Corra!",  
        "genero": "Suspense/Terror Psicológico",  
        "classificacao": "16 anos"  
    },  
    {  
        "titulo": "A Vida é Bela",  
        "genero": "Drama/Comédia",  
        "classificacao": "12 anos"  
    },  
    {  
        "titulo": "Matrix",  
        "genero": "Ficção Científica/Ação",  
        "classificacao": "14 anos"  
    },  
    {  
        "titulo": "Pantera Negra",  
        "genero": "Ação/Aventura",  
        "classificacao": "12 anos"  
    },  
    {  
        "titulo": "Divertida Mente",  
        "genero": "Animação/Família",  
        "classificacao": "Livre"  
    },  
    {  
        "titulo": "O Lobo de Wall Street",  
        "genero": "Drama/Biografia",  
        "classificacao": "18 anos"  
    },  
    {  
        "titulo": "O Fabuloso Destino de Amélie Poulain",  
        "genero": "Comédia/Romance",  
        "classificacao": "12 anos"  
    }  
]
```

```
# Exibindo a lista  
for item in catalogo_diverso:  
    print(f"Título: {item['titulo']}")  
    print(f"Gênero: {item['genero']}")  
    print(f"Classificação: {item['classificacao']}")  
    print("-" * 30)
```

A estrutura de lista foi utilizada por ser ideal para representar o eixo central do projeto: o catálogo de produtos, que neste caso são os filmes. Como o universo cinematográfico está sempre recebendo novos lançamentos, a lista permite a manipulação dinâmica dos dados, como adição, remoção e organização dos filmes de forma prática e eficiente.

CARRINHO DE COMPRAS – Matrix

```
catalogo = [
    ["A Origem", "Ficção Científica/Suspense", "14 anos"],
    ["Vingadores: Ultimato", "Ação/Aventura", "12 anos"],
    ["Forrest Gump", "Drama/Comédia", "12 anos"],
    ["Corra!", "Suspense/Terror Psicológico", "16 anos"],
    ["A Vida é Bela", "Drama/Comédia", "12 anos"],
    ["Matrix", "Ficção Científica/Ação", "14 anos"],
    ["Pantera Negra", "Ação/Aventura", "12 anos"],
    ["Divertida Mente", "Animação/Família", "Livres"],
    ["O Lobo de Wall Street", "Drama/Biografia", "18 anos"],
    ["Amélie Poulain", "Comédia/Romance", "12 anos"]
]

carrinho = []

print("Catálogo:")
for i in range(len(catalogo)):
    print(i, "-", catalogo[i][0])

# Você pode adicionar até 3 filmes no carrinho
while len(carrinho) < 3:
    print(f"Pode escolher mais {3 - len(carrinho)} filme(s).")
    n = int(input("Número do filme: "))
    carrinho.append(catalogo[n])
    print("Filme adicionado!")

print("Carrinho cheio com 3 filmes:")
for filme in carrinho:
    print(filme[0])
```

```
↵ Catálogo:
0 - A Origem
1 - Vingadores: Ultimato
2 - Forrest Gump
3 - Corra!
4 - A Vida é Bela
5 - Matrix
6 - Pantera Negra
7 - Divertida Mente
8 - O Lobo de Wall Street
9 - Amélie Poulain
Pode escolher mais 3 filme(s).
Número do filme: 1
Filme adicionado!
Pode escolher mais 2 filme(s).
Número do filme: 5
Filme adicionado!
Pode escolher mais 1 filme(s).
Número do filme: 6
Filme adicionado!
Carrinho cheio com 3 filmes:
Vingadores: Ultimato
Matrix
Pantera Negra
```

Utilizar a matriz facilita a organização e o gerenciamento dos dados. As linhas da matriz representam cada item do carrinho. Esse formato permite acessar e manipular os dados de forma estruturada, tornando a atualização dos itens mais eficiente. Além disso, a matriz ajuda a manter o código organizado e simplifica a implementação de funções como adicionar, remover ou alterar produtos no carrinho.

FORMA DE PAGAMENTO – Fila

```
main.py
1
2 fila = ['dinheiro', 'pix', 'cartão']
3 print("Formas de pagamento aceitas:", fila)
4
5 pagamento = input("Qual será a forma de pagamento? ").lower().strip()
6 forma_pagamento_carrinho = None
7
8 if pagamento == 'cartão':
9     tipo_cartao = input("Será débito ou crédito? ").lower().strip()
10    forma_pagamento_carrinho = f"cartão ({tipo_cartao})"
11    print("Forma de pagamento salva no carrinho:", forma_pagamento_carrinho)
12    print("Pagamento aceito! Obrigado pela preferência.")
13
14 elif pagamento in fila:
15     forma_pagamento_carrinho = pagamento
16     print("Forma de pagamento salva no carrinho:", forma_pagamento_carrinho)
17     print("Pagamento aceito! Obrigado pela preferência.")
18
19 else:
20     print("Forma de pagamento inválida! Tente novamente.")
21 |
```

Escolhemos a fila para trabalhar com o código de forma de pagamento porque ela garante que o processamento seja realizado na ordem correta, evitando perda ou modificação indevida dos dados. Além disso, a fila processa cada forma de pagamento de modo isolado, o que contribui para a redução de erros durante o processamento.

DADOS DO USUÁRIO LOGADO – Pilha

```
#Dados do usuário logado
pilha = []
print("Digite até no máximo três aparelhos em que se deseja adicionar a sua conta do F5 Studios (um por vez). Digite 'sair' para finalizar:")
for i in range (4):
    aparelho = input("Adicione um aparelho: ")
    if aparelho.upper() == "SAIR":
        break
    pilha.append(aparelho)
if len(pilha) == 4:
    pilha.pop()
    aparelhos_formatados = ", ".join(pilha)
    print("A quantidade de aparelhos cadastrados excedeu. O último aparelho foi removido. O(s) aparelho(s) que permanece(m) cadastrado(s) são:", aparelhos_formatados, ".")
else:
    aparelhos_formatados = ", ".join(pilha)
    print("Os aparelhos cadastrados com sucesso são:", aparelhos_formatados, ".")
```

⇒ Digite até no máximo três aparelhos em que se deseja adicionar a sua conta do F5 Studios (um por vez). Digite 'sair' para finalizar:
Adicione um aparelho: Meu celular
Adicione um aparelho: Meu computador
Adicione um aparelho: Minha TV
Adicione um aparelho: O celular da minha mãe
A quantidade de aparelhos cadastrados excedeu. O último aparelho foi removido. O(s) aparelho(s) que permanece(m) cadastrado(s) são: Meu celular, Meu computador, Minha TV .

A pilha foi utilizada para controle de aparelhos “cadastrados”, para não ter mais do que o permitido. Se mais do que três aparelhos forem detectados, o último será excluído. Isso para melhor organização e desenvolvimento do sistema.



OBRIGADA PELA ATENÇÃO!

Alunos:

**GABRIELA W.
GABRIELLE G.
ISADORA
JOÃO VICTOR
MARIA LAURA**