

FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA – FIAP

GRUPO NOMAD:

JOÃO FELIPE BALLETT – RM 550979

GABRIEL RODRIGUES DUARTE – RM 550185

PEDRO HENRIQUE DE SOUZA ALMEIDA – RM 98609

VITOR EDUARDO MOURA – RM 98450

CHALLENGE SPRINT – COMPUTATIONAL THINKING OF
ENGINEERING – 2º SEMESTRE

SÃO PAULO

2023

Objetivo do código:

Simular uma carteira virtual que permite consultar o saldo, adicionar saldo, realizar pagamentos e exibir o histórico de transações.

O código faz uso das bibliotecas json e os do Python.

Funções:

carregar carteira() e salvar carteira(): Estas funções lidam com a leitura e escrita dos dados da carteira em um arquivo JSON chamado carteira.json. Se o arquivo não existe, a função carregar_carteira() cria um novo arquivo com um saldo inicial de R\$100 e um histórico vazio. A função salvar_carteira() atualiza o arquivo com os dados da carteira e histórico a cada transação.

```
#Função arquivo .json
def carregar_carteira():
    if os.path.isfile('carteira.json'):
        with open('carteira.json', 'r') as file:
            return json.load(file)
    else:
        dados_iniciais = {'saldo': 100.00, 'historico': []}
        with open('carteira.json', 'w') as file:
            json.dump(dados_iniciais, file)
        return dados_iniciais

#Função salvar no arquivo .json
def salvar_carteira(carteira):
    with open('carteira.json', 'w') as file:
        json.dump(carteira, file, indent=4)
```

add saldo(carteira, valor): Adiciona um valor específico ao saldo da carteira. Se o valor é maior que zero, atualiza o saldo, registra a operação no histórico e salva esses dados no arquivo JSON.

```
#Definição do Depósito
def add_saldo(carteira, valor):
    if valor > 0:
        carteira['saldo'] += valor
        carteira['historico'].append(("Adição de saldo", valor))
        salvar_carteira(carteira)
        print(f"\nSaldo atualizado para R$ {carteira['saldo']:.2f}")
    else:
        print("\nValor inválido. O valor deve ser maior que zero.")
```

exibe historico formatado(transacoes): Exibe o histórico de transações de forma formatada, mostrando a descrição da transação e o valor.

```
#Definição do Depósito
def add_saldo(carteira, valor):
    if valor > 0:
        carteira['saldo'] += valor
        carteira['historico'].append(("Adição de saldo", valor))
        salvar_carteira(carteira)
        print(f"\nSaldo atualizado para R$ {carteira['saldo']:.2f}")
    else:
        print("\nValor inválido. O valor deve ser maior que zero.")
```

O programa oferece um menu de opções ao usuário para consultar saldo, adicionar saldo, realizar pagamentos ou visualizar o histórico de transações.

```
#Programa Principal
cart_virt = carregar_carteira()

while True:
    menu = input("""\nBem-vindo a Carteira Virtual da Shell Box by Nomad

    Opções:

    1. Consultar Saldo
    2. Adicionar Saldo
    3. Realizar Pagamento
    4. Exibir Histórico de Transações

    0. Sair

    Sua Opção: """)
```

```
Bem-vindo a Carteira Virtual da Shell Box by Nomad

    Opções:





    1. Consultar Saldo
    2. Adicionar Saldo
    3. Realizar Pagamento
    4. Exibir Histórico de Transações

    0. Sair

    Sua Opção:
```

Cada opção do menu chama funções específicas para executar a operação desejada.

Ao realizar uma operação (adicionar saldo ou pagamento), o código atualiza a carteira, registra a transação no histórico e salva esses dados no arquivo carteira.json.

 2chal_2se_CTE_nomad.py		31/10/2023 20:23	Arquivo PY	4 KB
 carteira.json		31/10/2023 20:37	Arquivo JSON	1 KB

Após a execução de qualquer operação, o programa exibe mensagens para informar o usuário sobre o resultado da transação ou solicitações de entrada de dados, verificando se são válidos.

Armazenamento de dados:

O programa utiliza um arquivo JSON chamado carteira.json para armazenar os dados da carteira, que incluem o saldo e o histórico de transações. Cada vez que há uma modificação (adicionar saldo ou fazer um pagamento), os dados são atualizados no arquivo JSON para manter o registro das transações.