

Projeto - Banco QuemPoupaTem

PRIMEIRA PARTE

No início do projeto nós fizemos o que foi solicitado e estes foram os resultados:

```
PS C:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo> python -u "c:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo\Projeto\parte1.py"
1. Novo cliente
2. Apagar cliente
3. Listar clientes
4. Débito
5. Depósito
6. Extrato
7. Transferência entre contas
8. Invest
9. Sair
Escolha uma opção:
```

```
PS C:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo> python -u "c:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo\Projeto\parte1.py"
1. Novo cliente
2. Apagar cliente
3. Listar clientes
4. Débito
5. Depósito
6. Extrato
7. Transferência entre contas
8. Invest
9. Sair
Escolha uma opção: 1
Nome: Raphael
CPF: 57998244828
Tipo de conta: Comum
Valor inicial da conta: 2005
Senha: Projetobanco123
None
```

```
1. Novo cliente
2. Apagar cliente
3. Listar clientes
4. Débito
5. Depósito
6. Extrato
7. Transferência entre contas
8. Invest
9. Sair
Escolha uma opção: 2
Cliente que deseja excluir: raphael
None
```

```
PS C:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo> python -u "c:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo\Projeto\parte1.py"
1. Novo cliente
2. Apagar cliente
3. Listar clientes
4. Débito
5. Depósito
6. Extrato
7. Transferência entre contas
8. Invest
9. Sair
Escolha uma opção: 4
Digite o CPF da sua conta: 09471471027501824
Digite sua senha: projetobanco123
Digite o valor a ser debitado: 300
None
```

```

PS C:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo> python -u "c:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo\Projeto\parte1.py"
1. Novo cliente
2. Apagar cliente
3. Listar clientes
4. Débito
5. Depósito
6. Extrato
7. Transferência entre contas
8. invest
9. Sair
Escolha uma opção: 5
CPF que você deseja enviar: 130719842143
Senha: projetobacno123
Valor que irá depositar: 3000
None

```

```

def extrato():
    cpf_extrato = float(input('Digite o cpf para retirar o extrato: '))
    senhaextrato = input('Digite a senha: ')
    print(f'Nome: {nome}')
    print(f'CPF: {cpf_extrato}')
    print(f'Conta: {tipo_conta}')
    print(Transferencias)

```

Não esta funcionando, pois nao tem o banco de dados necessario para mostrar os dados do extrato do cliente

```

PS C:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo> python -u "c:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo\Projeto\parte1.py"
1. Novo cliente
2. Apagar cliente
3. Listar clientes
4. Débito
5. Depósito
6. Extrato
7. Transferência entre contas
8. invest
9. Sair
Escolha uma opção: 7
Digite o cpf para realizar a transferencia: 218391283013
Digite sua senha: projetobacno123
Digite o cpf do destinatário: 3902183912748
Valor da tranferência: 9000
None

```

```

PS C:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo> python -u "c:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo\Projeto\parte1.py"
1. Novo cliente
2. Apagar cliente
3. Listar clientes
4. Débito
5. Depósito
6. Extrato
7. Transferência entre contas
8. invest
9. Sair
Escolha uma opção: 8
Valor a investir: 20389123
Tempo de investimeto em meses: 3
None

```

```

PS C:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo> python -u "c:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo\Projeto\parte1.py"
1. Novo cliente
2. Apagar cliente
3. Listar clientes
4. Débito
5. Depósito
6. Extrato
7. Transferência entre contas
8. invest
9. Sair
Escolha uma opção: 9
Muito obrigado por ter escolhido a
PS C:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo> python -u "c:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo\Projeto\parte1.py"
1. Novo cliente
2. Apagar cliente
3. Listar clientes
4. Débito
5. Depósito
6. Extrato
7. Transferência entre contas
8. invest
9. Sair
Escolha uma opção: 9
Muito obrigado por ter escolhido nosso banco
PS C:\Users\rgara\OneDrive - FEI\Fundamentos de Algoritimo>

```

SEGUNDA PARTE

Na segunda parte do nosso projeto, nós finalizamos a função adicionar cliente adicionando-o em uma lista onde você pode encontrar todos os clientes do banco.

Código Novo cliente:

```
# Função que adiciona um cliente ao banco
def novo_cliente():
    nome = input("Nome: ")
    cpf = input("CPF: ")
    tipo_conta = input("Tipo de conta (Comum ou Plus?): ")
    valor_inicial_conta = input("Valor inicial da conta: ")
    senha_usuario = input("Senha: ")
    informacoes = nome + "," + cpf + "," + tipo_conta + "," + valor_inicial_conta + "," + senha_usuario
    clientes.append(informacoes)
```

FUNCIONAMENTO Novo cliente:

Podemos notar que ao longo da execução da função, as informações são adicionadas a lista de clientes.

```
1. Novo cliente
2. Apagar cliente
3. Listar clientes
4. Débito
5. Depósito
6. Extrato
7. Transferência entre contas
8. invest
9. Sair
Escolha uma opção: 1
Nome: raphael
CPF: 123213123123
Tipo de conta (Comum ou Plus?): Plus
Valor inicial da conta: 60000
Senha: eusoulindo
['raphael,123213123123,Plus,60000,eusoulindo']
1. Novo cliente
2. Apagar cliente
3. Listar clientes
4. Débito
5. Depósito
6. Extrato
7. Transferência entre contas
8. invest
9. Sair
Escolha uma opção: 1
Nome: Nathan
CPF: 654646456
Tipo de conta (Comum ou Plus?): Comum
Valor inicial da conta: 10
Senha: eusoufeio
['raphael,123213123123,Plus,60000,eusoulindo', 'Nathan,654646456,Comum,10,eusoufeio']
```

Também nós finalizamos a função apagar cliente, pesquisando o cpf do usuario desejado na lista e apagando todo o conteudo.

Código Apagar cliente:

```
def apagar_cliente():
    cpf_apagar = input("Digite o cpf do cliente que você deseja apagar: ")
    for cpf in clientes:
        if cpf_apagar in cpf:
            clientes.remove(cpf)
            print("Cliente excluído com sucesso!")
            break
    print(clientes)
```

Funcionamento Apagar cliente:

No início nós podemos observar o usuário Raphael antes de executar a opção, logo após executar-la, podemos notar que ela não está mais na lista de clientes.

```
['raphael,123213123123,Plus,60000,eusoulindo', 'Nathan,654646456,Comum,10,eusoufeio']
1. Novo cliente
2. Apagar cliente
3. Listar clientes
4. Débito
5. Depósito
6. Extrato
7. Transferência entre contas
8. Invest
9. Sair
Escolha uma opção: 2
Digite o cpf do cliente que você deseja apagar: 123213123123
Cliente excluído com sucesso!
['Nathan,654646456,Comum,10,eusoufeio']
```

TERCEIRA PARTE

Na terceira parte, nós reajustamos o todo o código feito até agora para armazenar as informações em dicionários e também nós adicionamos as funções “lista_cliente”, “debito”, “deposito” e Transferência entre contas

Código lista_clientes:

```
def lista_clientes():
    print('Lista de clientes:')
    for indice in range(len(clientes)):
        print(clientes[indice])
```

Funcionamento lista_clientes:

```
[1] Novo cliente
[2] Apagar cliente
[3] Listar clientes
[4] Débito
[5] Depósito
[6] Extrato
[7] Transferência entre contas
[8] Investimentos
[9] Sair
Escolha uma das opções acima: 3

Lista de clientes:
{'Nome': 'Raphael', 'CPF': '57998244828', 'Tipo de conta': 'Plus', 'Saldo': 1000.0, 'Senha': '1234'}
{'Nome': 'Nathan', 'CPF': '54170078896', 'Tipo de conta': 'Comum', 'Saldo': 1000.0, 'Senha': '1234'}
{'Nome': 'Denis', 'CPF': '25209896889', 'Tipo de conta': 'Comum', 'Saldo': 1000.0, 'Senha': '1234'}
```

Código debito:

```
def debito():
    cpf_debito = input("Digite o CPF da sua conta: ")
    senha_debito = input("Digite sua senha: ")
    valor_debito = float(input("Digite o valor a ser debitado: "))
    for indice in range(len(clientes)):
        if cpf_debito == clientes[indice]['CPF'] and senha_debito == clientes[indice]['Senha'] and clientes[indice]['Tipo de conta'] == 'Comum':
            clientes[indice]['Saldo'] -= valor_debito * 1.05
            if clientes[indice]['Saldo'] < -1000:
                clientes[indice]['Saldo'] += valor_debito * 1.05
                print("")
                print('Transação ultrapassou o limite de saldo negativo. Transação cancelada!')
            elif clientes[indice]['Saldo'] >= -1000:
                print("")
                print('Transação realizada com sucesso!')
                print(f'Valor cobrado com taxa: {valor_debito * 1.05:.2f}')
                print(f'Saldo atual: {clientes[indice]["Saldo"]:.2f}')
        elif cpf_debito == clientes[indice]['CPF'] and senha_debito == clientes[indice]['Senha'] and clientes[indice]['Tipo de conta'] == 'Plus':
            clientes[indice]['Saldo'] -= valor_debito * 1.03
            if clientes[indice]['Saldo'] < -5000:
                clientes[indice]['Saldo'] += valor_debito * 1.03
                print("")
                print('Transação ultrapassou o limite de saldo negativo. Transação cancelada!')
            elif clientes[indice]['Saldo'] >= -5000:
                print("")
                print('Transação realizada com sucesso!')
                print(f'Valor cobrado com taxa: {valor_debito * 1.03:.2f}')
                print(f'Saldo atual: {clientes[indice]["Saldo"]:.2f}')
    else:
        print("CPF ou/e senha inválidas, tente novamente.")
```

Funcionamento debito:

```
[1] Novo cliente
[2] Apagar cliente
[3] Listar clientes
[4] Débito
[5] Depósito
[6] Extrato
[7] Transferência entre contas
[8] Investimentos
[9] Sair
Escolha uma das opções acima: 4

Digite o CPF da sua conta: 57998244828
Digite sua senha: 1234
Digite o valor a ser debitado: 500

Transação realizada com sucesso!
Valor cobrado com taxa: 515.00
Saldo atual: 485.00
```

Código depósito:

```
def deposito():
    cpf_deposito = input("Digite o CPF da sua conta: ")
    senha_deposito = input('Digite sua senha: ')
    valor_deposito = float(input('Digite o valor a ser debitado: '))
    print("")
    for indice in range(len(clientes)):
        if cpf_deposito == clientes[indice]['CPF'] and senha_deposito == clientes[indice]['Senha']:
            clientes[indice]['Saldo'] += valor_deposito
            print(f'Depósito realizada com sucesso!\nSaldo: {clientes[indice]["Saldo"]:.2f}')
        else:
            print("CPF ou/e senha inválidas, tente novamente.")
```

Funcionamento deposito:

```
[1] Novo cliente
[2] Apagar cliente
[3] Listar clientes
[4] Débito
[5] Depósito
[6] Extrato
[7] Transferência entre contas
[8] Investimentos
[9] Sair
Escolha uma das opções acima: 5

Digite o CPF da sua conta: 57998244828
Digite sua senha: 1234
Digite o valor a ser debitado: 1000

Depósito realizada com sucesso!
Saldo: 1485.00
```

Código Transferência entre contas:

```
def tranf_contas():
    cpf_origem = input("Digite o CPF da sua conta: ")
    senha_origem = input('Digite sua senha: ')
    cpf_destino = input('Digite o cpf da conta que você deseja fazer o depósito: ')
    valor_destino = float(input('Digite o valor a ser depositado para o destinatário: '))
    print("")
    for indice in range(len(clientes)):
        if cpf_origem == clientes[indice]['CPF'] and senha_origem == clientes[indice]['Senha'] and clientes[indice]['Tipo de conta'] == 'Comum':
            for indice in range(len(clientes)):
                if cpf_destino == clientes[indice]['CPF']:
                    clientes[indice]['Saldo'] -= valor_destino
                    if clientes[indice]['Saldo'] < -1000:
                        clientes[indice]['Saldo'] += valor_destino
                        print('Transação ultrapassou o limite de saldo negativo. Transação cancelada!')
                    elif clientes[indice]['Saldo'] >= -1000:
                        print('Transação realizada com sucesso!')
                        print(f'Valor enviado para o {cpf_destino}: {valor_destino:.2f}')
            elif cpf_origem == clientes[indice]['CPF'] and senha_origem == clientes[indice]['Senha'] and clientes[indice]['Tipo de conta'] == 'Plus':
                for indice in range(len(clientes)):
                    if cpf_destino == clientes[indice]['CPF']:
                        clientes[indice]['Saldo'] -= valor_destino
                        if clientes[indice]['Saldo'] < -5000:
                            clientes[indice]['Saldo'] += valor_destino
                            print('Transação ultrapassou o limite de saldo negativo. Transação cancelada!')
                        elif clientes[indice]['Saldo'] >= -5000:
                            print('Transação realizada com sucesso!')
                            print(f'Valor enviado para o {cpf_destino}: {valor_destino:.2f}')
    else:
        print("Informações inválidas, verifique se as informações estão corretas e tente novamente.")
```

Funcionamento Transferência entre contas

```
[1] Novo cliente
[2] Apagar cliente
[3] Listar clientes
[4] Débito
[5] Depósito
[6] Extrato
[7] Transferência entre contas
[8] Investimentos
[9] Sair
Escolha uma das opções acima: 7

Digite o CPF da sua conta: 54170078896
Digite sua senha: 1234
Digite o cpf da conta que você deseja fazer o depósito: 25209896889
Digite o valor a ser depositado para o destinatário: 500

Transação realizada com sucesso!
Valor enviado para o 25209896889: 500.00
```

QUARTA PARTE

Nessa parte, criamos a função do extrato() e do invest().

Código extrato():

```
# Função que mostra o extrato da conta bancária do usuário
def extrato():
    cpf_extrato = input('Digite o cpf para retirar o extrato: ')
    senhaextrato = input('Digite a senha: ')
    print()
    for indice in range(len(clientes)): # Itera sobre cada dicionário
        if cpf_extrato == clientes[indice]['CPF'] and senhaextrato == clientes[indice]['Senha']: # Verifica os dados
            print(f'Nome: {clientes[indice]["Nome"]}')
            print(f'CPF: {clientes[indice]["CPF"]}')
            print(f'Conta: {clientes[indice]["Tipo de conta"]}')
            for index in range(len(clientes[indice]['Extrato'])): # Itera sobre cada elemento da lista que está na chave Extrato e imprime.
                print(clientes[indice]['Extrato'][index]) # Cada elemento da lista Extrato é uma string formatada
```

Funcionamento extrato():

```
[1] Novo cliente
[2] Apagar cliente
[3] Listar clientes
[4] Débito
[5] Depósito
[6] Extrato
[7] Transferência entre contas
[8] Investimentos
[9] Sair
Escolha uma das opções acima: 6

Digite o cpf para retirar o extrato: 54170078896
Digite a senha: bancoompoupatem

Nome: Nathan
CPF: 54170078896
Conta: Comum
Data: 13/05/2023 12:53:35 + 5000.0 Tarifa: 0.00 Saldo: 10000.00
```

Código invest():

```
def invest(): # Função que faz operações livres
    print()
    print('[1] Tesouro Direto')
    print('[2] Fundos Imobiliários')
    print('[3] Ações')
    print()
    cpf_investimento = input('Digite o seu CPF: ') # Linhas 147 a 151 recebe dados
    senha_investimento = input('Digite a sua senha: ')
    tipo_invest = input('Tipo de investimento que deseja fazer: ').capitalize()
    valor_investimento = float(input('Valor a investir: '))
    tempo_investimento = input('Data para retirar o valor(dia/mês/ano): ')
    tempo = tempo_investimento.split('/')
    data_investimento = date(int(tempo[2]), int(tempo[1]), int(tempo[0])) - date.today()
    for indice in range(len(clientes)): # Itera sobre os dicionários presentes na lista clientes
        if cpf_investimento == clientes[indice]['CPF'] and senha_investimento == clientes[indice]['Senha'] and tipo_invest == '1':
            clientes[indice]['Saldo'] -= valor_investimento # tira o valor do investimento
            investimento = valor_investimento * data_investimento.days/365 * 1.08
            clientes[indice]['Investimentos'].append(f'{investimento:.2f}') # Coloca o valor do investimento na chave Investimentos
            break
        elif cpf_investimento == clientes[indice]['CPF'] and senha_investimento == clientes[indice]['Senha'] and tipo_invest == '2':
            clientes[indice]['Saldo'] -= valor_investimento
            investimento = valor_investimento * data_investimento.days/365 * 1.08
            clientes[indice]['Investimentos'].append(f'{investimento:.2f}')
            break
        elif cpf_investimento == clientes[indice]['CPF'] and senha_investimento == clientes[indice]['Senha'] and tipo_invest == '3':
            clientes[indice]['Saldo'] -= valor_investimento
            investimento = valor_investimento * data_investimento.days/365 * 1.08
            clientes[indice]['Investimentos'].append(f'{investimento:.2f}')
            break
    else:
        print('Dados incorretos. Tente novamente mais tarde.')
```

Funcionamento invest():


```

[1] Novo cliente
[2] Apagar cliente
[3] Listar clientes
[4] Débito
[5] Depósito
[6] Extrato
[7] Transferência entre contas
[8] Investimentos
[9] Sair
Escolha uma das opções acima: 8

[1] Tesouro Direto
[2] Fundos Imobiliários
[3] Ações

Digite o seu CPF: 1
Digite a sua senha: 12
Tipo de investimento que deseja fazer: 1
Valor a investir: 1111
Data para retirar o valor(dia/mês/ano): 10/10/2030

```

```

[1] Novo cliente
[2] Apagar cliente
[3] Listar clientes
[4] Débito
[5] Depósito
[6] Extrato
[7] Transferência entre contas
[8] Investimentos
[9] Sair
Escolha uma das opções acima: 3

Lista de clientes:
{'Nome': 'n', 'CPF': '1', 'Tipo de conta': 'C', 'Saldo': 10000.0, 'Senha': '12', 'Extrato': [], 'Investimentos': ['8898.84']}
{'Nome': 'n', 'CPF': '2', 'Tipo de conta': 'P', 'Saldo': 22222.0, 'Senha': '32', 'Extrato': [], 'Investimentos': []}

```

QUINTA PARTE

Para finalizar o nosso projeto, nós incrementamos a função de adicionar todas as informações inseridas pelo usuário em um arquivo chamado clientes. Para realizar este processo nós incrementamos duas funções. A primeira é a “**ler_arquivo()**” que lê o arquivo, pega todas as informações e coloca dentro de uma variável que armazena uma lista que é utilizada para fazer todos os processos do porta e a segunda função que é a “**substituir_informacoes_arquivos()**” que pega todas as informações que estão na variável criada pela primeira função, então, depois de pegar a lista, ela substitui as informações que estavam no arquivo pelas informações que estão na variável.

Código da função “ler_arquivo()”

```

# Funções para alterar tudo dentro do arquivo
def ler_arquivo(nome_arquivo):
    lista_informacoes = []
    with open(nome_arquivo, 'r') as arquivo: # Abre o arquivo no modo leitura na variável arquivo
        for linha in arquivo: # Laço que itera sobre cada linha do arquivo
            dicionario = eval(linha.strip()) # Cada linha do arquivo é transformada em dicionário
            lista_informacoes.append(dicionario) # Adiciona cada dicionário na lista lista_informacoes
    return lista_informacoes

```

Código da função “substituir_informacoes_arquivos()”

```
def substituir_informacoes_arquivo(nome_arquivo, lista_informacoes):  
    with open(nome_arquivo, 'w') as arquivo: # Abre o arquivo no modo  
        for informacao in lista_informacoes: # Para cada item(dicionário)  
            arquivo.write(str(informacao) + '\n')
```

CONCLUSÃO

Ao longo do projeto, tivemos algumas dificuldades para definir qual seria o método para armazenar as informações dos clientes em uma lista pois a lista não armazenava números, então, nós armazenamos dicionários em uma lista. Nós também para finalizar o projeto tivemos dificuldade para colocar todas as informações dentro do arquivo, mas depois de muito tempo pensando e testando diversos códigos chegamos ao resultado esperado por nós.

Por fim, no decorrer do nosso primeiro semestre, nós praticamos e desenvolvemos muito a nossa lógica de programação e nossos conhecimentos de python nas aulas com o Charles, da Gabriela e ao longo do desenvolvimento deste projeto proposto por ambos os professores.