

CENTRO UNIVERSITÁRIO FEI

**Artur Chaves Paiva
Giovanni Antonio Moreira**

Banco QuemPoupaTem PJ

São Bernardo do Campo

2023

SUMÁRIO

1. INTRODUÇÃO.....	3
2. INÍCIO DO PROGRAMA.....	3
2.1. Login	3
2.2. Menu inicial	4
2.3. Armazenamento	4
3. OPERAÇÕES SIMPLES.....	6
3.1. Novo cliente	6
3.2. Apagar cliente	7
3.3. Listagem de clientes.....	7
4. OPERAÇÕES BANCÁRIAS	8
4.1. Operação de débito	8
4.2. Operação de depósito	8
4.3. Extrato.....	9
4.4. Transferência entre contas	9
5. OPERAÇÃO DE GERENCIAMENTO DE FUNCIONÁRIOS	10
5.1. Menu da operação	10
5.2. Cadastrar funcionários	10
5.3. Apagar funcionários.....	11
5.4. Listar funcionários	11
5.5. Alterar salários	12
5.6. Pagamento dos funcionários	12
6. FIM DO PROGRAMA	13

1. INTRODUÇÃO

Documentação do projeto: “QuemPoupa tem PJ”

Neste documento é possível visualizar todas as funções que gerenciam nosso banco de uma maneira sucinta e prática.

2. INÍCIO DO PROGRAMA

2.1. Login

Na tela de login o usuário tem a opção de fazer login no banco, considerando que a conta do usuário já tenha sido cadastrada e caso contrário o usuário pode realizar o cadastro de sua conta.

```
def login():  
    # função para login ou registro de cliente  
    estilo_azul = Style.RESET_ALL + Fore.CYAN + Style.BRIGHT  
    estilo_branco = Style.RESET_ALL + Fore.LIGHTWHITE_EX  
  
    while True:  
        login = input(estilo_azul + """)  
  
        Bem-vindo ao Banco QuemPoupaTem PJ  
  
        1. Login  
        2. Registrar  
  
        Escolha a operação desejada:  
  
        """)
```

Após o usuário se cadastrar seguimos para a tela do menu inicial;

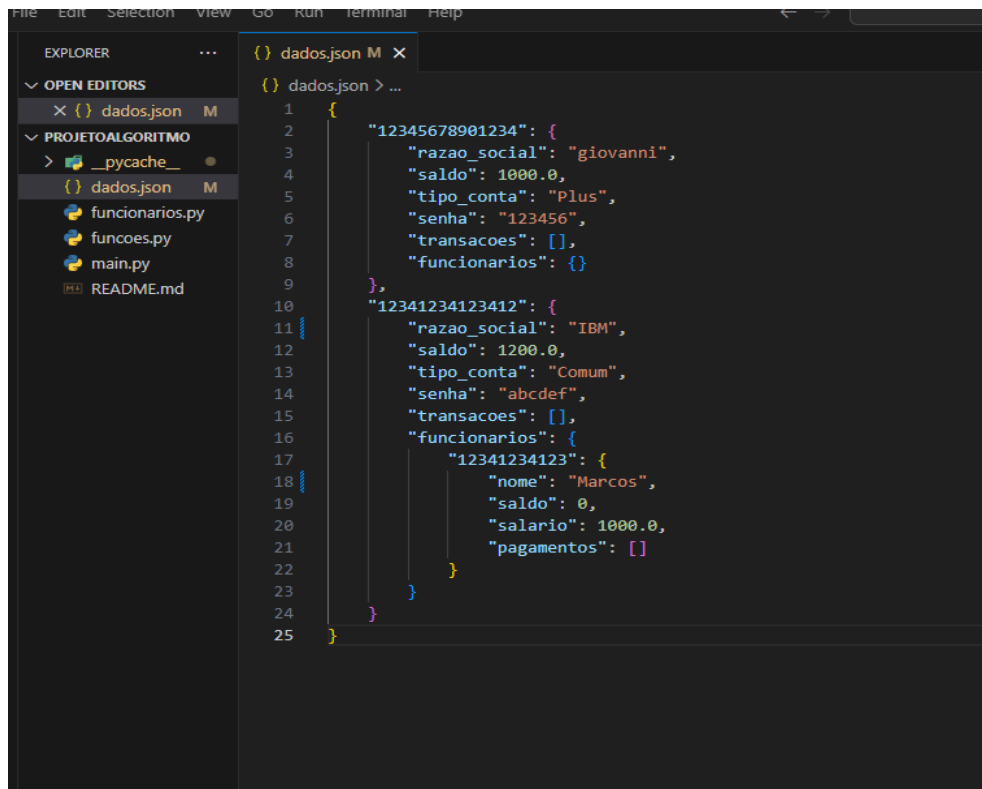
2.2. Menu inicial

No menu inicial os nossos usuários podem ter acesso as diversas funcionalidades do nosso banco, onde ao escolher o tipo de operação só é preciso digitar o número referente a mesma.

```
2 while(True):
3     operacao = input( estilo_azul + ""
4
5
6     Escolha a operação desejada:
7
8
9     1. Novo cliente
10    2. Apaga cliente
11    3. Listar clientes
12    4. Débito
13    5. Depósito
14    6. Extrato
15    7. Transfêrencia entre contas
16    8. Gerenciar Funcionários
17    9. Sair
18
19
20    "" + estilo_branco)
```

2.3. Armazenamento

Durante a formulação da estrutura de dados do nosso banco escolhemos utilizar do JSON (JavaScript Object Notation), manipular dados via JSON torna os dados dinâmicos e de fácil manipulação entre códigos devido sua estrutura de dados hierárquica que permite representar informações muito grandes com facilidade.



```
1 {
2   "12345678901234": {
3     "razao_social": "giovanni",
4     "saldo": 1000.0,
5     "tipo_conta": "Plus",
6     "senha": "123456",
7     "transacoes": [],
8     "funcionarios": {}
9   },
10  "12341234123412": {
11    "razao_social": "IBM",
12    "saldo": 1200.0,
13    "tipo_conta": "Comum",
14    "senha": "abcdef",
15    "transacoes": [],
16    "funcionarios": {
17      "12341234123": {
18        "nome": "Marcos",
19        "saldo": 0,
20        "salario": 1000.0,
21        "pagamentos": []
22      }
23    }
24  }
25 }
```

Durante os próximos tópicos as entradas de dados vão ser explicadas, porém é muito importante entender como são estruturados os dados para futuras alterações e desenvolvimentos de funções novas, visando a progressão do código em futuras versões.

Mais sobre JSON em:

<https://tinyurl.com/2s4fv392>

3. OPERAÇÕES SIMPLES

Nossas operações foram construídas num arquivo separado para máxima a otimização de código evitando mal desempenho e/ou repetição excessiva de código, claro que em futuros patches o nosso banco pode se adaptar mais ainda conforme a demanda de nossos clientes.

3.1. Novo cliente

Como função principal, cadastrar clientes se torna algo muito fácil, o programa apenas solicita os dados do usuário via terminal

Após o requerimento dos dados nosso programa é capaz de armazenar esses dados em um arquivo JSON (que foi citado no tópico “Armazenamento”), este arquivo possibilita uma otimização na manipulação de dados visando o melhor desempenho e flexibilidade do programa.

```
128     dados[cnpj] = {
129         "razao_social": razao_social,
130         "saldo" : saldo,
131         "tipo_conta" : tipo_conta,
132         "senha" : senha,
133         "transacoes": [],
134         "funcionarios": {}
135     } # modelo do dicionário que é salvo no arquivo .JSON
136
137     escrever(dados) # escreve este dicionário no arquivo .JSON
```

Dados se refere ao nosso arquivo JSON que armazena o CNPJ como identificador principal da conta e com os dados da conta.

3.2. Apagar cliente

Para a nossa operação apagar cliente, o usuário necessita apenas colocar o seu CNPJ, e dessa maneira o programa irá procurar dentro dos seus dados, e dessa maneira, deletar o usuário de maneira fácil e simples.

```
if str(cnpj) in dados:  
    del dados[str(cnpj)]  
    escrever(dados)
```

3.3. Listagem de clientes

Nossa listagem de clientes apenas mostra os usuários e seus dados por meio das informações escritas no arquivo JSON

Clientes:

=====

Razão social: giovanni
CNPJ: 12345678901234
Saldo: 794.0
Tipo da conta: Plus
Senha: 123456

=====

Razão social: sim
CNPJ: 12341234123412
Saldo: 1200.0
Tipo da conta: Comum
Senha: abcdef

4. OPERAÇÕES BANCÁRIAS

As próximas operações são aquelas que envolvem a movimentação do dinheiro e seu histórico, e que em nosso programa temos 4 funções: Débito, Depósito, Extrato e a Transferência entre contas do banco.

4.1. Operação de Débito

Para esta operação, o programa irá solicitar apenas três dados: o CNPJ do cliente com a sua senha e o valor que ele será debitado da conta, como o exemplo abaixo:

```
Operação escolhida: 4. Débito
  Digite o seu CNPJ: 12345678901234
  Digite sua senha: 123456
  Digite o valor que será debitado: 100

Valor debitado com sucesso, o seu saldo atual é de: R$ 585.0
```

Dessa maneira, será debitado do saldo do cliente no mesmo instante, e atualizado em todo o sistema.

4.2. Operação de Depósito

Da mesma maneira da operação de débito, a operação de depósito solicitará ao cliente o seu CNPJ e o valor que será depositado, como no exemplo abaixo:

```
Operação escolhida: 4. Débito
  Digite o seu CNPJ: 12345678901234
  Digite o valor que será debitado: 100

Valor debitado com sucesso, o seu saldo atual é de: R$ 691.0
```

Dessa forma, o saldo será atualizado em todo o programa, adicionando o valor desejado pelo cliente.

4.3. Extrato

O extrato, por sua vez, mostrará todas as transações feitas pelo usuário, com informações cruciais, como o tipo da operação, o valor, a data e o seu saldo atual, para acessá-lo o programa irá pedir o CNPJ e a senha do cliente para poder prosseguir, como mostrado abaixo:

```
Operação escolhida: 6. Extrato
Digite o seu CNPJ: 12345678901234
Digite sua senha: 123456

Tipo: Debito | Valor: R$ 206.0 | Tarifa: R$ 6.0 | Saldo atual: R$ 794.0 | Data: 13/11/2023 22:54:20

Tipo: Debito | Valor: R$ 103.0 | Tarifa: R$ 3.0 | Saldo atual: R$ 691.0 | Data: 16/11/2023 00:41:19

Tipo: Deposito | Valor: R$ 100.0 | Saldo atual: R$ 791.0 | Data: 16/11/2023 00:43:38

Tipo: Debito | Valor: R$ 103.0 | Tarifa: R$ 3.0 | Saldo atual: R$ 688.0 | Data: 16/11/2023 00:52:18

Tipo: Debito | Valor: R$ 103.0 | Tarifa: R$ 3.0 | Saldo atual: R$ 585.0 | Data: 16/11/2023 00:55:22
```

Com essa operação, o cliente pode checar as informações e a veracidade de suas ações, de maneira transparente e eficiente.

4.4. Transferência entre contas

Essa operação será responsável por transferir o dinheiro de um cliente para outro, para poder seguir com a função, será solicitado ao usuário, o CNPJ e a sua senha, o valor da transferência, e o CNPJ do destinatário, como mostrado abaixo:

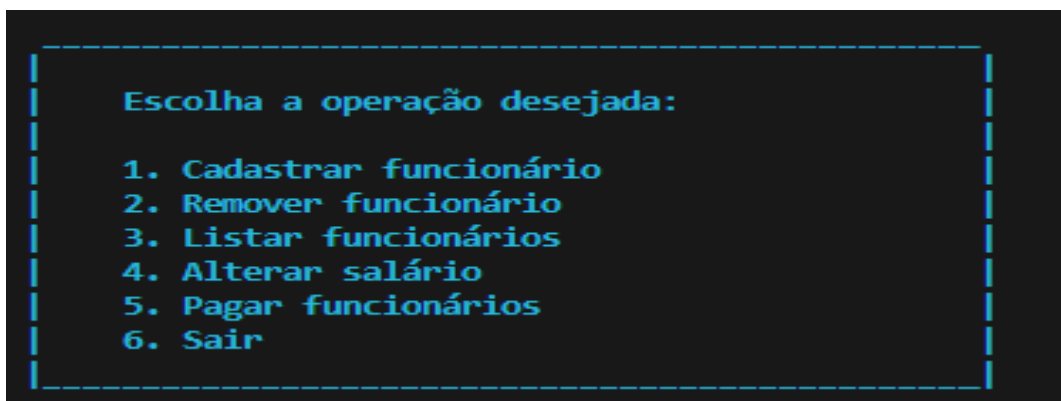
```
Operação escolhida: 7. Transfêrencia entre contas
Digite o seu CNPJ: 12345678901234
Digite sua senha: 123456
Digite o cnpj do destinatário: 12341234123412
Digite o valor da transfêrencia: 200
```

5. OPERAÇÃO DE GERENCIAMENTO DE FUNCIONÁRIOS

Essa operação é a responsável por gerenciar os funcionários de sua empresa, e conta com seu próprio menu, e operações como o: cadastro de funcionário, remoção de funcionários, alteração nos valores do salário e o pagamento de todos os subordinados.

5.1. Menu da operação

O menu de gerenciamento de funcionários funciona igual os outros já mostrados, rodando em loop e permitindo o cliente usar números para se locomover dentro do sistema.



5.2. Cadastrar funcionários

Para cadastrar funcionários, os valores dos CPFs dos funcionários são responsáveis para a mudança de seus dados, e esse é o layout e os valores necessários para o cadastro:

```
dados[cnpj]["funcionarios"][cpf] = {  
    "nome": nome,  
    "saldo" : saldo,  
    "salario": salario,  
    "pagamentos": [],  
}  
# esse é o modelo do dicionario que consta no arquivo .json para o cadastro dos funcionários
```

5.3. Apagar funcionários

Para a remoção dos usuários, o programa pede o CNPJ, o CPF do funcionário desejado, e dessa forma, o programa deleta o funcionário de seu banco de dados, como mostra abaixo:

```
Operação escolhida 2. Remover funcionário
```

```
CNPJ: 12345678901234
```

```
CPF: 12345678900
```

```
O funcionário portador do CPF: 12345678900 foi deletado com sucesso
```

5.4. Listar funcionários

Essa operação serve para listar todos os funcionários, junto com seus dados informados no cadastro, segue exemplo:

```
Operação escolhida 3. Listar funcionários
```

```
Informe o CNPJ: 12345678901234
```

```
=====
```

```
Funcionários:
```

```
=====
```

```
Nome: Isabela
```

```
CPF: 12345678911
```

```
Saldo: 0
```

```
Salário: 1500.0
```

5.5. Alterar salários

Para alterar o salário de algum funcionário, essa função pergunta o CNPJ do cliente, o CPF do funcionário desejado junto com o novo valor de salário, como no exemplo:

```

4

Operação escolhida 4. Alterar salário
Informe o CNPJ: 12345678901234
=====

Funcionários:

=====

Nome: Isabela
CPF: 12345678911
Saldo: 0
Salário: 1500.0

Informe o cpf do colaborador para alterar seu salário: 12345678911
Informe o novo salário do colaborador: 2000
Salário alterado com sucesso

```

5.6. Pagar salários

Essa operação é responsável por pagar os salários descritos nas informações de cada funcionário, essa função abre um pequeno menu:

```

Informe a operação:

1. Pagar funcionarios
2. Extrato dos pagamentos
3. Sair

```

Ao teclar “1”, os salários são automaticamente pagos, o extrato dos pagamentos estará na posição 2, e vêm dessa maneira:

```

Operação escolhida 2. Extrato dos pagamentos
Extrato de pagamentos para o Isabela - 12345678911:

Tipo: Recebimento | Valor: R$ 2000.0 | Saldo atual: R$ 2000.0 | Data: 16/11/2023 01:19:31
=====

```

6. FIM DO PROGRAMA

Com todas essas funcionalidades e funções, chega-se ao fim desse projeto, agradecemos a atenção e assim como no programa: 9. Sair