

Python para Dados

Professor: Luiz Eduardo Lima da Silva



Kaike Torres da silva

03.11.2024

Análise e Desenvolvimento de Sistemas (ADS)

INTRODUÇÃO

Este projeto em Python trabalha com perfis de usuários simulados para uma rede chamada "INFwebNET". Ele cria e organiza dados dos usuários, filtrando informações incompletas e armazenando apenas perfis válidos. A partir de uma base de dados inicial, realiza operações de manipulação, análise e atualização de conexões entre usuários. Funções permitem verificar popularidade, encontrar amigos em comum e atualizar dados, com o progresso salvo em um arquivo atualizado ao final.

RESULTADOS

1 - Aquecendo os motores ★

- Crie uma lista chamada 'usuarios' que contenha ao menos 5 listas. Cada lista interna deve representar um usuário do INFwebNET com as seguintes informações: nome (string), idade (inteiro), cidade (string) e estado (string).

```
1 - Aquecendo os motores ★
Crie uma lista chamada 'usuarios' que contenha ao menos 5 listas.
    Cada lista interna deve representar um usuário do INFwebNET com as seguintes
    informações: nome (string), idade (inteiro), cidade (string) e estado (string).
Usuarios: ['Maria da Silva', 45, 'São Paulo', 'SP']
Usuarios: ['José Santos', 52, 'São Paulo', 'SP']
Usuarios: ['Ana Oliveira', 28, 'São Paulo', 'SP']
Usuarios: ['João Pereira', 33, 'São Paulo', 'SP']
Usuarios: ['Carlos Sousa', 40, 'São Paulo', 'SP']
```

2 - Perfil ★★

- Escreva um programa que leia os dados da lista 'usuarios' criada no exercício anterior e crie um dicionário para cada usuário. Cada dicionário deve ter as chaves "nome" e "idade" com os respectivos valores, e a chave "localização" contendo uma tupla (cidade, estado). Armazene esses dicionários em uma nova lista chamada perfis.

```
2 - Perfil ★★
Escreva um programa que leia os dados da lista 'usuarios' criada no exercício
anterior e crie um dicionário para cada usuário.
    Cada dicionário deve ter as chaves "nome" e "idade" com os respectivos valores, e a
    chave "localização" contendo uma tupla (cidade, estado).
    Armazene esses dicionários em uma nova lista chamada perfis.
Perfil: {'nome': 'Maria da Silva', 'idade': 45, 'localizacao': ('São Paulo', 'SP')}
Perfil: {'nome': 'José Santos', 'idade': 52, 'localizacao': ('São Paulo', 'SP')}
Perfil: {'nome': 'Ana Oliveira', 'idade': 28, 'localizacao': ('São Paulo', 'SP')}
Perfil: {'nome': 'João Pereira', 'idade': 33, 'localizacao': ('São Paulo', 'SP')}
Perfil: {'nome': 'Carlos Sousa', 'idade': 40, 'localizacao': ('São Paulo', 'SP')}
```

3 - Comparando Estruturas ★★★★★

- Explique, em poucas palavras, as principais diferenças entre uma lista, um dicionário e uma tupla em Python. Dê exemplos de como cada estrutura pode ser usada no contexto da análise de dados do INFwebNET.

```
3 - Comparando Estruturas ★★★★★
Explique, em poucas palavras, as principais diferenças entre uma lista, um dicionário
e uma tupla em Python. Dê exemplos de como cada estrutura pode ser usada no contexto da
análise de dados do INFwebNET.
Lista: [['Maria', 45, 'São Paulo', 'SP'], ['José', 52, 'São Paulo', 'SP']]
Dicionário: {'nome': 'Maria', 'idade': 45, 'localização': ('São Paulo', 'SP')}
Tupla: ('São Paulo', 'SP')
```

4 - Limpando o terreno ★★★★★

- Alguns usuários do INFwebNET forneceram informações incompletas. Remova da lista perfis todos os perfis que não possuem as informações de "nome" ou "cidade". Mantenha a lista perfis original intacta, criando uma nova lista chamada perfis_validos para armazenar os perfis válidos.

```
4 - Limpando o terreno ★★★★★
Alguns usuários do INFwebNET forneceram informações incompletas.
Remova da lista perfis todos os perfis que não possuem as informações
de "nome" ou "cidade". Mantenha a lista perfis original intacta, criando
uma nova lista chamada perfis_validos para armazenar os perfis válidos.
Perfis Validos: {'nome': 'Maria da Silva', 'idade': 45, 'localizacao': ('São Paulo', 'SP')}
Perfis Validos: {'nome': 'José Santos', 'idade': 52, 'localizacao': ('São Paulo', 'SP')}
Perfis Validos: {'nome': 'Ana Oliveira', 'idade': 28, 'localizacao': ('São Paulo', 'SP')}
Perfis Validos: {'nome': 'João Pereira', 'idade': 33, 'localizacao': ('São Paulo', 'SP')}
Perfis Validos: {'nome': 'Carlos Sousa', 'idade': 40, 'localizacao': ('São Paulo', 'SP')}
```

5 - Carregando dados ★★★★★

- Crie uma implementação que leia os dados presentes no arquivo "base_inicial.txt" e os armazene na lista perfis_validos, criando novas palavras-chave para os dados adicionais encontrados. (O arquivo está disponível no repositório.)

```

5 - Carregando dados ★★
Crie uma implementação que leia os dados presentes no arquivo "base_inicial.txt"
e os armazene na lista perfis_validos, criando novas palavras-chave para os dados
adicionais encontrados. (O arquivo está disponível no repositório.)
Perfis validos atualizados:
Perfil: Perfil_Valido(nome='Maria da Silva', idade=45, localizacao=('São Paulo', 'SP'), amigos=[])
Perfil: Perfil_Valido(nome='José Santos', idade=52, localizacao=('São Paulo', 'SP'), amigos=[])
Perfil: Perfil_Valido(nome='Ana Oliveira', idade=28, localizacao=('São Paulo', 'SP'), amigos=[])
Perfil: Perfil_Valido(nome='João Pereira', idade=33, localizacao=('São Paulo', 'SP'), amigos=[])
Perfil: Perfil_Valido(nome='Carlos Sousa', idade=40, localizacao=('São Paulo', 'SP'), amigos=[])
Perfil: Perfil_Valido(nome='João', idade=25, localizacao='Rio de Janeiro, RJ', amigos=['Ana', 'Carlos', 'Marcos'])
Perfil: Perfil_Valido(nome='Ana', idade=22, localizacao='São Paulo, SP', amigos=['João', 'Fernanda'])
Perfil: Perfil_Valido(nome='Carlos', idade=30, localizacao='Belo Horizonte, MG', amigos=['João', 'Marcos', 'Caio'])
Perfil: Perfil_Valido(nome='Fernanda', idade=28, localizacao='Curitiba, PR', amigos=['Ana'])
Perfil: Perfil_Valido(nome='Marcos', idade=27, localizacao='Porto Alegre, RS', amigos=['João', 'Carlos'])
Perfil: Perfil_Valido(nome='Clara', idade=26, localizacao='Brasília, DF', amigos=['Fernanda', 'Livia'])
Perfil: Perfil_Valido(nome='Livia', idade=24, localizacao='São Paulo, SP', amigos=['Clara', 'Lucas', 'Marcos'])
Perfil: Perfil_Valido(nome='Lucas', idade=31, localizacao='Salvador, BA', amigos=['Livia', 'Fernando'])
Perfil: Perfil_Valido(nome='Fernando', idade=29, localizacao='Fortaleza, CE', amigos=['Lucas', 'Aline'])
Perfil: Perfil_Valido(nome='Aline', idade=23, localizacao='Florianópolis, SC', amigos=['Fernando', 'Caio', 'Clara'])
Perfil: Perfil_Valido(nome='Caio', idade=27, localizacao='Curitiba, PR', amigos=['Aline', 'Clara'])
Perfil: Perfil_Valido(nome='Sofia', idade=21, localizacao='Campinas, SP', amigos=['Rafaela', 'Fernanda'])
Perfil: Perfil_Valido(nome='Rafaela', idade=22, localizacao='Recife, PE', amigos=['Sofia', 'Clara'])
Perfil: Perfil_Valido(nome='Mateus', idade=32, localizacao='Rio de Janeiro, RJ', amigos=['Caio', 'Fernando', 'João'])
Perfil: Perfil_Valido(nome='Tiago', idade=28, localizacao='Brasília, DF', amigos=['Mateus', 'Livia', 'Aline'])
Perfil: Perfil_Valido(nome='Paulo', idade=33, localizacao='', amigos=['Tiago', 'Fernanda', 'Lucas'])
Perfil: Perfil_Valido(nome='Juliana', idade=25, localizacao='São Paulo, SP', amigos=['Rafaela', 'Sofia', 'Marcos'])
Perfil: Perfil_Valido(nome='Pedro', idade=26, localizacao='Belo Horizonte, MG', amigos=['João', 'Aline'])
Perfil: Perfil_Valido(nome='Amanda', idade=24, localizacao='Porto Alegre, RS', amigos=['Pedro', 'Caio', 'Livia'])
Perfil: Perfil_Valido(nome='Bruno', idade=30, localizacao='Cuiabá, MT', amigos=['Ana', 'Marcos', 'Mateus'])
Perfil: Perfil_Valido(nome='Carla', idade=29, localizacao='Salvador, BA', amigos=['Fernando', 'Tiago', 'Lucas'])
Perfil: Perfil_Valido(nome='Gabriela', idade=27, localizacao='Fortaleza, CE', amigos=['Carla', 'Aline', 'Rafaela'])
Perfil: Perfil_Valido(nome='Thiago', idade=31, localizacao='São Paulo, SP', amigos=['João', 'Amanda', 'Lucas', 'Clara'])
Perfil: Perfil_Valido(nome='Renata', idade=23, localizacao='Curitiba, PR', amigos=['Clara', 'Fernando', 'Sofia'])
Perfil: Perfil_Valido(nome='Leonardo', idade=29, localizacao='Campinas, SP', amigos=['Renata', 'Mateus', 'Caio'])
Perfil: Perfil_Valido(nome='Camila', idade=28, localizacao='Niterói, RJ', amigos=['Juliana', 'Bruno', 'Lucas', 'Caio'])
Perfil: Perfil_Valido(nome='Felipe', idade=27, localizacao='Brasília, DF', amigos=['Renata', 'Carla'])
Perfil: Perfil_Valido(nome='Mariana', idade=30, localizacao='Porto Alegre, RS', amigos=['Felipe', 'Livia', 'Fernando'])
Perfil: Perfil_Valido(nome='Daniel', idade=25, localizacao='Rio de Janeiro, RJ', amigos=['Camila', 'Paulo', 'Amanda'])

```

6 - Concatenando dados ★

- Com os dados carregados no exercício anterior, adicione os usuários dos exercícios 1 e 2, definindo um padrão para lidar com os dados ausentes e salve estas informações em um arquivo "rede_INFNET.txt".

```

6 - Concatenando dados ★
Com os dados carregados no exercício anterior, adicione os usuários dos exercícios 1 e 2, definindo um padrão para
lidar com os dados ausentes e salve estas informações em um arquivo "rede_INFNET.txt".

```

7 - Concatenando dados ★

- Com o dicionário criado no exercício anterior, adicione um novo amigo ao set de amigos de um usuário específico.

```

7 - Concatenando dados ★
Com o dicionário criado no exercício anterior, adicione um novo amigo ao set de amigos de um usuário específico.
Add ao usuario andre um amigo chamado kaike
Nome não encontrado.
Add ao usuario Maria um amigo chamado kaike
Nome não encontrado.

```

8 - Verificando Conexões ★★

- Crie um programa que permita verificar se um determinado usuário foi adicionado

como amigo de mais de 4 usuários. Caso tenha, exiba uma mensagem afirmando que o usuário é "popular".

```
8 - Verificando Conexões ★★
Crie um programa que permita verificar se um determinado usuário foi adicionado como amigo de mais de 4 usuários.
Caso tenha, exiba uma mensagem afirmando que o usuário é "popular".
Buscando por Fernando:
Usuario Fernando é amigo de 6 pessoas
Buscando por kaíke:
Usuario kaíke é amigo de 0 pessoas
```

9 - Amigos em Comum ★★

- Crie um programa que selecione dois perfis aleatórios e utilize sets para armazenar os amigos de cada um desses usuários do INFwebNET. Exiba os amigos em comum entre esses dois usuários, utilizando métodos e operação de sets.

```
9 Amigos em Comum ★★
Crie um programa que selecione dois perfis aleatórios e utilize sets para armazenar os amigos de cada um desses usuários do INFwebNET.
Exiba os amigos em comum entre esses dois usuários, utilizando métodos e operação de sets.
Usuário 1: Aline
Amigos de Aline: {'Caio', 'Clara', 'Fernando'}

Usuário 2: Tiago
Amigos de Tiago: {'Livia', 'Mateus', 'Aline'}

Amigos em comum entre Aline e Tiago: set()
```

10 - Conexões Exclusivas ★★

- Utilizando os sets do exercício anterior, exiba os amigos que são exclusivos de cada usuário, ou seja, aqueles que não são amigos em comum.

```
10 Conexões Exclusivas ★★
Utilizando os sets do exercício anterior, exiba os amigos que são exclusivos de cada usuário,
ou seja, aqueles que não são amigos em comum.
Usuário 1: Aline
Amigos de Aline: {'Caio', 'Clara', 'Fernando'}
Amigos exclusivos de Aline: {'Caio', 'Clara', 'Fernando'}

Usuário 2: Tiago
Amigos de Tiago: {'Livia', 'Mateus', 'Aline'}
Amigos exclusivos de Tiago: {'Livia', 'Mateus', 'Aline'}

Amigos em comum entre Aline e Tiago: set()
```

11 - Removendo Conexões ★★

- Permita que o usuário remova um amigo da lista de conexões de um membro do INFwebNET específico no dicionário criado no exercício 4.

11 Removendo Conexões ★★

Permita que o usuário remova um amigo da lista de conexões de um membro do INF específico no dicionário criado no exercício 4.

Lista

```
Usuario: Maria da Silva Amigos:
Usuario: José Santos Amigos:
Usuario: Ana Oliveira Amigos:
Usuario: João Pereira Amigos:
Usuario: Carlos Sousa Amigos:
Usuario: João Amigos: Ana, Carlos, Marcos
Usuario: Ana Amigos: João, Fernanda
Usuario: Carlos Amigos: João, Marcos, Caio
Usuario: Fernanda Amigos: Ana
Usuario: Marcos Amigos: João, Carlos
Usuario: Clara Amigos: Fernanda, Livia
Usuario: Livia Amigos: Clara, Lucas, Marcos
Usuario: Lucas Amigos: Livia, Fernando
Usuario: Fernando Amigos: Lucas, Aline
Usuario: Aline Amigos: Fernando, Caio, Clara
Usuario: Caio Amigos: Aline, Clara
Usuario: Sofia Amigos: Rafaela, Fernanda
Usuario: Rafaela Amigos: Sofia, Clara
Usuario: Mateus Amigos: Caio, Fernando, João
Usuario: Tiago Amigos: Mateus, Livia, Aline
Usuario: Paulo Amigos: Tiago, Fernanda, Lucas
Usuario: Juliana Amigos: Rafaela, Sofia, Marcos
Usuario: Pedro Amigos: João, Aline
Usuario: Amanda Amigos: Pedro, Caio, Livia
Usuario: Bruno Amigos: Ana, Marcos, Mateus
Usuario: Carla Amigos: Fernando, Tiago, Lucas
Usuario: Gabriela Amigos: Carla, Aline, Rafaela
Usuario: Thiago Amigos: João, Amanda, Lucas, Clara
Usuario: Renata Amigos: Clara, Fernando, Sofia
Usuario: Leonardo Amigos: Renata, Mateus, Caio
Usuario: Camila Amigos: Juliana, Bruno, Lucas, Caio
Usuario: Felipe Amigos: Renata, Carla
Usuario: Mariana Amigos: Felipe, Livia, Fernando
Usuario: Daniel Amigos: Camila, Paulo, Amanda
Remova um amigo pelo nome (pressione Enter para usar 'Marcos'):
De quem você deseja remover o amigo? (pressione Enter para usar 'João')
Amigos de João Antes: ['Ana', 'Carlos', 'Marcos']
Marcos foi removido da lista de amigos de João.
Amigos de João Depois: ['Ana', 'Carlos']
```

12 - Salvando o Progresso ★★

- Após adicionar ou remover amigos, salve o dicionário atualizado em um novo arquivo chamado "rede_INFNET_atualizado.txt".

12 Salvando o Progresso ★★

Após adicionar ou remover amigos, salve o dicionário atualizado em um novo arquivo chamado "rede_INFNET_atualizado.txt".

13 - Listando Usuários ★★

- Listar todos os usuários e seus amigos, exibindo informações formatadas.

13 Listando Usuários ★

Escreva um programa que leia o arquivo "rede_INFNET.txt" e imprima na tela a lista dos nomes de todos os usuários da rede social.

```
Nome do Usuario: João
Nome do Usuario: Ana
Nome do Usuario: Carlos
Nome do Usuario: Fernanda
Nome do Usuario: Marcos
Nome do Usuario: Clara
Nome do Usuario: Livia
Nome do Usuario: Lucas
Nome do Usuario: Fernando
Nome do Usuario: Aline
Nome do Usuario: Caio
Nome do Usuario: Sofia
Nome do Usuario: Rafaela
Nome do Usuario: Mateus
Nome do Usuario: Tiago
Nome do Usuario: Paulo
Nome do Usuario: Juliana
Nome do Usuario: Pedro
Nome do Usuario: Amanda
Nome do Usuario: Bruno
Nome do Usuario: Carla
Nome do Usuario: Gabriela
Nome do Usuario: nan
Nome do Usuario: Thiago
Nome do Usuario: Renata
Nome do Usuario: Leonardo
Nome do Usuario: Camila
Nome do Usuario: Felipe
Nome do Usuario: Mariana
Nome do Usuario: Daniel
```

14 - Verificando Amizades ★★★

- Verifique se um determinado usuário possui um amigo específico na lista de amigos.

14 Quantidade de Amigos ★★

Crie uma função que leia o arquivo "rede_INFNET.txt" e mostre quantos amigos cada usuário possui, imprimindo o nome do usuário e a quantidade de amigos.

```
Nome do Usuario: João | Quantidade de amigos: 3
Nome do Usuario: Ana | Quantidade de amigos: 2
Nome do Usuario: Carlos | Quantidade de amigos: 3
Nome do Usuario: Fernanda | Quantidade de amigos: 1
Nome do Usuario: Marcos | Quantidade de amigos: 2
Nome do Usuario: Clara | Quantidade de amigos: 2
Nome do Usuario: Livia | Quantidade de amigos: 3
Nome do Usuario: Lucas | Quantidade de amigos: 2
Nome do Usuario: Fernando | Quantidade de amigos: 2
Nome do Usuario: Aline | Quantidade de amigos: 3
Nome do Usuario: Caio | Quantidade de amigos: 2
Nome do Usuario: Sofia | Quantidade de amigos: 2
Nome do Usuario: Rafaela | Quantidade de amigos: 2
Nome do Usuario: Mateus | Quantidade de amigos: 3
Nome do Usuario: Tiago | Quantidade de amigos: 3
Nome do Usuario: Paulo | Quantidade de amigos: 3
Nome do Usuario: Juliana | Quantidade de amigos: 3
Nome do Usuario: Pedro | Quantidade de amigos: 2
Nome do Usuario: Amanda | Quantidade de amigos: 3
Nome do Usuario: Bruno | Quantidade de amigos: 3
Nome do Usuario: Carla | Quantidade de amigos: 3
Nome do Usuario: Gabriela | Quantidade de amigos: 3
Nome do Usuario: nan | Quantidade de amigos: 2
Nome do Usuario: Thiago | Quantidade de amigos: 4
Nome do Usuario: Renata | Quantidade de amigos: 3
Nome do Usuario: Leonardo | Quantidade de amigos: 3
Nome do Usuario: Camila | Quantidade de amigos: 4
Nome do Usuario: Felipe | Quantidade de amigos: 2
Nome do Usuario: Mariana | Quantidade de amigos: 3
Nome do Usuario: Daniel | Quantidade de amigos: 3
```

15 - Analisando Idades ★★★★★

- Calcule a média de idade dos usuários e exiba se a média é acima ou abaixo de um certo valor.

15 Usuários Mais Populares ★★★

Analise o arquivo "rede_INFNET_atualizado.txt" e identifique os 5 usuários que foram marcados como amigos pelo maior número de usuários cadastrados. Exiba o nome desses usuários e a quantidade de amigos que cada um possui.

Os 5 usuários mais marcados como amigos:

```
João: 6 amigos
Caio: 6 amigos
Clara: 6 amigos
Lucas: 6 amigos
Fernando: 6 amigos
```

16 - Gerando Relatórios ★★★★★

- Crie um relatório que sintetize as informações de usuários, suas idades, e a quantidade de amigos que possuem, exibindo tudo em formato legível.

```
16 Lidando com arquivos ★★
Explique com suas palavras a importância de utilizar o
recurso 'with' ao lidar com arquivos em Python.
R: O principal motivo de se usar o 'with' é evitar de deixar os arquivos abertos, pois sem ele você precisa fechar manualmente, as vezes o código pode dar erro antes de conseguir fechar o arquivo, deixando ele aberto impossibilitando que outro processo o abra, já com o 'with' mesmo dando erro ele mesmo fecha o arquivo sem salvar.
```

Considerações Finais

Este projeto fornece uma base sólida para entender a manipulação e análise de dados de perfis em redes sociais. Através das operações realizadas, foi possível observar a importância de uma base de dados consistente e a necessidade de filtrar informações para assegurar a qualidade dos dados armazenados. A simulação de conexões entre usuários e as funções de análise destacaram o valor de algoritmos eficientes em aplicações de rede.

Além disso, o trabalho ilustra o uso de operações fundamentais em Python para armazenar e processar dados de maneira persistente, abrindo caminho para futuras expansões. Funcionalidades adicionais, como recomendações de amigos ou algoritmos de análise de popularidade, poderiam aprimorar o projeto, tornando-o ainda mais aplicável a cenários reais. Em resumo, este exercício é um passo inicial valioso no desenvolvimento de ferramentas analíticas e sistemas de rede social, demonstrando a capacidade do Python para gerenciar informações e oferecer insights.