**Considerando que na área da computação existem muitos padrões e que esses padrões são os que garantem a interoperabilidade entre sistemas e equipamentos, encontre a RFC 8259 sobre JSON. Após encontra-la, faça um breve resumo a respeito do JSON Grammar abordado por essa RFC. Em seguida, para demonstrar que você entendeu, faça o que se pede:**

**- Escreva um programa para cadastrar alunos. Os dados solicitados serão: nome, idade e telefone. Esses dados deverão ser armazenados em um arquivo chamado ‘aluno.db.json’.**

**Além de cadastrar, o programa também precisa mostrar os alunos cadastrados. Por fim, o programa deve oferecer um mecanismo de busca pelo nome do aluno. Por exemplo: se eu digitar ‘Alex’, o programa precisa mostrar todos os nomes que comecem com ‘Alex’, como Alex e Alexandre, por exemplo.**

**RESPOSTA:**

**Resumo da Gramática JSON segundo RFC 8259:**

O JSON é composto por:

1. **Objetos**: Denotados por chaves ({}), que contêm pares de chave/valor. As chaves são strings, e os valores podem ser strings, números, objetos, arrays, valores booleanos (true ou false) ou null.

Exemplo:

json

{

"nome": "João",

"idade": 30,

"telefone": "1234-5678"

}

1. **Arrays**: Denotados por colchetes ([]), que contêm uma lista ordenada de valores. Os valores dentro de um array podem ser de qualquer um dos tipos descritos acima.

Exemplo:

json

[

"João",

"Maria",

30,

true

]

1. **Strings**: São sequências de caracteres envolvidas por aspas duplas (").
2. **Números**: Podem ser inteiros ou de ponto flutuante.
3. **Valores Booleanos**: Representados como true ou false.
4. **Null**: Representa a ausência de valor.

A gramática JSON é leve e sua estrutura é hierárquica, facilitando a representação de dados complexos de maneira legível.

**Programa para Cadastrar Alunos**

O programa a seguir implementa um sistema simples em Python que atende à solicitação de cadastrar, mostrar e buscar alunos pelo nome.

python

import json

import os

def cadastrar\_aluno(alunos):

nome = input("Digite o nome do aluno: ")

idade = input("Digite a idade do aluno: ")

telefone = input("Digite o telefone do aluno: ")

aluno = {

"nome": nome,

"idade": idade,

"telefone": telefone

}

alunos.append(aluno)

salvar\_alunos(alunos)

print("Aluno cadastrado com sucesso!")

def salvar\_alunos(alunos):

with open('aluno.db.json', 'w', encoding='utf-8') as f:

json.dump(alunos, f, ensure\_ascii=False, indent=4)

def mostrar\_alunos(alunos):

if alunos:

print("\nAlunos cadastrados:")

for idx, aluno in enumerate(alunos, start=1):

print(f"{idx}. Nome: {aluno['nome']}, Idade: {aluno['idade']}, Telefone: {aluno['telefone']}")

else:

print("Nenhum aluno cadastrado.")

def buscar\_aluno(alunos):

termo = input("Digite o nome (ou parte do nome) que deseja buscar: ")

resultados = [aluno for aluno in alunos if aluno['nome'].startswith(termo)]

if resultados:

print("\nResultados da busca:")

for aluno in resultados:

print(f"Nome: {aluno['nome']}, Idade: {aluno['idade']}, Telefone: {aluno['telefone']}")

else:

print("Nenhum aluno encontrado com esse termo.")

def carregar\_alunos():

if os.path.exists('aluno.db.json'):

with open('aluno.db.json', 'r', encoding='utf-8') as f:

return json.load(f)

return []

def main():

alunos = carregar\_alunos()

while True:

print("\nMenu:")

print("1. Cadastrar Aluno")

print("2. Mostrar Alunos")

print("3. Buscar Aluno")

print("4. Sair")

opcao = input("Escolha uma opção: ")

if opcao == '1':

cadastrar\_aluno(alunos)

elif opcao == '2':

mostrar\_alunos(alunos)

elif opcao == '3':

buscar\_aluno(alunos)

elif opcao == '4':

print("Saindo do programa.")

break

else:

print("Opção inválida. Tente novamente.")

if \_\_name\_\_ == "\_\_main\_\_":

main()