

Sistema de Gerenciamento de Alunos em Python - Uma Abordagem Orientada a Objetos

Apresentação detalhada do código e suas funcionalidades.



Visão Geral



Objetivo

Apresentar um sistema simples para gerenciar informações de alunos.

Linguagem

 Python

Estrutura

Classes: **Aluno** e **GerenciadorAlunos**

Classe Aluno

Atributos:

- `matricula`
- `nome`
- `curso`
- `nota1`, `nota2`, `nota3`

Métodos:

- `__init__`: Construtor da classe
- `obter_nota`: Adiciona novas notas a um aluno
- `media`: Calcula a média das notas.

```
class Aluno:

    def __init__(self, matricula, nome, curso, nota1, nota2, nota3):
        self.matricula = matricula
        self.nome = nome
        self.curso = curso
        self.nota1 = nota1
        self.nota2 = nota2
        self.nota3 = nota3

    def obter_nota(self, nome_ (function) def input(
        while True:
            try:
                if nota_atual
                entrada = input(
                    f"Nota do {nome_nota} atual ({nota_atual}): ")
                if entrada:
                    nota = float(entrada)
                else:
                    return nota_atual
            else:
                nota = float(input(f"Nota do {nome_nota}: "))
                if nota < 0 or nota > 10:
                    print("A nota deve estar entre 0 e 10.")
                    continue
                return nota
        except ValueError:
            print("Valor inválido. A nota deve ser um número real.")
            continue

    def media(self):
        return (self.nota1 + self.nota2 + self.nota3) / 3
```

Ln 237, Col 11 Spaces: 4 UTF-8 CRLF

```

class GerenciadorAlunos:

    def __init__(self):
        self.alunos = {}

    def mensagem_erro(self):
        print("Valor inválido. Matrículas dev

    def trilha(self):
        terminal_size = os.get_terminal_size(
        print("-" * terminal_size.columns)

    def cadastrar_novo_aluno(self):
        self.trilha()
        nome = input("Nome do aluno: ")
        curso = input("Curso: ")

        while True:
            try:
                nova_matricula = int(input("M
            except ValueError:
                self.mensagem_erro()
                continue

            if nova_matricula in self.alunos:
                print("Matrícula já está cada
            else:
                break

        aluno = Aluno(nova_matricula, no

```

Classe GerenciadorAlunos

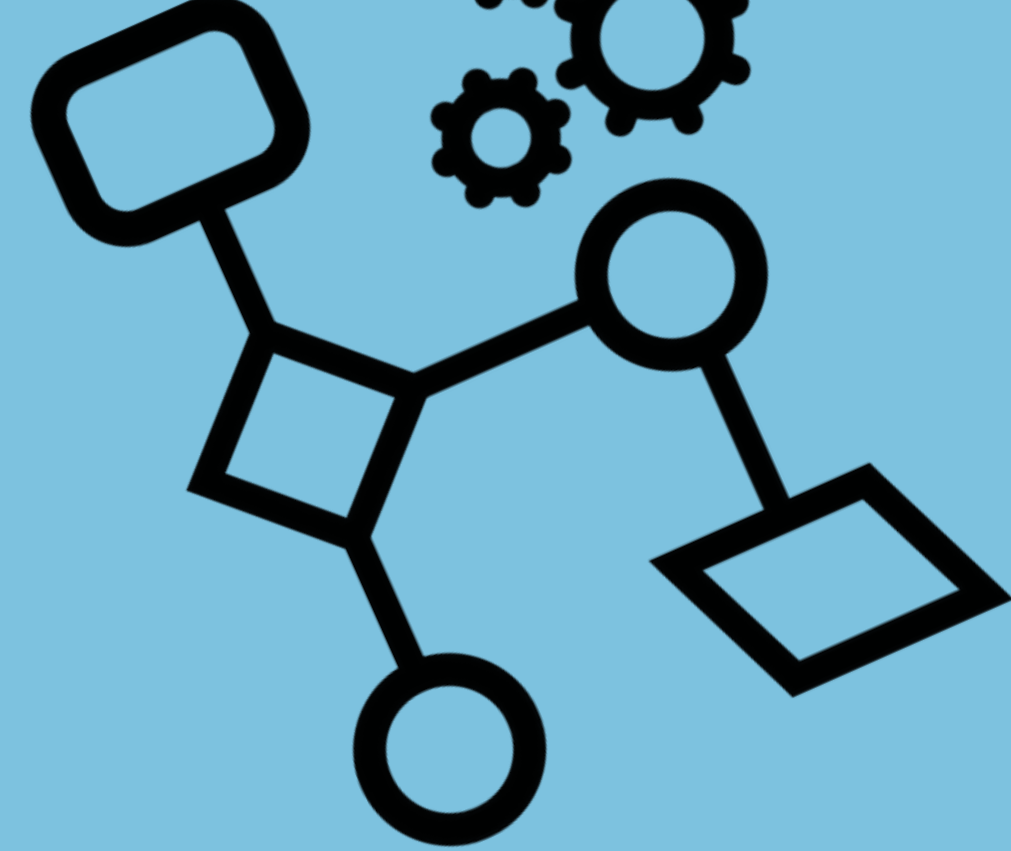
Atributos:

- `alunos`

Métodos:

- `cadastrar_aluno`: Adiciona um novo ao dicionário.
- `listar_alunos`: Exibe os dados de todos os alunos.
- `editar_aluno`: Edita os dados de um aluno específico.
- `excluir_aluno`: Remove um aluno do dicionário.
- `exibir_media`: Exibe a média de cada aluno.
- `carregar_dados`: Carrega dados dos alunos do arquivo JSON.
- `salvar_alunos`: Salva dados dos alunos no arquivo JSON.

Fluxo do Programa



Inicialização

O programa inicia a execução, onde o dicionário dentro do objeto GerenciadorAluno é inicializado como vazio

Menu Interativo

- Cadastrar um novo aluno
- Listar todos os alunos
- Editar um aluno
- Excluir um aluno
- Calcular e exibir a média
- Carregar dados
- Salvar dados
- Sair do programa

Processamento da Opção escolhida

O programa processa e retorna a opção que o usuário solicitou

Exibição do resultado

Após cada operação, o programa exibirá uma mensagem confirmando a ação realizada ou informando sobre algum erro



Obrigado!

