

Documento relacionado ao Exercício: Sistema Simples de Gerenciamento de Alunos

Principais conceitos que um aluno precisaria para implementar o **sistema procedural**

- ➔ Segue também uma breve explicação sobre cada conceito, mostrando partes relevantes do programa.

Conceitos principais (estudem, revisem cada um):

1. Variáveis e tipos de dados
2. Estruturas de controle (if, else, while)
3. Funções
4. Listas e dicionários
5. Entrada e saída de dados
6. Manipulação de strings
7. Conversão de tipos
8. Módulos (import)

Agora, vamos explicar cada conceito e mostrar onde ele é aplicado no programa:

1. Variáveis e tipos de dados: Usados para armazenar informações como nomes, matrículas, notas, etc. Exemplo:

```
alunos = [] # Lista para armazenar alunos
nome = input("Nome do aluno: ") # String
notas = [] # Lista para armazenar notas
```

2. Estruturas de controle: Usadas para controlar o fluxo do programa, como o menu principal e loops para entrada de dados. Exemplo:

```
while True:
    opcao = input("Escolha uma opção: ")
    if opcao == "1":
        cadastrar_aluno()
    elif opcao == "5":
        break
```

3. Funções: Usadas para organizar o código em blocos reutilizáveis. Exemplo:

```
def cadastrar_aluno():
    # Código para cadastrar um aluno

def listar_alunos():
    # Código para listar alunos
```

4. Listas e dicionários: Usados para armazenar coleções de dados, como a lista de alunos e as informações de cada aluno. Exemplo:

```
alunos = [] # Lista de alunos
aluno = {"nome": nome, "matricula": matricula, "curso": curso,
"notas": notas} # Dicionário representando um aluno
alunos.append(aluno)
```

5. Entrada e saída de dados: Usadas para interagir com o usuário, recebendo informações e exibindo resultados. Exemplo:

```
nome = input("Nome do aluno: ") # Entrada
print("Aluno cadastrado com sucesso!") # Saída
```

6. Manipulação de strings: Usada para formatar a saída e processar entradas do usuário. Exemplo:

```
print("{:<20} {:<15} {:<20} {:<20} {:<10}".format("Nome",
"Matrícula", "Curso", "Notas", "Média"))
```

7. Conversão de tipos: Usada para converter entradas do usuário (geralmente strings) em outros tipos de dados, como números. Exemplo:

```
nota = float(input("Digite uma nota: "))
```

8. Módulos: Usados para importar funcionalidades adicionais, como cálculos estatísticos. Exemplo:

```
import statistics

media = statistics.mean(notas)
```

Além desses conceitos básicos, o aluno também precisaria entender:

9. Escopo de variáveis: Compreender a diferença entre variáveis locais e globais. Exemplo:

```
alunos = [] # Variável global

def cadastrar_aluno():
    nome = input("Nome do aluno: ") # Variável local
```

10. Tratamento de erros básico: Lidar com entradas inválidas do usuário. Exemplo:

```
try:
    nota = float(input("Digite uma nota: "))
except ValueError:
    print("Por favor, digite um número válido.")
```

11. Formatação de strings: Usar f-strings ou o método .format() para criar saídas formatadas. Exemplo:

```
print(f"{aluno['nome']:<20} {aluno['matricula']:<15}  
{aluno['curso']:<20}")
```

Estes conceitos formam a base necessária para implementar o sistema procedural de gerenciamento de alunos.

Ao dominar esses conceitos, o aluno será capaz de criar um programa funcional que atenda aos requisitos especificados.