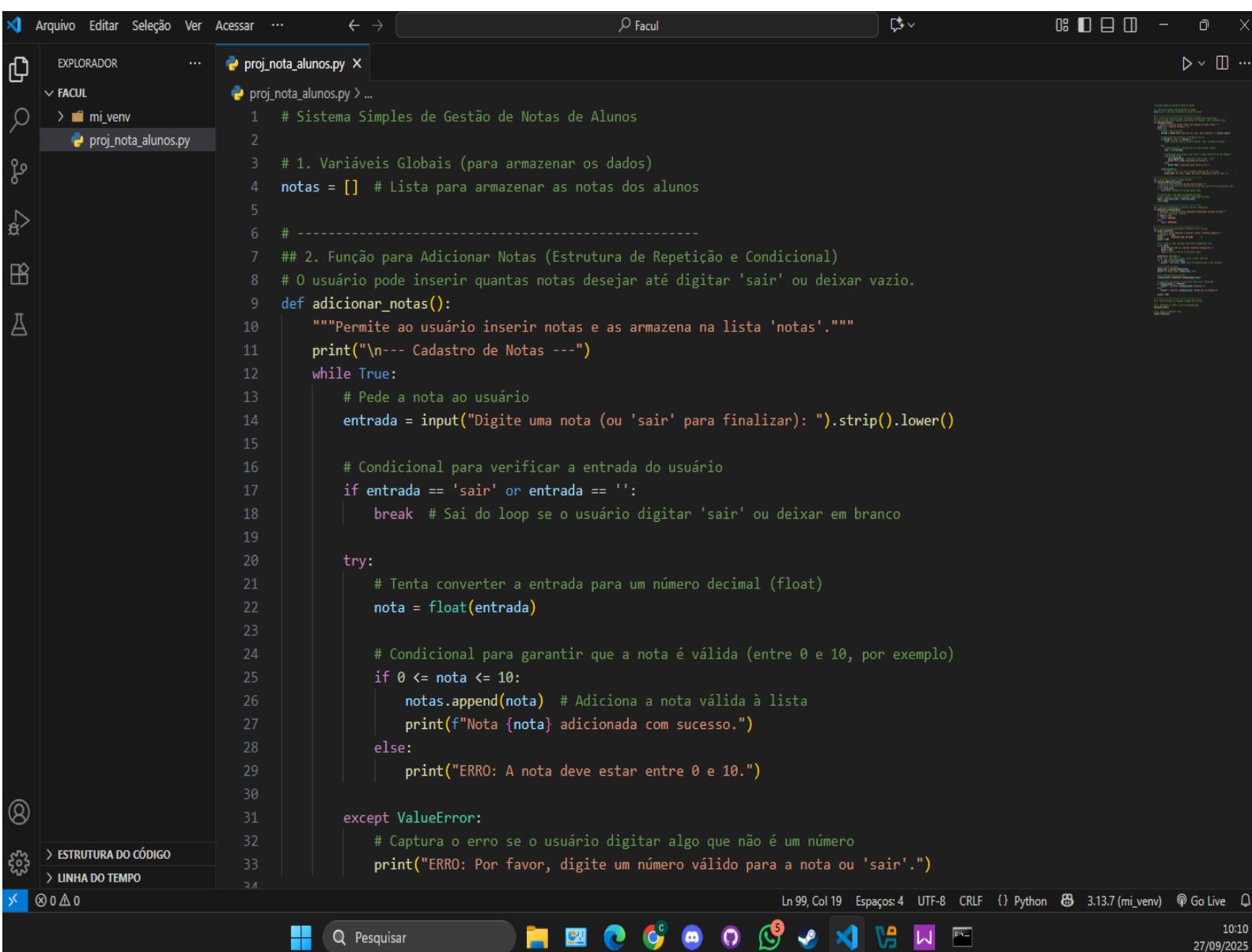


TRABALHO DE LINGUAGEM DE PROGRAMAÇÃO EM PYTHON

Nome: César Freitas Costa

Turma: ADS 2º Semestre



```
Arquivo Editar Seleção Ver Acessar ... ⏪ ⏫ Facul ⏪ ⏫ ... 🔍 Projeto Explorador ... EXPLORADOR FACUL mi_venv proj_nota_alunos.py proj_nota_alunos.py ... proj_nota_alunos.py ... 1 # Sistema Simples de Gestão de Notas de Alunos 2 3 # 1. Variáveis Globais (para armazenar os dados) 4 notas = [] # Lista para armazenar as notas dos alunos 5 6 # ----- 7 ## 2. Função para Adicionar Notas (Estrutura de Repetição e Condicional) 8 # O usuário pode inserir quantas notas desejar até digitar 'sair' ou deixar vazio. 9 def adicionar_notas(): 10     """Permite ao usuário inserir notas e as armazena na lista 'notas'.""" 11     print("\n--- Cadastro de Notas ---") 12     while True: 13         # Pede a nota ao usuário 14         entrada = input("Digite uma nota (ou 'sair' para finalizar): ").strip().lower() 15 16         # Condisional para verificar a entrada do usuário 17         if entrada == 'sair' or entrada == '': 18             break # Sai do loop se o usuário digitar 'sair' ou deixar em branco 19 20     try: 21         # Tenta converter a entrada para um número decimal (float) 22         nota = float(entrada) 23 24         # Condisional para garantir que a nota é válida (entre 0 e 10, por exemplo) 25         if 0 <= nota <= 10: 26             notas.append(nota) # Adiciona a nota válida à lista 27             print(f"Nota {nota} adicionada com sucesso.") 28         else: 29             print("ERRO: A nota deve estar entre 0 e 10.") 30 31     except ValueError: 32         # Captura o erro se o usuário digitar algo que não é um número 33         print("ERRO: Por favor, digite um número válido para a nota ou 'sair'.") 34
```

ESTRUTURA DO CÓDIGO LINHA DO TEMPO

X 0 △ 0 🔍 Pesquisar 10:10 27/09/2025

A lógica para desenvolver o sistema de gestão de notas foi baseada na divisão do problema em **funções**, o que torna o código organizado e reutilizável:

Armazenamento: Uma **lista** global (`notas`) foi criada para armazenar de forma dinâmica todas as notas inseridas pelo usuário.

adicionar_notas(): Utiliza um *loop while True* para permitir a inserção de múltiplas notas até que o usuário decida parar, criando uma **estrutura de repetição** controlada. Dentro do *loop*, uma **estrutura condicional (if/elif/else)** e um bloco **try/except** garantem que apenas notas numéricas válidas (entre 0 e 10) sejam adicionadas à lista.

Sequência do código: 1

The screenshot shows a code editor interface with a Python script named `proj_nota_alunos.py` open. The code defines four functions: `adicionar_notas()`, `calcular_media()`, `determinar_situacao()`, and `exibir_relatorio()`. The code includes comments explaining the purpose of each function and how it interacts with user input and conditional logic.

```
Arquivo Editar Seleção Ver Acessar ... ← → 🔍 Facul ⚙️ □ - X

EXPLORADOR ...
FACUL
> imagens
> mi_venv
proj_nota_alunos.py

proj_nota_alunos.py ...
1  #!/usr/bin/python3
2
3  def adicionar_notas():
4      try:
5          nota = float(input("Digite uma nota: "))
6          return nota
7      except ValueError:
8          print("ERRO: Por favor, digite um número válido para a nota ou 'sair'.")
9
10
11  # -----
12  ## 3. Função para Calcular a Média (Função)
13  def calcular_media(lista_notas):
14      """Calcula a média aritmética de uma lista de notas."""
15      # Condicional: Verifica se a lista não está vazia para evitar erro de divisão por zero
16      if not lista_notas:
17          return 0.0 # Retorna 0.0 se não houver notas
18
19      # A função sum() soma todos os elementos da lista.
20      # len() retorna o número de elementos (quantidade de notas).
21      media = sum(lista_notas) / len(lista_notas)
22      return media
23
24
25  # -----
26  ## 4. Função para Determinar a Situação (Função e Condicional)
27  def determinar_situacao(media):
28      """Determina a situação do aluno (Aprovado ou Reprovado) com base na média."""
29      # Estrutura Condicional (If/Else)
30      if media >= 7.0:
31          return "APROVADO"
32      else:
33          return "REPROVADO"
34
35
36  # -----
37  ## 5. Função Principal para Exibir o Relatório Final (Função)
38  def exibir_relatorio():
39      """Calcula a média, determina a situação e exibe o relatório completo."""
40      print("\n" + "*40")
41      print("          RELATÓRIO FINAL DO ALUNO          ")
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
```

ESTRUTURA DO CÓDIGO
LINHA DO TEMPO

Ln 99 Col 19 Espaços: 4 UTF-8 CRLF {} Python 3.13.7 (mi_venv) Go Live 10:11 27/09/2025

Sequência do código: 2

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with various icons for file operations like Open, Save, Find, and Settings. The main area displays a Python script named `proj_nota_alunos.py`. The code is color-coded for syntax, with numbers on the left indicating line numbers. The code itself is a function that prints a message if no grades are entered, then iterates through a list of grades to print each one with two decimal places. It calculates the average grade and prints it. Then, it determines the student's situation based on the average grade ('APROVADO' or 'REPROVADO') and prints a corresponding message. Finally, it prints a separator line of asterisks.

```
Arquivo Editar Seleção Ver Acessar ... ⏪ ⏫ Facul ⏮ ⏯ - X
EXPLORADOR ...
FACUL
    > imagens
    > mi_venv
    proj_nota_alunos.py
proj_nota_alunos.py ...
60 def exibir_relatorio():
61     print("Nenhuma nota foi inserida. Relatório indisponível.")
62     print("*40)
63     return # Encerra a função se não houver notas
64
65     print("Notas Inseridas:")
66     # Loop 'for' para iterar sobre a lista e exibir cada nota
67     for i, nota in enumerate(notas):
68         print(f"- Nota {i+1}: {nota:.2f}") # Formatação para 2 casas decimais
69
70     # 5.2. Cálculo da Média
71     media_final = calcular_media(notas)
72     print(f"\n- Média Final: {media_final:.2f}")
73
74     # 5.3. Determinação da Situação
75     situacao_aluno = determinar_situacao(media_final)
76
77     # Condicional para exibir a situação de forma clara e destacada
78     if situacao_aluno == "APROVADO":
79         print(f"- Situação: {situacao_aluno} (Parabéns!)")
80     else:
81         print(f"- Situação: {situacao_aluno} (Estude mais na próxima!)")
82
83     print("*40)
84
85     # -----
86     ## 6. Fluxo Principal do Programa (Chamada das Funções)
87
88     # 6.1. Adicionar as notas (o usuário interage aqui)
89     adicionar_notas()
90
91     # 6.2. Exibir o relatório final
92     exibir_relatorio()
```

ESTRUTURA DO CÓDIGO
LINHA DO TEMPO

Ln 99, Col 19 Espaços: 4 UTF-8 CRLF () Python 3.13.7 (mi_venv) Go Live

10:11
27/09/2025

Código em funcionamento no Visual Studio Code. 1

Introduzindo os valores:

The screenshot shows the Visual Studio Code interface with a Python script named `proj_nota_alunos.py` open in the editor. The code defines a function `exibir_relatorio()` that handles the input of student grades and calculates the average. The terminal below shows the execution of the script and the input of several grades.

```
proj_nota_alunos.py
60 def exibir_relatorio():
61     print("Nenhuma nota foi inserida. Relatório indisponível.")
62     print("*"*40)
63     return # Encerra a função se não houver notas
64
65     print("Notas Inseridas:")
66     # Loop 'for' para iterar sobre a lista e exibir cada nota
67     for i, nota in enumerate(notas):
68         print(f"- Nota {i+1}: {nota:.2f}") # Formatação para 2 casas decimais
69
70     # 5.2. Cálculo da Média
71     media_final = calcular_media(notas)
72     print(f"\n- Média Final: {media_final:.2f}")

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS
PS D:\Facul> & D:/Facul/mi_venv/Scripts/Activate.ps1
(mi_venv) PS D:\Facul> & D:/Facul/mi_venv/Scripts/python.exe d:/Facul/proj_nota_alunos.py

--- Cadastro de Notas ---
Digite uma nota (ou 'sair' para finalizar): 10
Nota 10.0 adicionada com sucesso.
Digite uma nota (ou 'sair' para finalizar): 5
Nota 5.0 adicionada com sucesso.
Digite uma nota (ou 'sair' para finalizar): 7
Nota 7.0 adicionada com sucesso.
Digite uma nota (ou 'sair' para finalizar): 9
Nota 9.0 adicionada com sucesso.
Digite uma nota (ou 'sair' para finalizar): 5
```

Código em funcionamento no Visual Studio Code. 2

Retorno de resposta ok!

The screenshot shows the Visual Studio Code interface with a Python script named `proj_nota_alunos.py` open in the editor. The code defines a function `exibir_relatorio` that handles user input for grades and calculates a final average. The terminal below shows the execution of the script and its output, which includes a report of the entered grades and a final average calculation.

```
proj_nota_alunos.py
proj_nota_alunos.py > ...
60  def exibir_relatorio():
61      print("Nenhuma nota foi inserida. Relatório indisponível.")
62      print("*40)
63      return # Encerra a função se não houver notas
64
65  print("Notas Inseridas:")
66  # Loop 'for' para iterar sobre a lista e exibir cada nota
67  for i, nota in enumerate(notas):
68      print(f"- Nota {i+1}: {nota:.2f}") # Formatação para 2 casas decimais
69
70  # 5.2. Cálculo da Média
71  media_final = calcular_media(notas)
72  print(f"\n- Média Final: {media_final:.2f}")
73
74
75
76
77
78
79
80

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS
PS D:\Facul> & D:/Facul/mi_venv/Scripts/Activate.ps1
Digite uma nota (ou 'sair' para finalizar): 7
Nota 7.0 adicionada com sucesso.
Digite uma nota (ou 'sair' para finalizar): 9
Nota 9.0 adicionada com sucesso.
Digite uma nota (ou 'sair' para finalizar): 5
Nota 5.0 adicionada com sucesso.
Digite uma nota (ou 'sair' para finalizar): sair

=====
RELATÓRIO FINAL DO ALUNO
=====
Notas Inseridas:
- Nota 1: 10.00
- Nota 2: 5.00
- Nota 3: 7.00
- Nota 4: 9.00
- Nota 5: 5.00

- Média Final: 7.20
- Situação: APROVADO (Parabéns!)
=====

ESTRUTURA DO CÓDIGO LINHA DO TEMPO
(mi_venv) PS D:\Facul> |
```

Ln 99, Col 19 Espaços: 4 UTF-8 CRLF () Python 3.13.7 (mi_venv) Go Live 10:13 27/09/2025