



**CENTRO UNIVERSITARIO ANHANGUERA PITAGORAS UNOPAR DE CAMPO GRANDE**

**CURSO: ENGENHARIA DE SOFTWARE**

**DISCIPLINA: LINGUAGEM DE PROGRAMAÇÃO**

**ATIVIDADE PRÁTICA – UNIDADE 1, AULA 4**

**FUNÇÕES EM PYTHON**

Aluno: Edmar Radanovis.

RA: 2025223493

Polo: Itapira / SP-UN944038

---

**Ano 2025 / 2º Semestre**

**ANHANGUERA EDUCACIONAL**

**ENGENHARIA DE SOFTWARE**

DISCIPLINA: Linguagem de Programação

PROFESSOR: Anderson I. S. Abreu / Vanessa Matias Leite

ALUNO: Edmar Radanovis

RA: 2025223493

TÍTULO: Relatório da Aula Prática – Funções em Python – Unidade 1 – Aula 4

POLO: Itapira / SP-UN944038

Monte Sião, 07 de Setembro de 2025.

## RESULTADOS DA ATIVIDADE PRÁTICA

### Proposta:



#### Introdução a Linguagem Python - Funções.

- Você foi contratado para desenvolver um sistema simples de gestão de notas de alunos. O sistema deve permitir que o usuário adicione notas, calcule a média das notas, determine a situação do aluno (aprovado ou reprovado), e exiba um relatório final. Utilize estruturas condicionais, de repetição e funções.
  - o Cadastro de Notas:
    - O sistema deve permitir que o usuário insira as notas dos alunos.
    - As notas devem ser armazenadas em uma lista.
  - o Cálculo da Média:
    - O sistema deve calcular a média das notas inseridas.
  - o Determinação da Situação:
    - Se a média for maior ou igual a 7, o aluno está aprovado.
    - Se a média for menor que 7, o aluno está reprovado.
  - o Relatório Final:
    - Exibir as notas inseridas, a média e a situação do aluno

link do repositório no GitHub:

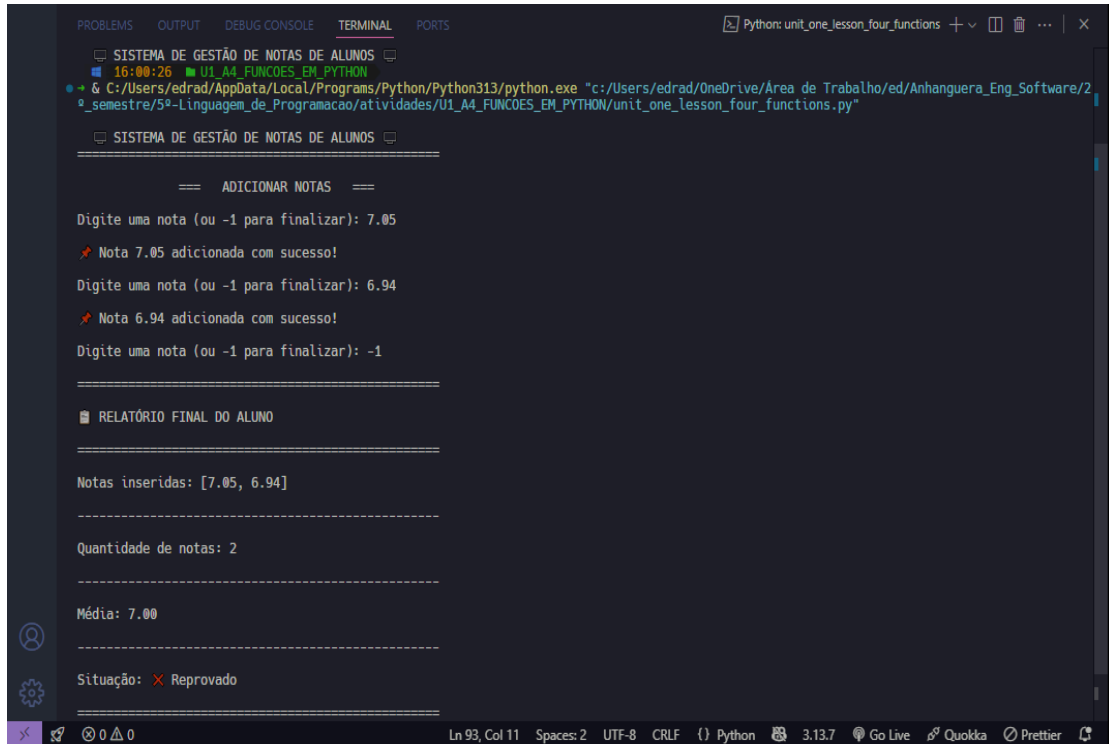
[https://github.com/ed-radanovis/-Eng\\_Software\\_L-P\\_U1-A4\\_09-2025.git](https://github.com/ed-radanovis/-Eng_Software_L-P_U1-A4_09-2025.git)

```
File Edit Selection View ... U1_A4_FUNCÕES_EM_PYTHON
EXPLORER
unit_one_lesson_...
U1_A4_FUNCÕES_EM_PYT...
ATIVIDADE_PRA...
roteiro_U1_A4_FU...
unit_one_lesson_f...

unit_one_lesson_four_functions.py
1  # -*- coding: utf-8 -*-
2  # Sistema de Gestão de Notas de Alunos
3  # Desenvolvido para a disciplina de Linguagem de Programação
4
5  # Função para adicionar notas
6  def adicionar_notas():
7      """
8      Solicita ao usuário que insira as notas do aluno e as armazena em uma lista.
9      Retorna a lista de notas.
10     """
11     notas = []
12     print("===== ADICIONAR NOTAS ===== \n")
13
14     while True:
15         try:
16             nota = float(input("Digite uma nota (ou -1 para finalizar): "))
17             if nota == -1:
18                 break
19             elif nota < 0 or nota > 10:
20                 print("\n❗ Por favor, digite uma nota entre 0 e 10.\n")
21             else:
22                 notas.append(nota)
23                 print(f"✅ Nota {nota} adicionada com sucesso!\n")
24         except ValueError:
25             print("\n❗ Entrada inválida. Digite um número.\n")
26
27     return notas
28
29 # Função para calcular a média
30 def calcular_media(notas):
31     """
32     Calcula a média das notas fornecidas.
33     Retorna a média aritmética das notas.
34     """
35     if len(notas) == 0:
36         return 0
37     return sum(notas) / len(notas)
38
39 # Função para determinar a situação
40 def determinar_situacao(media):
41     """
42     Determina a situação do aluno com base na média.
43     Retorna 'Aprovado' se média ≥ 7, caso contrário 'Reprovado'.
44     """
45     if media ≥ 7:
46         return "✅ Aprovado"
47     else:
48         return "❌ Reprovado"
49
50 # Função para exibir relatório
51 def exibir_relatorio(notas, media, situacao):
52     """
53     Exibe um relatório completo com todas as notas, a média e a situação do aluno.
54     """
55     print("\n" + "="*50)
56     print("\n📄 RELATÓRIO FINAL DO ALUNO")
57     print("\n" + "="*50 + "\n")
58     print(f"Notas inseridas: {notas}")
59     print(f"\n" + "="*50)
60     print(f"Quantidade de notas: {len(notas)}")
61     print("\n" + "-"*50)
62     print(f"Média: {media:.2f}")
63     print("\n" + "-"*50)
64     print(f"Situação: {situacao}")
65     print("\n" + "="*50 + "\n\n")
66
67 # Função principal
68 def main():
69     """
70     Função principal que orquestra todo o sistema de gestão de notas.
71     """
72     print("\n🎓 SISTEMA DE GESTÃO DE NOTAS DE ALUNOS 🎓")
73     print("="*50 + "\n")
74
75     # Adicionar notas
76     notas = adicionar_notas()
77
78     if not notas:
79         print("\n! Nenhuma nota foi inserida. Encerrando o sistema. !\n")
80         return
81
82     # Calcular média
83     media = calcular_media(notas)
84
85     # Determinar situação
86     situacao = determinar_situacao(media)
87
88     # Exibir relatório
89     exibir_relatorio(notas, media, situacao)
90
91 # Executar o programa
92 if __name__ == "__main__":
93     main()

> OUTLINE
> TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python: unit_one_lesson_four_functions
15:48:30 U1_A4_FUNCÕES_EM_PYTHON 8 83ms
C:\Users\edrad\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/edrad/OneDrive/Área de Trabalho/ed/Anhanguera_Eng_Software/2º_Semestre/5º-Linguagem_de_Programacao/atividades/U1_A4_FUNCÕES_EM_PYTHON/unit_one_lesson_four_functions.py"
SISTEMA DE GESTÃO DE NOTAS DE ALUNOS
===== ADICIONAR NOTAS =====
Digite uma nota (ou -1 para finalizar): 10
✅ Nota 10.0 adicionada com sucesso!
Digite uma nota (ou -1 para finalizar): 7
✅ Nota 7.0 adicionada com sucesso!
Digite uma nota (ou -1 para finalizar): -1
=====
📄 RELATÓRIO FINAL DO ALUNO
=====
Notas inseridas: [10.0, 7.0]
-----
Quantidade de notas: 2
-----
Média: 8.50
-----
Situação: ✅ Aprovado
-----
15:48:44 U1_A4_FUNCÕES_EM_PYTHON 8 13.869s
Ln 93, Col 11 Spaces: 2 UTF-8 CRLF Python 3.13.7 Go Live Quokka Prettier
```

=> Figura 1: Print da tela com o código e resultado da execução do script no terminal.



```
Python: unit_one_lesson_four_functions + v [ ] [ ] ... | x
SISTEMA DE GESTÃO DE NOTAS DE ALUNOS
16:00:26 U1_A4_FUNC0ES_EM_PYTHON
C:\Users\edrad\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/edrad/OneDrive/Área de Trabalho/ed/Anhanguera_Eng_Software/2_
semestre/5ª-Linguagem_de_Programacao/atividades/U1_A4_FUNC0ES_EM_PYTHON/unit_one_lesson_four_functions.py"

SISTEMA DE GESTÃO DE NOTAS DE ALUNOS

=====
                ADICIONAR NOTAS
=====

Digite uma nota (ou -1 para finalizar): 7.05
✖ Nota 7.05 adicionada com sucesso!

Digite uma nota (ou -1 para finalizar): 6.94
✖ Nota 6.94 adicionada com sucesso!

Digite uma nota (ou -1 para finalizar): -1

=====

RELATÓRIO FINAL DO ALUNO

=====

Notas inseridas: [7.05, 6.94]

-----

Quantidade de notas: 2

-----

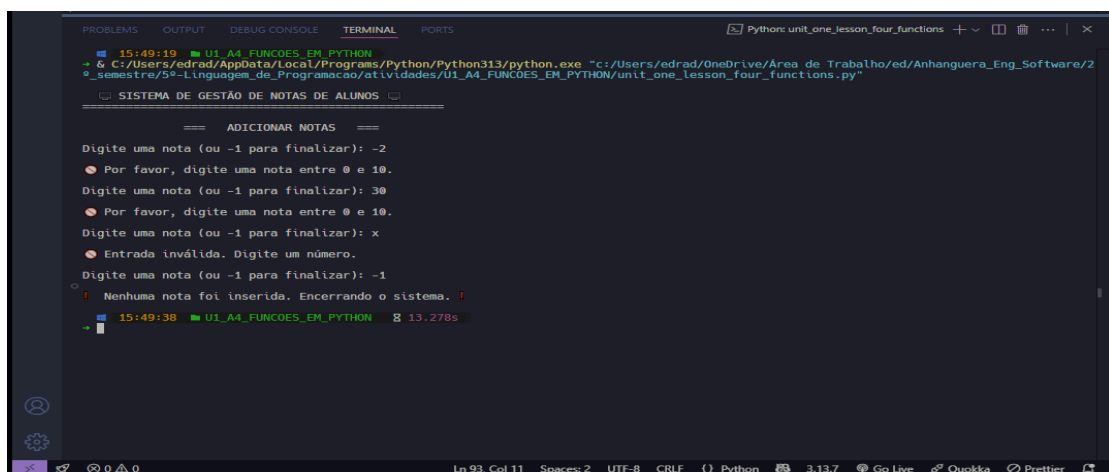
Média: 7.00

-----

Situação: ✖ Reprovado

=====
```

=> Figura 2: Print da tela com parte do teste e o resultado da execução do script no terminal.



```
Python: unit_one_lesson_four_functions + v [ ] [ ] ... | x
SISTEMA DE GESTÃO DE NOTAS DE ALUNOS
15:49:19 U1_A4_FUNC0ES_EM_PYTHON
C:\Users\edrad\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/edrad/OneDrive/Área de Trabalho/ed/Anhanguera_Eng_Software/2_
semestre/5ª-Linguagem_de_Programacao/atividades/U1_A4_FUNC0ES_EM_PYTHON/unit_one_lesson_four_functions.py"

SISTEMA DE GESTÃO DE NOTAS DE ALUNOS

=====
                ADICIONAR NOTAS
=====

Digite uma nota (ou -1 para finalizar): -2
✖ Por favor, digite uma nota entre 0 e 10.

Digite uma nota (ou -1 para finalizar): 30
✖ Por favor, digite uma nota entre 0 e 10.

Digite uma nota (ou -1 para finalizar): x
✖ Entrada inválida. Digite um número.

Digite uma nota (ou -1 para finalizar): -1
✖ Nenhuma nota foi inserida. Encerrando o sistema.

15:49:38 U1_A4_FUNC0ES_EM_PYTHON 8 13.278s
```

=> Figura 3: Print da tela com parte do teste e o resultado da execução do script no terminal.

A lógica utilizada para realizar a atividade baseia-se na modularização do código em funções específicas, conforme solicitado no roteiro, utilizando estruturas condicionais (if/else), de repetição (while) e listas para manipulação de dados.

- Na função `adicionar_notas()`, um loop `while` permite a inserção contínua de notas válidas (entre 0 e 10), armazenando-as em uma lista, com validação de entrada via `try-except` e interrupção ao digitar -1.
- A função `calcular_media()` computa a média aritmética usando `sum()` e `len()`, retornando 0 se a lista estiver vazia para evitar erros.
- `determinar_situacao()` aplica uma condicional simples: aprova se a média for  $\geq 7$ , senão reprova.
- `exibir_relatorio()` formata e imprime os resultados de forma organizada.
- A função `main()` orquestra o fluxo: coleta notas, calcula média e situação, e exibe o relatório apenas se houver dados, garantindo robustez.

Essa estrutura promove reutilização de código e atende aos objetivos de compreender e aplicar funções em Python.