

IFMA- São José de Ribamar.

São José de Ribamar, 31 de agosto de 2025.

Discente: Renato da Fonseca Oliveira

ATIVIDADE AVALIATIVA PYTHON

1. Peça ao usuário 5 números, armazene em uma lista e mostre:

- A soma de todos os números
- O maior e o menor número
- (use for e if/else, não max() e min()).
- É para o programa ficar em execução enquanto o usuário não digitar a palavra sair.
- Mostre os resultados usando f-string e format().

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python

# 1. Peça ao usuário 5 números, armazene em uma lista e mostre:
# • A soma de todos os números
# • O maior e o menor número
# • (use for e if/else, não max() e min()).
# • É para o programa ficar em execução enquanto o usuário não digitar a palavra sair.
# • Mostre os resultados usando f-string e format().

while True:
    numeros = []

    print("\nDigite 5 números ou 'sair' a qualquer momento para encerrar:")

    i = 0
    while i < 5:
        entrada = input(f"Digite o {i+1}º número: ")
        if entrada.lower() == 'sair':
            print("Encerrando o programa. Até mais!")
            exit()

        try:
            numero = float(entrada)
            numeros.append(numero)
            i += 1 # Só avança se a entrada for válida
        except ValueError:
            print("Entrada inválida. Por favor, digite um número válido.")

# Inicializa variáveis
soma = 0
maior = numeros[0]
```

```
menor = numeros[0]
# Processa os números
for num in numeros:
    soma += num
    if num > maior:
        maior = num
    elif num < menor:
        menor = num

# Exibe os resultados
print("\nResultados:")
print("A soma dos números é: {}".format(soma))
print(f"O maior número é: {maior}")
print(f"O menor número é: {menor}")
```

2. Dada a tupla notas = (7.5, 8.0, 5.0, 9.0, 6.5), calcule a média e classifique o desempenho da turma:

- "Excelente" se média ≥ 9
- "Bom" se $7 \leq \text{média} < 9$
- "Regular" se $5 \leq \text{média} < 7$
- "Ruim" se média < 5
- É para o programa ficar em execução enquanto o usuário não digitar a palavra sair.
- Mostre os resultados usando f-string e format().

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python

# 2. Dada a tupla notas = (7.5, 8.0, 5.0, 9.0, 6.5), calcule a média e classifique o
desempenho da turma:
# • "Excelente" se média  $\geq 9$ 
# • "Bom" se  $7 \leq \text{média} < 9$ 
# • "Regular" se  $5 \leq \text{média} < 7$ 
# • "Ruim" se média  $< 5$ 
# • É para o programa ficar em execução enquanto o usuário não digitar a palavra sair.
# • Mostre os resultados usando f-string e format().

notas = (7.5, 8.0, 5.0, 9.0, 6.5)
while True:
    comando = input("Digite 'sair' para encerrar ou pressione Enter para continuar: ")
    if comando.lower() == 'sair':
        break
    media = sum(notas) / len(notas)
    if media >= 9:
        desempenho = "Excelente"
```

```
elif 7 <= media < 9:
    desempenho = "Bom"
elif 5 <= media < 7:
    desempenho = "Regular"
else:
    desempenho = "Ruim"
print(f"A média da turma é {media:.2f}")
print(f"Tendo a média {media:.2f} a turma teve um desempenho '{desempenho}'.format(media, desempenho))
```

3. Crie um dicionário com 3 alunos e suas notas:

- por exemplo: alunos = {"Ana": 8, "João": 6, "Pedro": 9}
- Percorra o dicionário com for e mostre cada aluno e sua situação: "Aprovado" (nota ≥ 7) ou "Reprovado".
- É para o programa ficar em execução enquanto o usuário não digitar a palavra sair.
- Mostre os resultados usando f-string e format().

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python

# 3. Crie um dicionário com 3 alunos e suas notas:
# • por exemplo: alunos = {"Ana": 8, "João": 6, "Pedro": 9}
# • Percorra o dicionário com for e mostre cada aluno e sua situação: "Aprovado" (nota
# ≥ 7) ou
# "Reprovado".
# • É para o programa ficar em execução enquanto o usuário não digitar a palavra sair.
# • Mostre os resultados usando f-string e format().

alunos = {"Ana": 8, "João": 6, "Pedro": 9}
while True:
    comando = input("Digite 'sair' para encerrar ou pressione Enter para continuar: ")
    if comando.lower() == 'sair':
        break
    for aluno, nota in alunos.items():
        situacao = "Aprovado(a)" if nota >= 7 else "Reprovado(a)"
        print(f"O(a) aluno(a) {aluno} teve a nota {nota}.")
        print(f"{aluno} com nota {nota} está com a situação '{situacao}'.format(aluno, nota, situacao))
```

4. Peça ao usuário uma palavra e conte quantas vogais (a, e, i, o, u) ela possui.

- É para o programa ficar em execução enquanto o usuário não digitar a palavra sair.
- Mostre os resultados usando f-string e format().

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python

# 4. Peça ao usuário uma palavra e conte quantas vogais (a, e, i, o, u) ela possui.
# • É para o programa ficar em execução enquanto o usuário não digitar a palavra sair.
# • Mostre os resultados usando f-string e format().

while True:
    palavra = input("Digite uma palavra (ou 'sair' para encerrar): ").strip().lower()

    if palavra == 'sair':
        print("Encerrando o programa. Até mais!")
        break

    vogais = 'aeiou'
    contador_vogais = sum(1 for letra in palavra if letra in vogais)

    # Usando f-string
    print(f"A palavra '{palavra}' possui {contador_vogais} vogais.")

    # Usando format()
    print("A palavra '{}' possui {} vogais.".format(palavra, contador_vogais))
```

5. Peça ao usuário dois números float (números reais) e crie uma função que calcule:

- soma
- subtração
- multiplicação
- divisão
- É para o programa ficar em execução enquanto o usuário não digitar a palavra sair.
- Mostre os resultados usando f-string e format().

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python

# 5. Peça ao usuário dois números float (números reais) e crie uma função que calcule:
# • soma
# • subtração
# • multiplicação
```

```
# • divisão
# • É para o programa ficar em execução enquanto o usuário não digitar a palavra sair.
# • Mostre os resultados usando f-string e format().

def calcular_operacoes(num1, num2):
    soma = num1 + num2
    subtracao = num1 - num2
    multiplicacao = num1 * num2
    divisao = num1 / num2 if num2 != 0 else 'Indefinido (divisão por zero)'
    return soma, subtracao, multiplicacao, divisao

while True:
    entrada1 = input("Digite o primeiro número (ou 'sair' para encerrar): ").strip().lower()
    if entrada1 == 'sair':
        print("Encerrando o programa. Até mais!")
        break
    entrada2 = input("Digite o segundo número (ou 'sair' para encerrar): ").strip().lower()
    if entrada2 == 'sair':
        print("Encerrando o programa. Até mais!")
        break

    try:
        num1 = float(entrada1)
        num2 = float(entrada2)
    except ValueError:
        print("Por favor, digite números válidos.")
        continue

    soma, subtracao, multiplicacao, divisao = calcular_operacoes(num1, num2)

    # Resultados usando f-string e format()
    print("Resultados para os números {} e {}".format(num1, num2))
    print("Soma: {}".format(soma))
    print("Subtração: {}".format(subtracao))
    print(f"Multiplicação: {multiplicacao}")
    print(f"Divisão: {divisao}")
```

6. Crie um programa que peça nomes ao usuário e guarde em uma lista até que ele digite "sair". Ao final, mostre todos os nomes digitados.

- Mostre os resultados usando f-string e format().

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python
```

```
# 6. Crie um programa que peça nomes ao usuário e guarde em uma lista até que ele digite
"sair".
# Ao final, mostre todos os nomes digitados.
# • Mostre os resultados usando f-string e format().

nomes = []
while True:
    nome = input("Digite um nome (ou 'sair' para encerrar): ").strip()
    if nome.lower() == 'sair':
        break
    nomes.append(nome)
print("Nomes digitados:")
for i, nome in enumerate(nomes, start=1):
    print("{}.".format(i, nome))
print(f"Total de nomes digitados: {len(nomes)}")
```

7. Escreva um programa que percorra uma string e conte quantos caracteres são letras maiúsculas:
- Mostre os resultados usando f-string e format().

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python

# 7. Escreva um programa que percorra uma string e conte quantos caracteres são letras
maiúsculas.
# • Mostre os resultados usando f-string e format().

string = input("Digite uma string: ").strip()
contador_maiusculas = sum(1 for char in string if char.isupper())
print("Número de letras maiúsculas na string: {}".format(contador_maiusculas))
print(f"{contador_maiusculas} é o número de letras maiúsculas na string.")
```

8. Crie uma função chamada busca_preco que receba um dicionário de produtos e preços, de um produto específico, retornando o preço ou a mensagem "Produto não encontrado".
- Mostre os resultados usando f-string e format().

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python
```

```
# 8. Crie uma função chamada busca_preco que receba um dicionário de produtos e preços,
# de um produto específico, retornando o preço ou a mensagem "Produto não encontrado".
# • Mostre os resultados usando f-string e format().

def busca_preco(produtos, produto_buscado):
    if produto_buscado in produtos:
        return f"O preço do {produto_buscado} é R$ {produtos[produto_buscado]:.2f}"
    else:
        return "Produto não encontrado"

# Exemplo de uso
produtos = {
    "maçã": 3.50,
    "banana": 2.00,
    "laranja": 4.00
}

produto_buscado = "banana"
resultado = busca_preco(produtos, produto_buscado)
print(resultado) # Usando f-string
print("O preço do {} é R$ {:.2f}".format(produto_buscado, produtos.get(produto_buscado,
0))) # Usando format()

# Se o produto não for encontrado, a segunda linha mostrará "O preço do banana é R$ 0.00"
```

9. Crie uma função chamada `analise_lista(numeros)` que receba uma lista de inteiros e retorne a soma, o maior e o menor número.

- O programa deve pedir 5 números ao usuário, guardar em uma lista e chamar a função.
- Mostre os resultados usando f-string e `format()`.

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python

# 9. Crie uma função chamada analise_lista(numeros) que receba uma lista de inteiros e
retorne a soma,
# o maior e o menor número.
# • O programa deve pedir 5 números ao usuário, guardar em uma lista e chamar a
função.
# • Mostre os resultados usando f-string e format().

def analise_lista(numeros):
    soma = sum(numeros)
    maior = max(numeros)
    menor = min(numeros)
    return soma, maior, menor
```

```
while True:
    numeros = []
    print("\nDigite 5 números ou 'sair' a qualquer momento para encerrar:")
    i = 0
    while i < 5:
        entrada = input(f"Digite o {i+1}º número: ")
        if entrada.lower() == 'sair':
            print("Encerrando o programa. Até mais!")
            exit()

        try:
            numero = int(entrada)
            numeros.append(numero)
            i += 1 # Só avança se a entrada for válida
        except ValueError:
            print("Entrada inválida. Por favor, digite um número inteiro válido.")

    soma, maior, menor = analise_lista(numeros)
    print("\nResultados:")
    print("A soma dos números é: {}".format(soma))
    print(f"O maior número é: {maior}")
    print(f"O menor número é: {menor}")
```

10. Crie uma função chamada `media_turma(notas)` que receba uma tupla de notas (float) e retorne a média e a classificação da turma:

- "Excelente" se $\text{média} \geq 9$
- "Bom" se $7 \leq \text{média} < 9$
- "Regular" se $5 \leq \text{média} < 7$
- "Ruim" se $\text{média} < 5$
- Mostre os resultados usando f-string e `format()`.

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python

# 10. Crie uma função chamada media_turma(notas) que receba uma tupla de notas (float)
# e retorne a média e a classificação da turma:
# • "Excelente" se média ≥ 9
# • "Bom" se 7 ≤ média < 9
# • "Regular" se 5 ≤ média < 7
# • "Ruim" se média < 5
# • Mostre os resultados usando f-string e format().

def media_turma(notas):
    media = sum(notas) / len(notas)
```



```
if media >= 9:
    classificacao = "Excelente"
elif 7 <= media < 9:
    classificacao = "Bom"
elif 5 <= media < 7:
    classificacao = "Regular"
else:
    classificacao = "Ruim"

return f"Média: {media:.2f}, Classificação: {classificacao}"

# Exemplo de uso
notas = (8.5, 9.0, 7.5, 6.0, 10.0)
resultado = media_turma(notas)
print(resultado)
```

11. Crie uma função chamada `resultado_alunos(alunos)` que receba um dicionário no formato `{nome: nota}` e retorne outro dicionário com `{nome: "Aprovado"/"Reprovado"}`.

- Mostre os resultados usando f-string e `format()`.

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python

# 11. Crie uma função chamada resultado_alunos(alunos) que receba um dicionário no formato
# {nome: nota} e retorne outro dicionário com {nome: "Aprovado"/"Reprovado"}.
# • Mostre os resultados usando f-string e format().

def resultado_alunos(alunos):
    resultados = {}
    for nome, nota in alunos.items():
        if nota >= 7:
            resultados[nome] = "Aprovado(a)"
        else:
            resultados[nome] = "Reprovado(a)"
    return resultados

# Exemplo de uso
alunos = {
    "Ana": 8,
    "João": 6,
    "Pedro": 9,
    "Maria": 5
}

resultados = resultado_alunos(alunos)
for nome, situacao in resultados.items():
```

```
print(f"O(a) aluno(a) {nome} está {situacao}.")
print("{} está {}".format(nome, situacao))
```

12. Crie uma função chamada `conta_vogais(texto)` que receba uma string e retorne a quantidade de vogais presentes.

- Mostre os resultados usando f-string e `format()`.

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python

# 12. Crie uma função chamada conta_vogais(texto) que receba uma string e retorne
# a quantidade de vogais presentes.
# ● Mostre os resultados usando f-string e format().

def conta_vogais(texto):
    vogais = "aáâãäåæçèéêëíîóôûúÄÅÄÅÆÉÊËÏÍÎÏÓÔÛÜ"
    contador = sum(1 for char in texto if char in vogais)
    return contador

texto = input("Digite uma string: ")
quantidade_vogais = conta_vogais(texto)
print(f"A quantidade de vogais na string é: {quantidade_vogais}")
print("{} é a quantidade de vogais na string".format(quantidade_vogais))
```

13. Crie uma função chamada `operacoes(a, b)` que receba dois números float e retorne uma tupla contendo: soma, subtração, multiplicação e divisão.

- Mostre os resultados usando f-string e `format()`.

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python

# 13. Crie uma função chamada operacoes(a, b) que receba dois números float
# e retorne uma tupla contendo: soma, subtração, multiplicação e divisão.
# ● Mostre os resultados usando f-string e format().

def operacoes(a, b):
    soma = a + b
    subtracao = a - b
```

```
multiplicacao = a * b
divisao = a / b if b != 0 else "Divisão por zero não é permitida"
return (soma, subtracao, multiplicacao, divisao)

# Solicita ao usuário dois números float
try:
    num1 = float(input("Digite o primeiro número (float): "))
    num2 = float(input("Digite o segundo número (float): "))

    # Chama a função e obtém os resultados
    resultados = operacoes(num1, num2)

    # Mostra os resultados usando f-string e format()
    print(f"Soma: {resultados[0]:.2f}")
    print("Subtração: {:.2f}".format(resultados[1]))
    print(f"Multiplicação: {resultados[2]:.2f}")
    print("Divisão: {}".format(resultados[3]))
except ValueError:
    print("Entrada inválida. Por favor ao rodar programa, digite números float válidos.")
```

14. Crie uma função chamada coletar_nomes() que peça nomes ao usuário até que ele digite "sair". A função deve retornar a lista de nomes digitados.

- Mostre os resultados usando f-string e format().

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python

# 14. Crie uma função chamada coletar_nomes() que peça nomes ao usuário até que
# ele digite "sair". A função deve retornar a lista de nomes digitados.
# • Mostre os resultados usando f-string e format().

def coletar_nomes():
    nomes = []
    while True:
        nome = input("Digite um nome (ou 'sair' para encerrar): ")
        if nome.lower() == 'sair':
            break
        nomes.append(nome)
    return nomes

nomes_coletados = coletar_nomes()
print(f"Nomes coletados (usando f-string): {nomes_coletados}")
print("Nomes coletados (usando format()): {}".format(nomes_coletados))
```

15. Crie uma função chamada `conta_maiusculas(texto)` que receba uma string e retorne quantas letras maiúsculas existem nela.

- Mostre os resultados usando f-string e `format()`.

RESPOSTA: Segue linhas de código do programa resposta.

```
# Bolsa Futuro Digital
# São José de Ribamar, 31 de agosto de 2025.
# Discente: Renato da Fonseca Oliveira

# Atividade Avaliativa - Python

# 15. Crie uma função chamada conta_maiusculas(texto) que receba uma string
# e retorne quantas letras maiúsculas existem nela.
# • Mostre os resultados usando f-string e format().

def conta_maiusculas(texto):
    contador = 0
    for letra in texto:
        if letra.isupper():
            contador += 1
    return contador

texto = input("Digite um texto: ")
resultado = conta_maiusculas(texto)
print(f"Número de letras maiúsculas (usando f-string): {resultado}")
print("Número de letras maiúsculas (usando format()): {}".format(resultado))
```