

## Trabalho Avaliativo - GA

Aluno(a): Cintia da Silva Bulcão

RA: 145107

Data de Entrega: 15/06/25

01) Crie uma rotina com a definição de três funções, uma que calcule o produto escalar em 3D, uma que calcule o produto vetorial e uma que calcule o produto misto. Crie um menu para que o usuário escolha o que quer calcular. Se escolher o produto escalar ou o vetorial, peça que ele insira os dois vetores 3D necessários, se ele escolher o produto misto, peça que ele insira os três vetores 3D. Não há necessidade de criar ilustrações.

```
In [3]: def produto_escalar(v1: list, v2: list) -> float:
        return sum([v1[i] * v2[i] for i in range(3)])

def produto_vetorial(v1: list, v2: list) -> list:
    x = v1[1] * v2[2] - v1[2] * v2[1]
    y = -(v1[0] * v2[2] - v1[2] * v2[0])
    z = v1[0] * v2[1] - v1[1] * v2[0]
    return [x, y, z]

def produto_misto(v1: list, v2: list, v3: list) -> float:
    from sage.all import matrix, RR
    mat = matrix(RR, 3, 3, v1 + v2 + v3)
    return round(mat.determinant(), 2)

def solicitar_vetor(nome: str) -> list:
    while True:
        entrada = input(f'Digite as 3 coordenadas do vetor {nome} separadas por vírgulas: ')
        vetor = list(map(float, entrada.strip().split(',')))
        if len(vetor) == 3:
            return vetor
```

```
        print(f'O vetor informado possui {len(vetor)} coordenadas. Tente novamente.')

print('\n--- Calculadora de Produtos entre Vetores 3D ---\n')

opcao = 0
while opcao != 4:
    print('Escolha a operação desejada:')
    print('1 - Produto Escalar')
    print('2 - Produto Vetorial')
    print('3 - Produto Misto')
    print('4 - Sair')

    try:
        opcao = int(input('Digite sua opção: '))
    except ValueError:
        print('Por favor, insira um número válido.\n')
        continue

    if opcao == 1:
        print('\n[Produto Escalar Selecionado]')
        a = solicitar_vetor('A')
        b = solicitar_vetor('B')
        resultado = produto_escalar(a, b)
        print(f'Resultado: o produto escalar entre {a} e {b} é {resultado}\n')

    elif opcao == 2:
        print('\n[Produto Vetorial Selecionado]')
        a = solicitar_vetor('A')
        b = solicitar_vetor('B')
        resultado = produto_vetorial(a, b)
        print(f'Resultado: o produto vetorial entre {a} e {b} é {resultado}\n')

    elif opcao == 3:
        print('\n[Produto Misto Selecionado]')
        a = solicitar_vetor('A')
        b = solicitar_vetor('B')
        c = solicitar_vetor('C')
        resultado = produto_misto(a, b, c)
        print(f'Resultado: o produto misto entre {a}, {b} e {c} é {resultado}\n')

    elif opcao == 4:
        print('\nSaindo do programa. Até logo!\n')
```

```
else:  
    print('\nOpção inválida. Por favor, escolha uma das opções disponíveis.\n')
```

Out[3]:

--- Calculadora de Produtos entre Vetores 3D ---

Escolha a operação desejada:

- 1 - Produto Escalar
- 2 - Produto Vetorial
- 3 - Produto Misto
- 4 - Sair

Digite sua opção:

[Produto Escalar Selecionado]

Digite as 3 coordenadas do vetor A separadas por vírgulas:

Digite as 3 coordenadas do vetor B separadas por vírgulas:

Digite sua opção:

Resultado: o produto escalar entre [4.0, 2.0, 3.0] e [5.0, 8.0, 7.0] é 57.0

Escolha a operação desejada:

- 1 - Produto Escalar
- 2 - Produto Vetorial
- 3 - Produto Misto
- 4 - Sair

Saindo do programa. Até logo!

02)Crie uma rotina que receba as coordenadas de três vetores u, v e w do espaço e retorne:

. (i) a área do paralelogramo determinado por u e v e sua ilustração.

. (ii) o volume do paralelepípedo determinado por u, v e w e sua ilustração.

. (iii) a altura do paralelepípedo relativa à base determinada por u e v (sem ilustração).

```
In [4]: import sage.all as sg

def vetorial(a: list, b: list) -> list:
    return [
        a[1]*b[2] - a[2]*b[1],
        -(a[0]*b[2] - a[2]*b[0]),
        a[0]*b[1] - a[1]*b[0]
    ]

def comprimento(v: list) -> float:
    return sg.sqrt(sum([v[i]**2 for i in range(3)]))

def prod_misto(a: list, b: list, c: list) -> float:
    mat = sg.matrix(sg.RR, 3, 3, a + b + c)
    return round(mat.determinant(), 2)

def ler_vetor(rótulo: str) -> list:
    while True:
        entrada = input(f'Digite as 3 coordenadas do vetor {rótulo}, separadas por vírgulas: ')
        dados = list(map(float, entrada.strip().split(',')))
        if len(dados) == 3:
            return dados
        print(f'O vetor {rótulo} possui {len(dados)} componentes. Insira exatamente 3 valores.\n')

print('\nAnálise tridimensional: vetores A, B e C')
print('Objetivo: calcular a área da base (A x B), o volume do paralelepípedo (A, B, C) e a altura relativa.\n')

while True:
    A = ler_vetor('A')
    B = ler_vetor('B')
    C = ler_vetor('C')

    if prod_misto(A, B, C) == 0:
        print('\nOs vetores estão contidos em um mesmo plano. Escolha vetores não coplanares.\n')
        continue
    break

origem = [0, 0, 0]
seta_A = sg.arrow(origem, A, arrowsize=2, color='darkred')
seta_B = sg.arrow(origem, B, arrowsize=2, color='darkblue')
seta_C = sg.arrow(origem, C, arrowsize=2, color='darkgreen')
```

```

sg.show(seta_A + seta_B + seta_C, axes=True)

vet_cruzado = vetorial(A, B)
area_fund = comprimento(vet_cruzado)
print(f'\nÁrea da base (paralelogramo de A e B): {area_fund:.2f} unidades².')

soma_AB = [A[i] + B[i] for i in range(3)]
base = sg.polygon([origem, A, soma_AB, B], color='lightgray')
sg.show(seta_A + seta_B + base, axes=True)

volume_total = abs(prod_misto(A, B, C))
print(f'Volume do paralelepípedo formado: {volume_total:.2f} unidades³.')
print()
print()
altura_relativa = volume_total / area_fund
print(f'Altura relativa ao plano base (A x B): {altura_relativa:.2f} unidades.')

```

Out[4]:

Análise tridimensional: vetores A, B e C

Objetivo: calcular a área da base (A x B), o volume do paralelepípedo (A, B, C) e a altura relativa.

Digite as 3 coordenadas do vetor A, separadas por vírgulas:

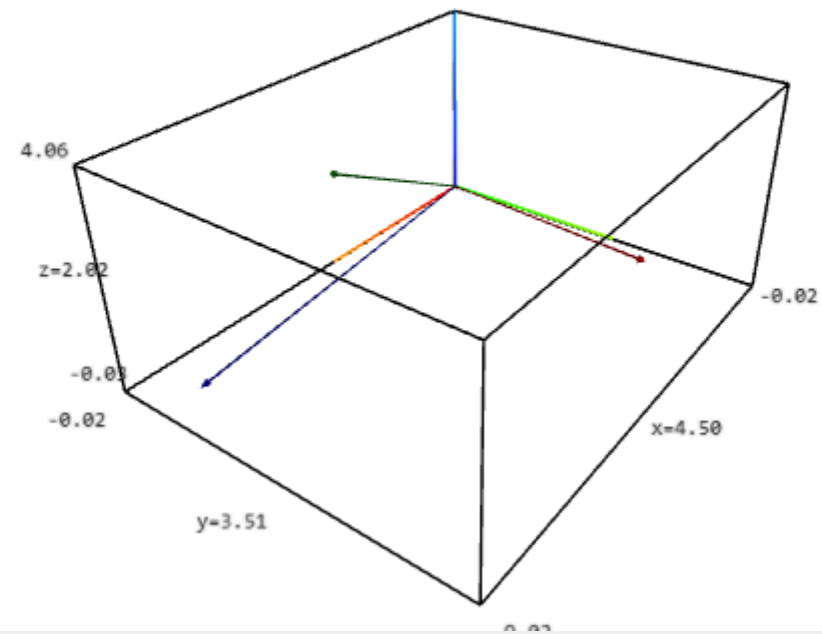
Digite as 3 coordenadas do vetor B, separadas por vírgulas:

Digite as 3 coordenadas do vetor C, separadas por vírgulas:

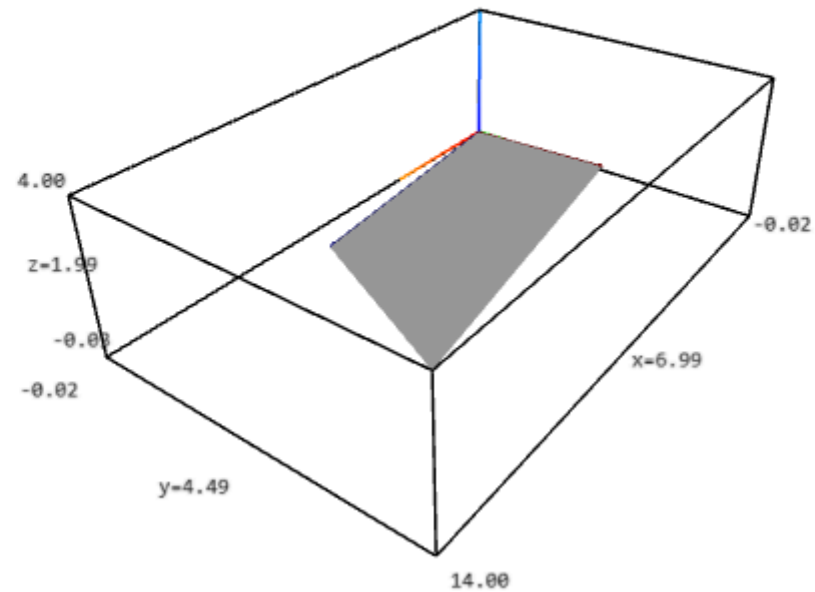
```

/ext/sage/10.6/local/var/lib/sage/venv-python3.12.5/lib/python3.12/site-packages/scikits/__init__.py:1: DeprecationWarning:
pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
__import__("pkg_resources").declare_namespace(__name__)
/ext/sage/10.6/local/var/lib/sage/venv-python3.12.5/lib/python3.12/site-packages/scikits/__init__.py:1: DeprecationWarning:
Deprecated call to `pkg_resources.declare_namespace('scikits')`.
Implementing implicit namespace packages (as specified in PEP 420) is preferred to `pkg_resources.declare_namespace`. See
https://setuptools.pypa.io/en/latest/references/keywords.html#keyword-namespace-packages
__import__("pkg_resources").declare_namespace(__name__)

```



Área da base (paralelogramo de A e B): 57.39 unidades<sup>2</sup>.



Volume do paralelepípedo formado: 139.00 unidades<sup>3</sup>.

Altura relativa ao plano base (A x B): 2.42 unidades.

03) Crie um programa no qual o usuário consiga inserir quantos vetores de 3 dimensões ele desejar. Impeça que ele insira vetores de dimensões diferentes de 3. Seu programa deve procurar conjuntos de três vetores coplanares na lista fornecida e imprimir os resultados.

```
def calcular_produto_misto(u, v, w):
    return u[0] * v[1] * w[2] + u[1] * v[2] * w[0] + u[2] * v[0] * w[1] - \
        (u[2] * v[1] * w[0] + u[0] * v[2] * w[1] + u[1] * v[0] * w[2])
```

```
quantidade_vetores = int(input('Quantos vetores você deseja inserir? '))
vetores = []
```

```
for i in range(quantidade_vetores):
    print(f'\nInsira o vetor {i + 1} no formato "x, y, z":')

    while True:
        entrada = input().split(',')
        vetor = list(map(float, entrada))

        if len(vetor) == 3:
            vetores.append(vetor)
            break
        else:
            print(f'\nO vetor inserido possui {len(vetor)} coordenadas!')
            print('Por favor, insira um vetor com exatamente 3 coordenadas (x, y, z).')

print('\nOs vetores selecionados foram:')
for vetor in vetores:
    print(vetor)

vetores_coplanares = []

for i in range(quantidade_vetores):
    u = vetores[i]
    for j in range(i + 1, quantidade_vetores):
        v = vetores[j]
        for k in range(j + 1, quantidade_vetores):
            w = vetores[k]
            if calcular_produto_misto(u, v, w) == 0:
                vetores_coplanares.extend([u, v, w])

vetores_coplanares_unicos = []
for vetor in vetores_coplanares:
    if vetor not in vetores_coplanares_unicos:
        vetores_coplanares_unicos.append(vetor)

print('\nDos vetores selecionados, os vetores coplanares são:\n', vetores_coplanares_unicos)
for vetor in vetores_coplanares_unicos:
    print(vetor)
```



Out[5]: Quantos vetores você deseja inserir?

Insira o vetor 1 no formato "x, y, z":

Insira o vetor 2 no formato "x, y, z":

Insira o vetor 3 no formato "x, y, z":

Os vetores selecionados foram:

[1.0, 2.0, 3.0]

[7.0, 8.0, 9.0]

[15.0, 16.0, 17.0]

Dos vetores selecionados, os vetores coplanares são:

[[1.0, 2.0, 3.0], [7.0, 8.0, 9.0], [15.0, 16.0, 17.0]]

[1.0, 2.0, 3.0]

[7.0, 8.0, 9.0]

[15.0, 16.0, 17.0]