

Aluna: Cintia da Silva Bulcão Ra: 145107

## Code 1)

### 0.1.1 Exercício 1

Crie uma rotina que receba do usuário as coordenadas dos vértices de um triângulo  $ABC$ . Seu programa deve imprimir na tela as coordenadas dos vértices do triângulo e imprimir as coordenadas dos pontos médios dos lados desse triângulo. Ele deve esclarecer que o ponto médio impresso é do lado AB, ou AC ou BC. Ao final, construa ilustração mostrando os vértices, o triângulo, e os pontos médios

In [4]:

```
teste = 'y'

def ponto_medio(a: float, b:float) -> float:
    p_medio = [None]*2
    p_medio[0] = (a[0] + b[0]) / 2
    p_medio[1] = (a[1] + b[1]) / 2
    return p_medio

while teste == 'y':

    A = [None]*2
    print('Informe as coordenadas do ponto A, separadas por vírgula.\nEx: x , y')
    A = (list(map(float,input().split(','))))

    print('Ponto A: ',A)

    B = [None]*2
    print('Informe as coordenadas do ponto B, separadas por vírgula.\nEx: x , y')
    B = (list(map(float,input().split(','))))

    print('Ponto B: ',B)

    C = [None]*2
    print('Informe as coordenadas do ponto C, separadas por vírgula.\nEx: x , y')
    C = (list(map(float,input().split(','))))

    print('Ponto C: ',C)

    matriz_pontos = matrix(RR, 3,3, [A[0], A[1] , 1, B[0], B[1], 1, C[0],
C[1], 1])

    if matriz_pontos.determinant() == 0:
        print('\nOs pontos A, B e C estão alinhados.Informe três pontos
```

```

diferentes\n')
    teste == 'y'
else:
    print('Os pontos A, B e C formam um triângulo com as seguintes
propriedades: ', \
        '\nPonto médio entre A e B=', ponto_medio(A,B), \
        '\nPonto médio entre A e C=', ponto_medio(A,C), \
        '\nPonto médio entre B e C=', ponto_medio(B,C))
    print()

    pontoA=point(A, color = 'black')
    pontoB=point(B, color = 'black')
    pontoC=point(C, color = 'black')
    retaAB=line([A,B], color = 'black')
    retaAC=line([A,C], color = 'black')
    retaBC=line([B,C], color = 'black')
    retapm_AB=point(ponto_medio(A,B), color = 'black')
    retapm_AC=point(ponto_medio(A,C), color = 'black')
    retapm_BC=point(ponto_medio(B,C), color = 'black')

show(pontoA+pontoB+pontoC+retaAB+retaAC+retaBC+retaAB+retaAC+retaBC,figsize=4
,aspect_ratio=1)

    teste = str(input('Deseja testar outros pontos? (y/n)\n'))
    if teste == 'n':
        print()
        print ('Fim.\n')

```

Out[4]: Informe as coordenadas do ponto A, separadas por vírgula.

Ex: x , y

1,2

Ponto A: [1.0, 2.0]

Informe as coordenadas do ponto B, separadas por vírgula.

Ex: x , y

2,3

Ponto B: [2.0, 3.0]

Informe as coordenadas do ponto C, separadas por vírgula.

Ex: x , y

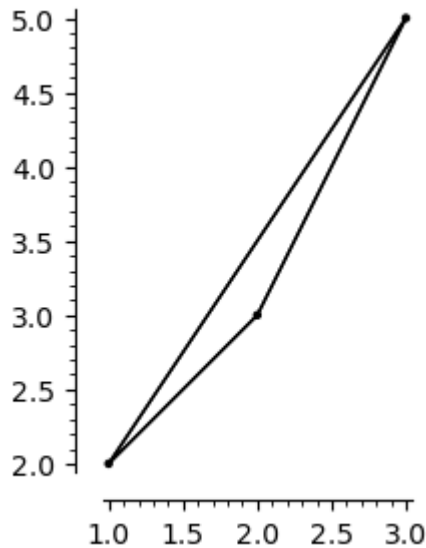
Ponto C: [3.0, 5.0]

Os pontos A, B e C formam um triângulo com as seguintes propriedades:

Ponto médio entre A e B= [1.5, 2.5]

Ponto médio entre A e C= [2.0, 3.5]

Ponto médio entre B e C= [2.5, 4.0]



Deseja testar outros pontos? (y/n)

n

Fim.

## Code 2)

### 0.1.2 Exercício 02

Crie um programa no qual o usuário defina em que dimensão quer trabalhar. Após tal escolha, seu programa deve mostrar um menu no qual o usuário escolha se quer calcular  $u + v$ ,  $u - v$  ou  $\square * v$ . Feita a escolha, o usuário deve inserir, separadas por vírgulas, as  $n$  coordenadas dos vetores necessários e/ou o escalar  $\square$  (trabalhe com "float"). O programa deve retornar os vetores e/ou o escalar inserido e o vetor resultante da operação escolhida. Ao final, seu programa deve indagar se o usuário quer realizar outra operação ou encerrar o programa

In [3]:

```
teste = 'y'

while teste == 'y':
    print('---OPERAÇÕES COM VETORES---')
    print ('0 - Calcule a soma de dois vetores(A + B).')
    print ('1 - Calcule a diferença de dois vetores(A - B).')
    print ('2 - Calcule o multiplo de um vetor(v * A ).')
    print()

    opcao = input('Digite o número da opção desejada (0, 1 ou 2): ')

    if opcao == '0':
        n = int(input('\nQuantas dimensões têm os vetores? '))
        print('\nInsira os componentes do vetor A:')
        A = [None] * n
        A = (list(map(float,input().split(','))))
        print('\nInsira os componentes do vetor B separados por virgula.\nEx:
x, y, z')
        B = [None] * n
```

```

B = (list(map(float,input().split(','))))
soma = (A + B)
print(f'\nResultado da soma (A + B): {soma}')

elif opcao == '1':
    n = int(input('\nQuantas dimensões têm os vetores? '))
    print('\nInsira os componentes do vetor')
    A = [None] * n
    A = (list(map(float,input().split(','))))
    print('\nInsira os componentes do vetor B:')
    B = [None] * n
    B = (list(map(float,input().split(','))))
    diferenca = (A - B)
    print(f'\nResultado da diferença (A - B): {diferenca}')

elif opcao == '2':
    n = int(input('\nQuantas dimensões tem o vetor? '))
    print('\nInsira os componentes do vetor B separados por vírgula.\nEx:
x, y, z')
    A = [None] * n
    A = (list(map(float,input().split(','))))
    v = int(input('\nInsira o valor do escalar v: '))
    multiplo = (v * A)
    if len(multiplo) > len(A):
        multiplo = multiplo[:len(A)]

    print(f'\nResultado do múltiplo (v * A): {multiplo}')

else:
    print('\nOpção inválida! Digite 0, 1 ou 2.')

teste = input('\nDeseja realizar outra operação? (y/n): ')
if teste == 'n':
    print('\nPrograma encerrado. Obrigado!')
else:
    print ('\nVocê inseriu um número não presente nas alternativas! Por
favor digite um número válido!\n')
    print()
    teste = 'y'

```

Out[3]: ---OPERAÇÕES COM VETRES---

- 0 - Calcule a soma de dois vetores(A + B).
- 1 - Calcule a diferença de dois vetores(A - B).
- 2 - Calcule o multiplo de um vetor(v \* A ).

Digite o número da opção desejada (0, 1 ou 2):

2

Quantas dimensões tem o vetor?

2

Insira os componentes do vetor B separados por vírgula.

Ex: x, y, z

1,2

Insira o valor do escalar v:

Resultado do múltiplo ( $v * A$ ): [1.0, 2.0]

Deseja realizar outra operação? (y/n):

Programa encerrado. Obrigado!

### Code 3)

#### 0.1.3 Exercício 3

Crie um programa que receba do usuário um ponto  $A = (x, y, z)$ . A rotina deve indagar que projeção o usuário deseja obter: se quer a projeção do vetor  $OA$  no eixo  $Ox$ , no eixo  $Oy$  ou no eixo  $Oz$ . O retorno em tela deve exibir as coordenadas do ponto inserido e as coordenadas da projeção com sua respectiva descrição. Ao final, deve mostrar uma ilustração com o vetor  $OA$  e o vetor projeção escolhido pelo usuário. Seu programa deve indagar se o usuário quer obter projeções de outro vetor ou quer encerrar o programa

In [6]:

```
teste = 'y'

while teste == 'y':
    A=[None]*3
    print('Informe as coordenadas do ponto A, separadas por vírgula.\nEx: x , y, z')
    A=(list(map(float,input().split(','))))
    print()
    print('Ponto A: ',A)
    print()

    fig_OA = line([(0,0,0),A],color='red')
    show(fig_OA,figsize=4,aspect_ratio=1)

    print ('Selecione o eixo para projeção:')
    print ('0 - Projeter o Vetor no eixo X.')
    print ('1 - Projeter o Vetor no eixo Y.')
    print ('2 - Projeter o Vetor no eixo Z.')

    teste = str(input('Digite sua opção:\n'))
    if teste == '0':
        X =[ None] * 3
        X[0] = 1
        X[1] = 0
        X[2] = 0
        aux = 0
        for i in range (0,3):
            aux = aux + A[i] * X[i]

        soma=0
        for i in range (0,3):
```

```

        soma = soma + X[i] ^ 2

    alfa = aux / soma

    vetor = [None]*3
    for i in range (0,3):
        vetor[i] = X[i] * alfa

    print()
    print('O vetor A inserido ao ser projetado no eixo X gera o seguinte
vetor:\n',vetor)
    print()

    fig_OA=line([(0,0,0),A],color='red')
    fig_OX=line([(0,0,0),X],color='blue')
    fig_AX=line([(0,0,0),vetor],color='green')
    show(fig_OA + fig_OX + fig_AX,figsize=4,aspect_ratio=1)

    print()
    teste = str(input('Deseja fazer mais operações? (y/n)\n'))
    if teste == 'n':
        print()
        print ('Fim.\n')

elif teste == '1':
    Y =[ None] * 3
    Y[0] = 0
    Y[1] = 1
    Y[2] = 0
    aux = 0
    for i in range (0,3):
        aux = aux + A[i] * Y[i]

    soma=0
    for i in range (0,3):
        soma = soma + Y[i] ^ 2

    alfa = aux / soma

    vet = [None]*3
    for i in range (0,3):
        vetor[i] = Y[i] * alfa

    print()
    print('O vetor A inserido ao ser projetado no eixo Y gera o seguinte
vetor:\n',vetor)
    print()

    fig_OA=line([(0,0,0),A],color='red')
    fig_OY=line([(0,0,0),Y],color='blue')
    fig_AY=line([(0,0,0),vetor],color='green')
    show(fig_OA + fig_OY + fig_AY,figsize=4,aspect_ratio=1)

    print()
    teste = str(input('Deseja fazer mais operações? (y/n)\n'))
    if teste == 'n':
        print()
        print ('Fim.\n')

elif teste == '2':
    Z =[ None] * 3

```

```

Z[0] = 0
Z[1] = 0
Z[2] = 1
aux = 0
for i in range (0,3):
    aux = aux + A[i] * Z[i]

soma=0
for i in range (0,3):
    soma = soma + Z[i] ^ 2

alfa = aux / soma

vet = [None]*3
for i in range (0,3):
    vet[i] = Z[i] * alfa

print()
print('O vetor A inserido ao ser projetado no eixo Z gera o seguinte
vetor:\n',vet)
print()

fig_OA=line([(0,0,0),A],color='red')
fig_OZ=line([(0,0,0),Z],color='blue')
fig_AZ=line([(0,0,0),vetor],color='green')
show(fig_OA + fig_OZ + fig_AZ,figsize=4,aspect_ratio=1)

print()
teste = str(input('Deseja fazer mais operações? (y/n)\n'))
if teste == 'n':
    print()
    print ('Fim.\n')

else:
    print ('\nErro! Por favor digite um número válido!\n')
    teste = 'y'

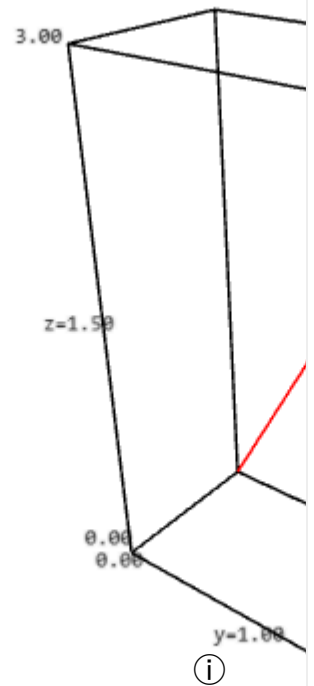
```

Out[6]: Informe as coordenadas do ponto A, separadas por vírgula.

Ex: x , y, z

1,2,3

Ponto A: [1.0, 2.0, 3.0]



Selecione o eixo para projeção:

0 - Projetar o Vetor no eixo X.

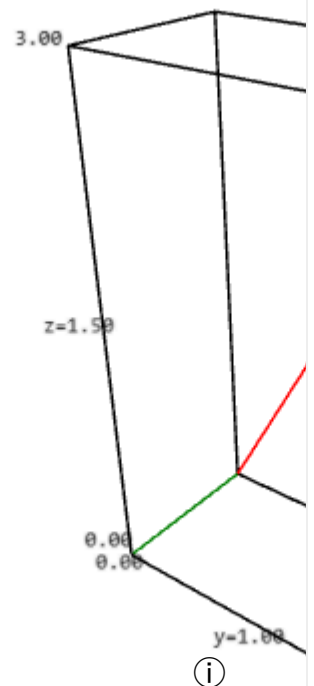
1 - Projetar o Vetor no eixo Y.

2 - Projetar o Vetor no eixo Z.

Digite sua opção:

O vetor A inserido ao ser projetado no eixo X gera o seguinte vetor:

[1.0, 0.0, 0.0]





Deseja fazer mais operações? (y/n)

Fim.

In [0]: