



```

xA=3
yA=-1
xB=-4
yB=-2
if xA==xB:
    print('A reta que une os dois pontos é vertical: x=',xA)
else:
    m=(yB-yA)/(xB-xA)
    b= yA-m*xA
    print('Declive da reta: ',m)
    print('Ordenada na origem: ',b)
    print('y=',m,'x+',b)
print('Uma equação vetorial da reta é:
(x,y)=(',xA,', ',yA,',)+k(',xB-xA,', ',yB-yA,',),kER')

```

Editar Python na TI-nspire CX II-T

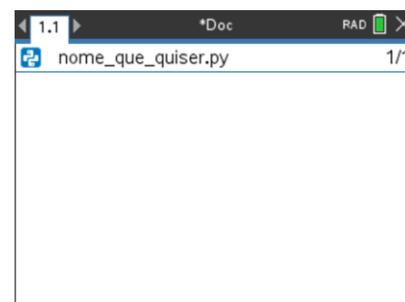
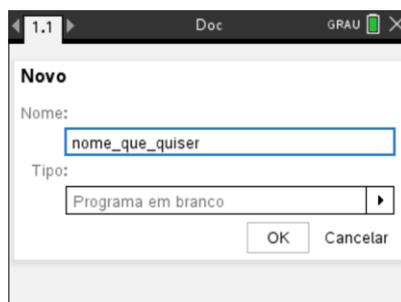
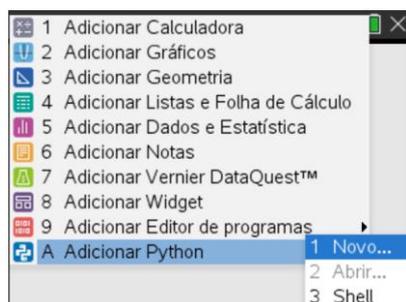
Ligue a sua calculadora e crie um novo documento.

Escolha uma página de *Python*:

A Adicionar Python → **1** Novo.

Coloque um nome à sua escolha, de seguida, prime em **OK**.

Abre-se uma página vazia, que é o editor de *Python* da calculadora/tecnologia TI-Nspire CX II-T, onde deve escrever o código.



1. A partir das coordenadas de dois pontos, como determinar o declive, a ordenada na origem, a equação reduzida e uma equação vetorial da reta que passa por esses dois pontos?

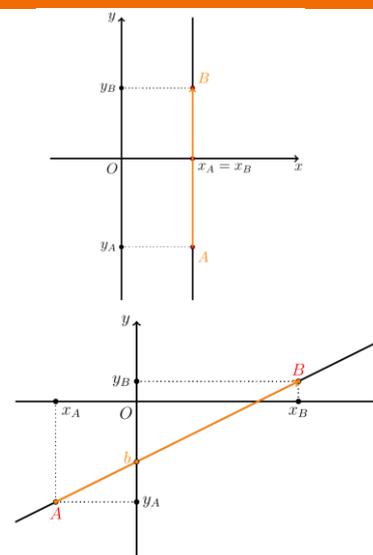
Pretende-se um programa que, depois de executado apresente, relacionando as coordenadas de dois pontos, descreva a reta que os contém, fornecendo informações acerca do declive, da ordenada na origem e também da equação vetorial.

Designando as coordenadas desses pontos, A e B , por (x_A, y_A) e (x_B, y_B) , respetivamente, há dois casos a considerar:

- $x_A = x_B$: A e B pertencem a uma reta vertical de equação $x = x_A$.
- $x_A \neq x_B$: A e B , pertencem a uma reta não vertical de equação $y = mx + b$, onde m é o declive da reta e b é a ordenada de origem.

Em qualquer dos casos referidos, a reta tem a respetiva a equação vetorial

$$(x, y) = A + k \overrightarrow{AB} = (x_A, y_A) + k(x_B - x_A, y_B - y_A), \quad k \in \mathbb{R}$$



- I. Inicialmente, é preciso, no programa, indicar as coordenadas dos dois pontos. Por isso, insira no editor, utilizando o teclado, as seguintes linhas código, as quais determinam a atribuição de valores às variáveis x_A , y_A , x_B e y_B :

```
xA = 3
yA = -1
xB = 3
yB = -2
```

```
*AEp43
1/5
xA=3
yA=-1
xB=3
yB=-2
```

É agora necessário que o programa siga os caminhos adequados aos casos referidos, ou seja, se os dados correspondem a pontos em reta vertical ou não. Repare, então, que pode escrever-se, em linguagem natural:

Se as abscissas dos dois pontos são iguais:

Os pontos estão sobre uma reta vertical de equação $x = x_A$ (ou $x = x_B$).

Caso contrário:

Os pontos estão sobre uma reta não vertical de equação do tipo $y = mx + b$, onde m é o declive da reta e b a ordenada na origem.

Onde:

- o declive é o quociente $m = \frac{y_A - y_B}{x_A - x_B}$ e
- a ordenada na origem é $b = y - mx$.

- II. Para passar agora a linguagem *Python*, é necessário utilizar uma estrutura condicional, que pode escrever com o teclado ou obter no menu.

menu → **4 Planos integrados** → **2 Controlo** → **3 if..elif..else..**

A saída da informação pretendida deverá ser feita com a função **print()**, a qual pode ser escrita diretamente no editor, com o teclado, ou obtida no menu:

menu **4 Planos integrados** → **5 Tipo** → **6 I/O** → **1 print()**

```
Ações
Executar
1 if..
2 if..else..
3 if..elif..else..
4 for index in range(size):
5 for index in range(start, stop):
6 for index in range(start, stop, step):
7 for index in list:
8 while..
9 elif:
A else:
```

Na sequência do que foi referido, escrevam-se agora as seguintes linhas de código:

```
if xA==xB:
    print("A reta que une os dois pontos é vertical: x=",xA)
else:
    m=(yB-yA)/(xB-xA)
    b=yA-m*xA
    print("Declive da reta:",m)
    print("Ordenada na origem:", b)
    print("y=",m, "x+",b)
```

```
*AEp43
2/13
yA=-1
xB=3
yB=-2
if xA==xB:
    print("A reta que une os dois pontos é vertical: x=",xA)
else:
    m=(yB-yA)/(xB-xA)
    b=yA-m*xA
    print("Declive da reta: ",m)
    print("Ordenada na origem: ",b)
    print("y=",m,"x+",b)
```

- III. Por fim, quer a reta que passe pelos pontos seja vertical ou não, pretende-se que o programa imprima uma equação vetorial da reta, considerando as coordenadas do um dos seus pontos, $A(x_A, y_A)$, adicionado do produto de um parâmetro real k pelo vetor diretor de coordenadas $(x_B - x_A, y_B - y_A)$.

Utilizando novamente função **print()**, insira no programa a seguinte linha:

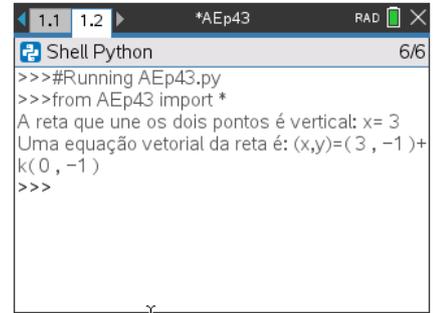
```
print("Uma equação vetorial da reta é: (x,y)=(", xA, ",", yA, ") + k(", xB-xA, ",", yB-yA, ")", k, " E R")
```

```
print("y=",m,"x+",b)
print("Uma equação vetorial da reta é: (x,y)=(",xA,",",yA,")+k(",xB-xA,",",yB-yA,")")
```

IV. Escrito o programa, falta executá-lo.

Pode utilizar-se uma instrução do menu ( **2** **1**), mas é claramente mais simples utilizar um atalho, uma combinação de teclas (**ctrl** + **R**).

O resultado aparece numa nova página destinada a mostrar o resultado da execução do programa, **Shell Python**, na qual também se podem fazer operações e programas, mas que não permanecerão gravados após o fecho da aplicação.



```
Shell Python 6/6
>>>#Running AEp43.py
>>>from AEp43 import *
A reta que une os dois pontos é vertical: x= 3
Uma equação vetorial da reta é: (x,y)=( 3 , -1)+
k( 0 , -1 )
>>>
```

Para voltar ao editor de *Python*, onde poderá alterar os dados de entrada, por exemplo, há mais do que um procedimento à escolha, baseados no botão do touchpad. Pode fazer deslocar o cursor com o dedo até o sobrepor ao retângulo com a designação da página, **1.1**, neste caso, e premir o touchpad na parte central (). Pode também utilizar os botões laterais do touchpad após premir a tecla **ctrl**. Neste caso, ao premir o botão lateral esquerdo, vai para a página anterior, a do editor. Pode voltar à página de *Shell Python* utilizando o mesmo tipo de procedimento.

Na parte superior do ecrã apenas se pode observar a designação e 3 páginas consecutivas, pelo que se o documento tiver mais páginas terá de conjugar os dois procedimentos referidos ou simplesmente o que recorre às teclas laterais do touchpad.



Algumas ideias sobre programação, relacionadas com o contexto

