

Unidade 4: Utilização da biblioteca TI PlotLib

Lição 1: Configurar uma representação


Nesta primeira lição da Unidade 4 vamos aprender como escrever e usar uma instrução para efetuar representações gráficas em Python. Iremos, também, aprender a como traçar um gráfico e configurar a representação.

Objetivos:

- Explorar o módulo **TI PlotLib**.
- Representar um ponto e um segmento de reta.
- Configurar uma representação gráfica

Etapa 1: A livraria ou módulo TI PlotLib

Para efetuar uma representação gráfica recorrendo a um programa teremos que fazer com que o programa “reconheça” as instruções de representação gráfica. Assim, teremos que, antes do uso dessas instruções, “embutir” as funções de representação gráfica de uma biblioteca **TI PlotLib**.


Abra um novo documento TI-Nspire com um novo programa da aplicação TI-Python e designe-o por U4L1, colocando como primeira instrução a inclusão do módulo **TI PLoTLib** (tecla , escolher a opção **7: TI PlotLib** e depois **1: import ti_plotlib as plt**.

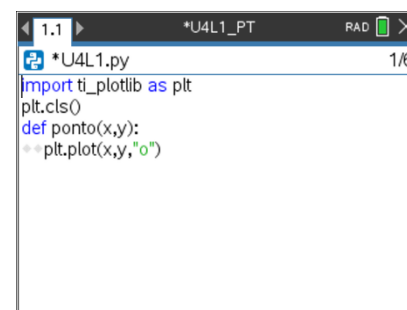
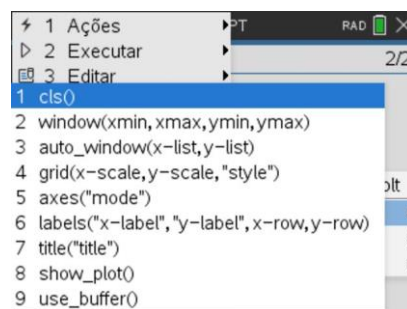
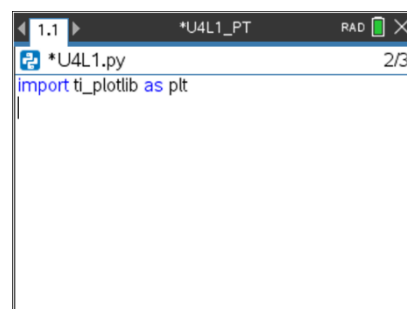
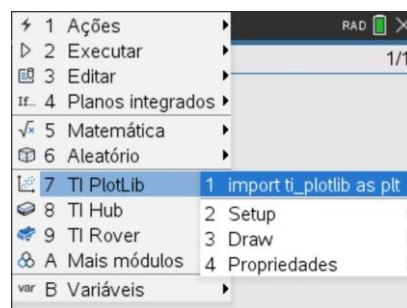
Desta forma, a partir desta instrução, o programa reconhecerá as habituais instruções usada na representação gráfica.

Numa primeira fase, vamos escrever um programa que permita representar um ponto conhecidas as suas coordenadas. De seguida, alteraremos o programa de modo a localizar o ponto no referencial e a alterar a sua cor.

Para terminar esta primeira lição, colocaremos visível o nome de cada eixo e daremos um título à representação gráfica.

Defina uma função que tenha como argumentos as coordenadas de um ponto e, em seguida, efetue a representação gráfica desse ponto.

- Começemos por limpar do ecrã a representação gráfica usando a função **plt.cls()**, que encontrará no módulo **TI PlotLib** (tecla , opção **7: TI PlotLib**, depois opção **2:Configurar** e por fim **1:cls()**).
- De seguida defina uma função, designe-a por **ponto()**, recorrendo ao menu **4: Planos integrados** e à opção **1: Funções**. Coloque como argumentos as variáveis **x** e **y**, coordenadas no plano.
- Para representar um ponto, selecione no submenu **3: Desenhar** a opção **6: plot(x,y,"mark")**, situado no menu do módulo **TI PlotLib**.



- Escolha, agora, a marca que pretende para representar o ponto.

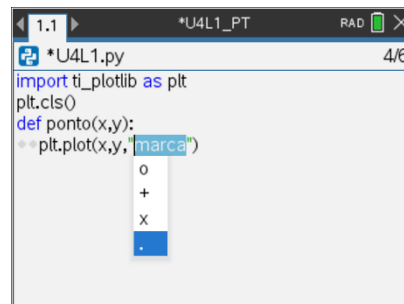
SUGESTÃO:

A marca utilizada para a representação do ponto deve ser escolhida tendo em atenção o número de pontos e outros elementos a representar na mesma janela, sendo aconselhado, pelo menos no caso de serem muitos pontos, a optar-se pela forma de pixel (.). Após a escolha da forma da marca, para a alterar terá que a inserir de novo através do menu respetivo.

- Termine o programa efetuando a representação gráfica, usando a função **show_plot()** (opção 3: **shown_plot()** do submenu 3: **Desenhar** do módulo **TI PlotLib**).
- Execute o programa, clicando simultaneamente nas teclas **ctrl** e **R**, e de seguida, no interpretador, clique na tecla **var** e selecione a função **ponto()**. Poderá, também, escrever o nome da função que o interpretador reconhecerá.
- Coloca as coordenadas do ponto que pretende representar, os dois argumentos da função, e execute a função clicando na tecla **enter**. Observe o ecrã.
- Para sair da representação gráfica pode clicar na tecla **enter** ou na tecla **del**, uma primeira vez para terminar a representação e uma segunda vez para voltar ao interpretador (Shell).
- Execute novamente a função **ponto**, agora com outras coordenadas (por exemplo **ponto(10,10)**, **ponto(-4,6)**, ...) e observe o ecrã com a representação gráfica do ponto. Constatará que para algumas coordenadas o ponto não surge visível no ecrã. Teste, usando esta função, a janela de visualização assumida por defeito.

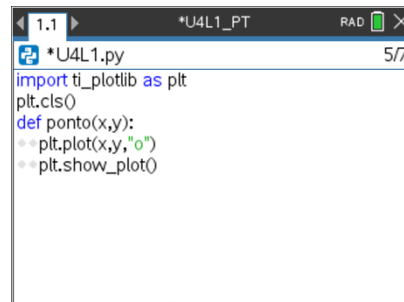
OBSERVAÇÃO:

Ao escrever um programa usando as instruções para obter uma representação gráfica, será necessário especificar os parâmetros da janela de visualização, assim como, eventualmente, tornar visível um sistema de eixos, o nome dos eixos, uma grelha de fundo, etc.



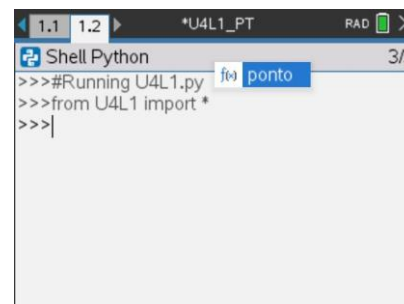
```

1.1 *U4L1_PT
*U4L1.py 4/6
import ti_plotlib as plt
plt.cls()
def ponto(x,y):
    plt.plot(x,y,"marca")
    
```



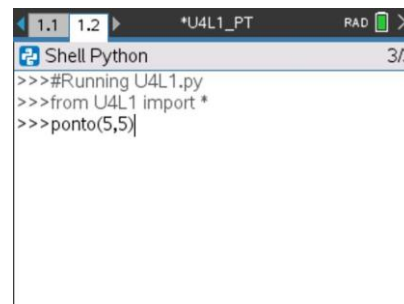
```

1.1 *U4L1_PT
*U4L1.py 5/7
import ti_plotlib as plt
plt.cls()
def ponto(x,y):
    plt.plot(x,y,"o")
    plt.show_plot()
    
```



```

1.1 1.2 *U4L1_PT
Shell Python 3/3
>>>#Running U4L1.py
>>>from U4L1 import *
>>>ponto
>>>|
    
```



```

1.1 1.2 *U4L1_PT
Shell Python 3/3
>>>#Running U4L1.py
>>>from U4L1 import *
>>>ponto(5,5)
    
```



Etapa 2: Melhorar a representação gráfica

Insira a função **plt.cls()** antes da definição da função, para evitar que outros elementos fiquem sobrepostos à representação gráfica.

A partir das diferentes opções do submenu de configurações do módulo TI PlotLib, adicione ao seu programas as instruções seguintes que lhe permitam:

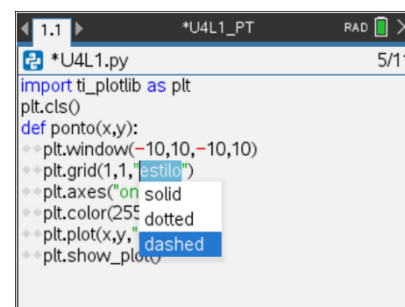
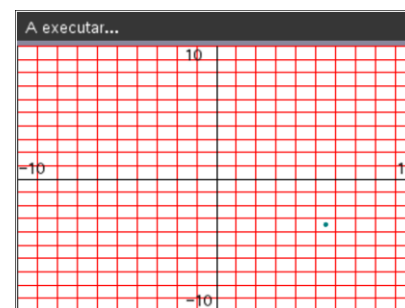
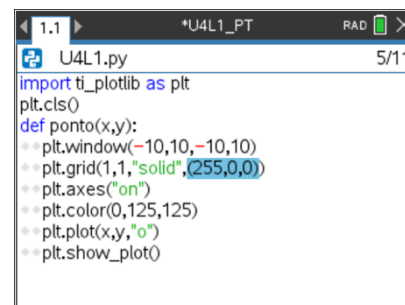
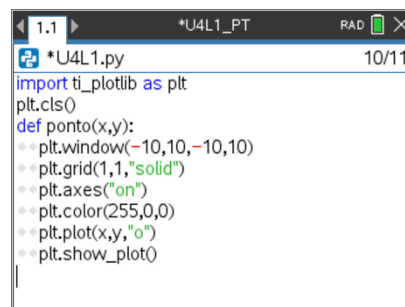
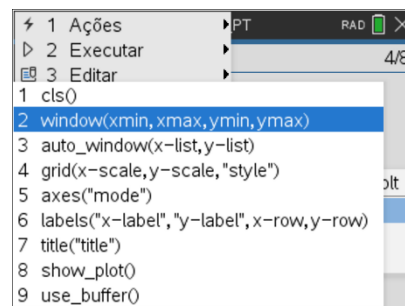
- definir uma janela gráfica com as seguintes características: Xmin = -10; Xmax = 10; Ymin = -10 e Ymax = 10 (submenu **2: Configurar** seguido de opção **2: window(xmin,xmax,ymin,ymax)**).
- exibir uma grelha (submenu **2: Configurar** seguido de opção **4: grade 4**), sendo que o tipo de grelha é deixado à sua escolha.
- exibir os eixos coordenados (opção **5: axes("mode")**).
- alterar a cor do ponto (menu **3: Desenhar**, a seguir **1: color(vermelho,verde,azul)**).

OBSERVAÇÕES:

- Para cortar, copiar ou colar uma linha, use os atalhos **Ctrl + X**, **Ctrl + C** e **Ctrl + V**, respetivamente.
- A cor de um ponto ou de uma linha deve ser definida pelos parâmetros da mistura de cores RGB (red-vermelho, green-verde, blue-azul), cada parâmetro pode assumir um valor no intervalo [0; 255]. As cores são codificadas em 8 bits, ou seja, $2^8 = 256$ possibilidades, incluindo o 0 que corresponde a ausência da cor, por exemplo RGB(255, 0, 0) corresponde ao vermelho "puro" já que está ausente o verde e o azul.
- Também é possível desenhar a grelha de fundo com cores, completando para tal a instrução **grid** da seguinte forma: **grid(x-scale, y-scale, "style", (r,g,b))**. Veja-se o exemplo de formatação usada no programa ao lado e a imagem de execução da função **ponto()**.

DICA:

Utilize a tecla **tab** para deslocar o cursor mais facilmente entre os diferentes campos de preenchimentos das funções. Em alguns campos de preenchimento surge uma ajuda contextual, abrindo-se para baixo um menu de escolha com as opções permitidas, conforme imagem ao lado. Esta situação ocorre por exemplo no campo de estilo de representação para um ponto, de seleção do tipo de grelha, etc. Note que, depois de preenchido o campo esta estratégia já não resulta.



Voltemos ao nosso programa melhorado, sem formatações extra.

- Execute o programa, atalho **ctrl** + **R**, e já na página do interpretador, escreva a função **ponto()** e execute-a clicando na tecla **enter**.
- Deverá obter um ecrã idêntico ao que se encontra ao lado.

OBSERVAÇÃO:

Como no programa a função **cls()** se encontra fora da função **ponto()**, ao executar o programa de imediato é executada a função **cls()** e por isso surge uma janela de desenho vazia, sendo necessário premir duas vezes a tecla **enter** para voltar ao interpretador, e aí executar a função **ponto()**. Esta situação resolve-se colocando a função **cls()** dentro e no início da função **ponto()**, experimente.

ATENÇÃO:

- Ao executar um programa, atalho **ctrl** + **R**, o interpretador é reiniciado, o que significa que se perde o histórico do interpretador, como por exemplo variáveis definidas anteriormente no interpretador ou até bibliotecas que haviam sido “embutidas” no interpretador (por exemplo outros programas).
- Note ainda que, ao guardar o programa, atalho **ctrl** + **B**, sem o executar a partir do editor, as alterações não são consideradas no interpretador, para tal sempre que alterar o programa tem que o executar novamente a partir do editor.

Altere novamente o programa de forma a colocar legendas nos eixos do referencial, por exemplo, “abscissa” e “ordenada”.

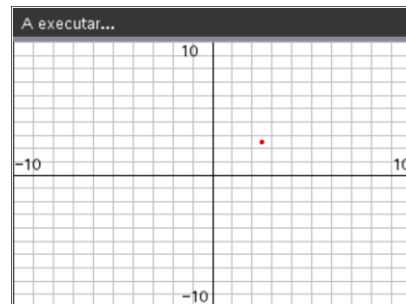
Para isso, inclua no seu programa (dentro da função, independentemente da localização) uma linha com a instrução **plt.labels**. Acede a esta função na opção **7: labels()** do submenu **2: Configurar** do módulo **7: TI PlotLib**.

Vamos, ainda, adicionar um título à nossa representação gráfica, usando para tal a função **plt.title**, inserindo a função **7: title(“title”)**, do submenu **2: Configurar** do módulo **7: TI PlotLib**.

OBSERVAÇÕES:

A função de **labels(“Etiqueta-x”, “etiqueta-y”, x-row, y-row)** permite designar os eixos coordenados, colocando as etiquetas na linha x e coluna y, que por defeito são a linha 12 para x e a coluna 2 para y, respetivamente justificado à esquerda e à direita.

- Execute novamente o programa, atalho **ctrl** + **R**, e na página do interpretador, execute a função **ponto(-3.7,2.4)**.
- Deverá obter um ecrã idêntico ao que se encontra ao lado.

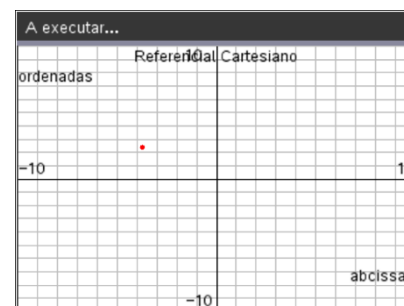


```

1.1 1.2 *U4L1_PT RAD 9/10
*U4L1.py
import ti_plotlib as plt
def ponto(x,y):
    plt.cls()
    plt.window(-10,10,-10,10)
    plt.grid(1,1,"solid")
    plt.axes("on")
    plt.color(250,0,0)
    plt.plot(x,y,"o")
    plt.show_plot()
    
```

```

1.1 1.2 *U4L1_PT RAD 9/12
*U4L1.py
import ti_plotlib as plt
def ponto(x,y):
    plt.cls()
    plt.window(-10,10,-10,10)
    plt.grid(1,1,"solid")
    plt.axes("on")
    plt.labels("abscissas","ordenadas",12,2)
    plt.title("Referencial Cartesiano")
    plt.color(250,0,0)
    plt.plot(x,y,"o")
    plt.show_plot()
    
```





RECORDE QUE:

Os caracteres especiais (acentuados e outros) podem ser obtidos de duas formas: pressionando sucessivamente a tecla **⌘** após inserir a letra ou procurando na tabela do código ASCII, usando o atalho **⌘** + **⌘**, o caracter pretendido. No software bastará utilizar, como habitual, o teclado do computador.

MAIS ALÉM:

Faça um *upgrade* ao seu programa, construa uma nova função que lhe permita representar graficamente um segmento de reta dadas as coordenadas dos seus extremos.

- Começamos por colocar uma linha de comentário (atalho **⌘** + **T**), antes de cada uma das funções do programa, para desta forma qualquer utilizador saber facilmente o objetivo da função.
- Deseguida construa a função `segmento()` tendo em atenção as instruções essenciais constantes na imagem ao lado. Poderá, se entender, alterar algumas das configurações/parâmetros definidos.
- Para representar o segmento de reta iremos utilizar a função `plt.pen()`, acessível através do menu **3: Desenhar** (opção **9: pen** ("**size**", "**style**"), que permite configurar a largura e o estilo da linha.)

```

1.1 1.2 *U4L1_PT RAD 2/12
*U4L1.py
import ti_plottlib as plt
# Representação de um ponto
def ponto(x,y):
    plt.cls()
    plt.window(-10,10,-10,10)
    plt.grid(1,1,"solid")
    plt.axes("on")
    plt.labels("abscissas","ordenadas",12,2)
    plt.title("Referencial Cartesiano")
    plt.color(250,0,0)
    plt.plot(x,y,"o")

```

```

1.1 1.2 *U4L1_PT RAD 13/23
*U4L1.py
# Representação de um segmento
def segmento(x0,y0,x1,y1):
    plt.cls()
    plt.window(-10,10,-10,10)
    plt.grid(1,1,"dashed")
    plt.axes("on")
    plt.title("Segmento de Reta")
    plt.color(250,0,0)
    plt.pen("medium","solid")
    plt.line(x0,y0,x1,y1,"default")
    plt.show_plot()

```

OBSERVAÇÃO:

Para configurar as funções `plt.pen()` e `line()`, a primeira com os campos relativos ao tamanho (estrito, médio, largo) e ao estilo (sólido, ponteadado, tracejado) e a segunda com o campo relativo ao modo (por feito, setas), surge uma janela de escolha de opção conforme imagem ao lado. Note que a imagem é uma composição de captura de ecrãs, não sendo possível obter como um único ecrã.

```

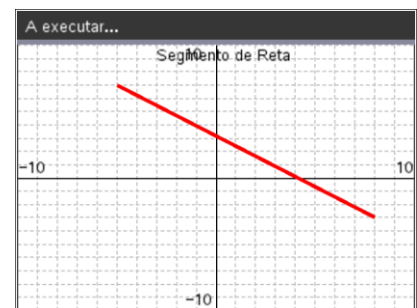
1.1 1.2 *U4L1_PT RAD 21/24
*U4L1.py
# Representação de um segmento
def segmento(x0,y0,x1,y1):
    plt.cls()
    plt.window(-10,10,-10,10)
    plt.grid(1,1,"dashed")
    plt.axes("on")
    plt.title("Segmento de Reta")
    plt.color("thick","dashed")
    plt.pen("tamanho","estilo")
    plt.line(x0,y0,x1,y1,"modo")
    plt.show_plot()

```

RECORDE QUE:

A captura de ecrã no software pode ser obtida através do atalho **Ctrl + J**, ou usando a ferramenta **Captura de ecrã**.

- Por fim, execute mais uma vez o programa, atalho **⌘** + **R**, e na página do interpretador, execute a função `segmento(-5,7,8,-3)`.
- Obterá um ecrã idêntico ao que se encontra ao lado.



ATENÇÃO:

Após executar, no interpretador, qualquer uma das funções definidas no programa a cor que fica definida para desenho é a vermelha (cor escolhida para o ponto e o segmento de reta), por isso se executarmos de seguida qualquer uma das funções todo os elementos representados terão essa cor. Assim, será indicado no início de cada função definir a cor por defeito como sendo o preto, RGB(0,0,0).

