Unidade 6: Utilização das bibliotecas TI Hub & TI Rover | **Lição 1: Os sensores integrados no Hub**

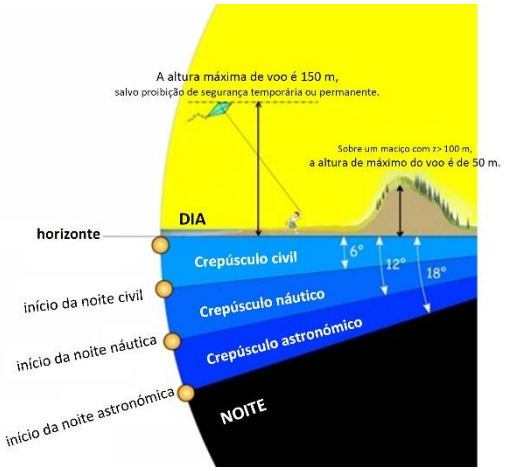
Nesta primeira lição da Unidade 6 vamos aprender como utilizar a biblioteca TI Hub com o objetivo de controlar os dispositivos integrados no TI-Innovator™ Hub.

Objetivos:

- Explorar o módulo **TI Hub**.
- Escrever um programa integrando a biblioteca TI Hub para dispositivos integrados.

Nesta lição, iremos utilizar a biblioteca do **TI Hub** para observar visualmente uma mudança na luminosidade para, subsequentemente, simular uma mudança de crepúsculo ou para registar um conjunto de medições durante o nascer ou o anoitecer.

Começará, também, a considerar como associar esta biblioteca com aquelas que já conhece das Unidades anteriores (**TI PlotLib** e **TI System**) por forma a desenvolver um projeto científico completo.



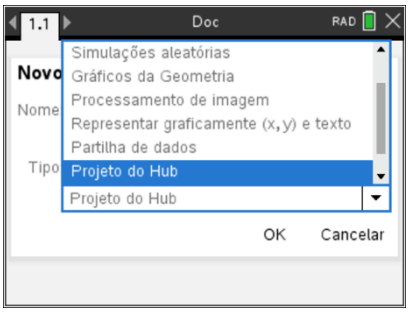
Considere o algoritmo simples que se apresenta abaixo, o programa que iremos elaborar basear-se-á nele.

ALGORITMO

Medir a intensidade luminosa ambiente:
Lum0 ← medida ± (tolerância?)
Alterar a intensidade luminosa
(# lâmpada; tampa na frente do sensor)
Lum1 ← medida
Se Lum1 > Lum0:
Então ligar LED RGB vermelho (2s)
Senão Se Lum0 < Lum1 :
Então ligar LED RGB verde (2s)
Senão não fazer nada

IMPLEMENTAÇÃO DO PROGRAMA:

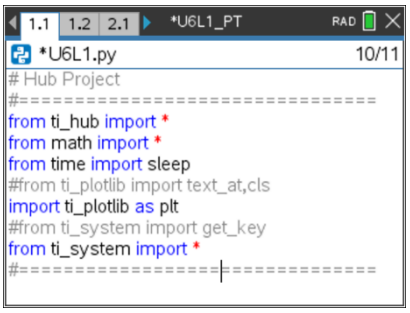
- Inicie um novo programa no editor de TI-Python, designe-o por **U6L1**.



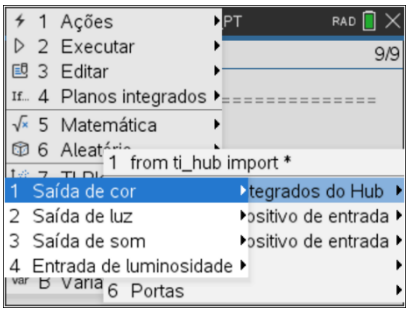
- Este programa deve integrar a biblioteca TI Hub, podendo optar por várias formas de o fazer:

  a) ao adicionar uma página com o editor de TI-Python, especificando o tipo geral de programa. Desta forma irá integrar, pelo menos, a biblioteca correspondente ao nome do tipo.

b) Ou, também, pode começar com um programa em branco e, de seguida no início do programa, incorporar manualmente as bibliotecas de que necessita. Estratégia usada nas unidades anteriores.



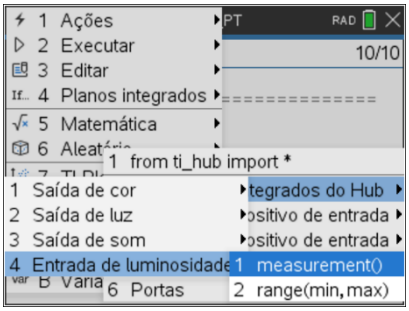
Ao selecionar o tipo de documento como Projeto do Hub, irá obter um ecrã idêntico ao do lado. Ficarão ativas duas bibliotecas TI Hub e Matemática, e três funções de três bibliotecas diferentes. Por vezes é necessário adicionar mais bibliotecas, algo que se pode aperceber ao executar o programa.



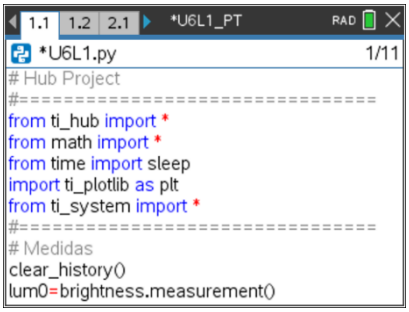
Usaremos o sensor de luz integrado no TI-Innovator, bem como o díodo RGB. Para que o programa seja capaz de os gerenciar, teremos de integrar as bibliotecas correspondentes. Para fazer isso, deve selecionar no menu o módulo **6: TI Hub**, depois opção **1: Dispositivos integrados do Hub**, e por fim **1: Saída de cor**.

OBSERVAÇÃO:
As outras duas opções dizem respeito aos sensores e atuadores que serão conectados diretamente às portas de entrada IN ... e OUT ... do Hub ou possivelmente a outras portas.

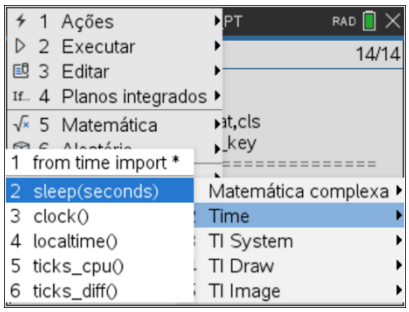
- Para que o programa apresente as informações num ecrã "limpo", limpe-o usando a instrução **clear_history()** acessível através da biblioteca **TI System**. (Tecla menu, seguido de opção **A: Mais módulos**, depois **3: TI System** e por fim **B: clear_history()**)



- Crie uma variável **lum0**, à qual deve atribuir a medida de luminosidade do momento. A instrução **brightns.measurement()** está no módulo **8: TI Hub**, depois opção **2: Dispositivos integrados do Hub**, seguido de **4: Entrada de luminosidade**, e por **1: measurement()**.



- De seguida insira o código que permita o programa exibir uma mensagem solicitando que o utilizador altere a intensidade da luz nas proximidades do sensor integrado: **plt.text_at()**. Esta função e a função **cls()**, para limpar a janela gráfica, encontram-se no módulo **7: TI PlotLib**.

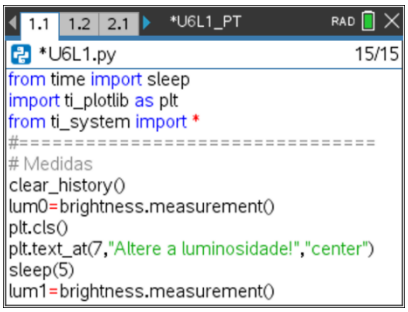


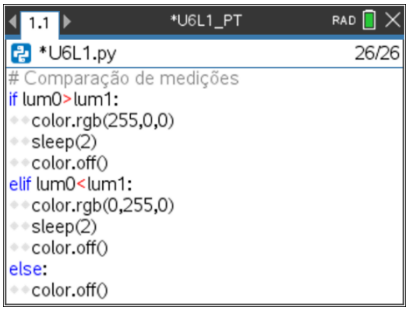
- Adicione um tempo de espera, função **sleep()**, que será útil para efetuar alteração da intensidade da luz. (Tecla menu, seguido de opção **A:Mais módulos**, depois **2:Time** e por fim **2: sleep(seconds)**)

- Crie uma nova variável **lum1** à qual irá atribuir o valor da nova medição da intensidade luminosa.

- Por fim a comparação entre as medidas **lum0** e **lum1**. Utilizando a estrutura de controlo condicional **IF** e em função do resultado lógico da condição **lum0** > **lum1**, o LED RGB ficará verde ou vermelho durante 2 segundos. Caso **lum0** = **lum1**, não ocorrerá nada!

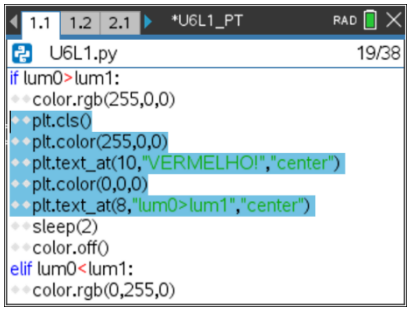
- As funções **color.rgb(Red,Green,Blue)** e **color.off()** encontram-se no módulo **8: TI Hub**, depois opção **2: Dispositivos integrados do Hub**, seguido de **1: Saída de cor**.

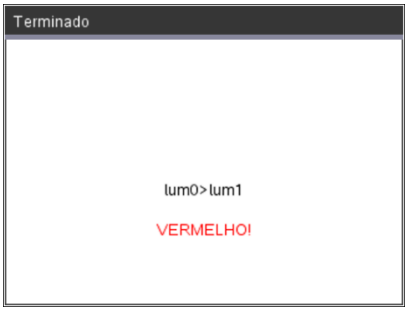


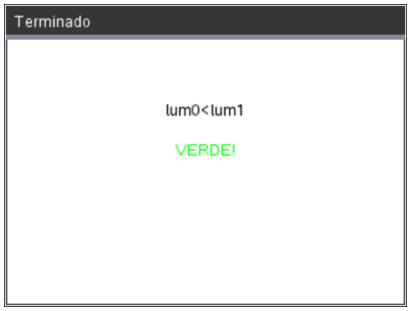


SUGESTÃO:

Pode acrescentar ao seu programa uma instrução que lhe permita observar, no ecrã da unidade portátil, qual a cor que deverá surgir no Hub em função das medidas, para isso bastará acrescentar uma instrução de texto após cada teste. Esta estratégia poderá ser útil para validar programas e verificar se o resultado é o esperado.

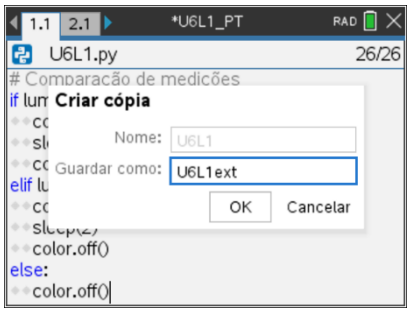






EXTENSÃO DA ATIVIDADE:

Faça uma cópia do programa anterior, designe-o por **U6L1ext** e mova-o para um Problema 2 do seu documento tns. Para mover uma página entre Problemas pode usar a vista de Gestor de Página, clicando ctrl + [:::] e usar o menu de contexto, ctrl + menu, sobre a página a mover.



Também poderá, se entender, abri um novo documento e começar de início o programa.

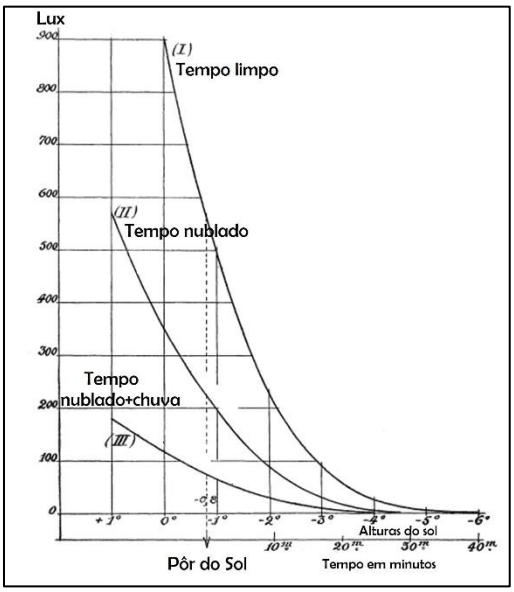
Altere o programa de forma que registe as medidas de luminosidade ao longo de 40 minutos.

OBSERVAÇÃO:

Note que o sensor de intensidade luminosa do TI-Innovator não é calibrado em Lux, mas isso não é problema, pois estamos interessados apenas nas variações de intensidade e não em sua medida em Lux.

No programa, as medidas da intensidade luminosa serão guardadas numa lista, que designaremos por **r**, que a iniciaremos como vazia **r=[ ]**. O registo dos tempos de recolha dos dados serão também guardados numa lista, **t[ ]**.

Abaixo, no ecrã da esquerda, propõe-se o código para a parte do programa que executará a recolha de dados e os guardará em listas, (#Recolha de dados), e à direita encontra-se um ecrã com o código relativo à representação gráfica dos dados, (#Representação gráfica).



RECOLHA DE DADOS

1.1 2.1 *U6L1_PT RAD

*U6L1ext.py 18/18

```
#================================
# Recolha de dados
clear_history()
def bri(n):
    r=[]
    t=[]
    for i in range(n):
        r.append(brightness.measurement())
        t.append(i)
        sleep(60)
    return t,r
```

REPRESENTAÇÃO GRÁFICA

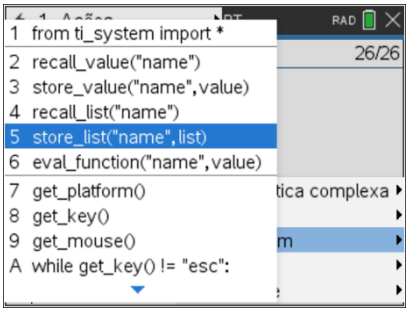
1.1 2.1 *U6L1_PT RAD

*U6L1ext.py 29/29

```
# Representação Gráfica
plt.cls()
plt.auto_window(t,r)
plt.labels("t (min)","r",12,2)
plt.title("Crepúsculo")
plt.color(255,0,255)
plt.scatter(t,r,"+")
store_list("Tempos",t)
store_list("Luminosidade",r)
plt.show_plot()
```

OBSERVAÇÃO:

No ecrã da direita, também se apresenta a proposta de exportação dos dados para listas das restantes aplicações da TI-Nspire CX II, podendo serem exploradas, por exemplo, na aplicação Listas e Folha de Cálculo.



- A função **store.list()** encontra-se no menu **A: Mais módulos**, na opção **3: System** e depois opção **5: store.list(“name”,list)**.
- A função **bri(n)** realiza aquisição de dados a cada minuto, dada a instrução **sleep(60)**, por n minutos e retorna as listas **t** e **r**.
- De seguida são representados graficamente os dados constantes nas listas **t** e **r** e exportados para as listas Tempos e Luminosidade da TI-Nspire CX II.