Unidade 8: micro:bit com Python | **Lição 3: Absorção de luz**

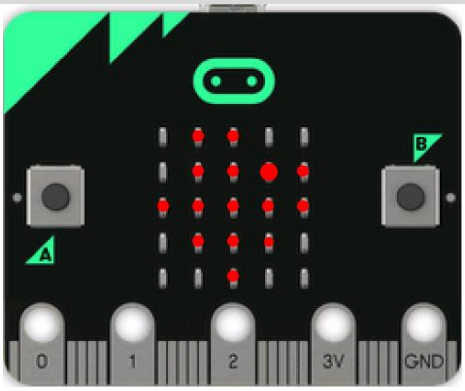
Nesta lição, deverá monitorizar o sensor de luminosidade do micro:bit e organizar os dados numa lista TI-Nspire para posterior análise.

Objetivos:

- Ler e exibir o sensor de brilho no micro:bit
- Transferir dados do python para a TI-Nspire CX II
- Analisar os dados recolhidos do micro:bit

Dica para o Professor: Esta lição é baseada numa aproximação da lei do inverso do quadrado para uma fonte de luz pontual. Esta lei relaciona a distância à fonte de luz e a intensidade da luz, mas não o tempo. No entanto, um movimento cuidadoso, regular e constante do micro:bit, distanciando-se da fonte de luz, criará uma relação linear entre o tempo e a distância, sendo o tempo um bom substituto para a distância.

1. O micro:bit é capaz de ler o nível de luminosidade ambiente usando os *display* LEDs. Sim, os *display* LEDs também podem ser usados como um dispositivo de entrada!



Dica para o Professor: para mais informação sobre o nível de luminosidade do micro:bit ver:
Sensing changes in light on the micro:bit : Help & Support

2. Iniciar um programa python num novo documento.
   Pressionar o botão **[home]** e selecionar **Novo > Adicionar Python >Novo…**
   Demos ao programa o nome de **brilho.**

   Para verificar os valores que o sensor de luz consegue detetar, escreva o programa abaixo usando os menus do BBC micro:bit:

```
from microbit import *

while get_key() != "esc":
    brilho = display.read_light_level()
    print("brilho = ",brilho)
```

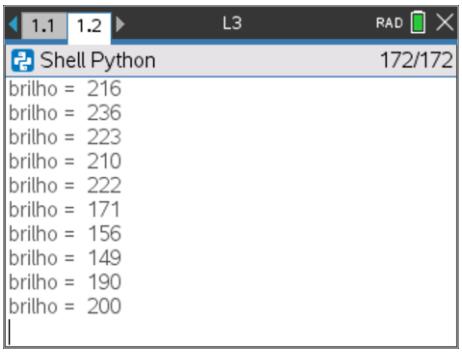
   Encontra **var= .read_light_level()** no **[menu] > BBC micro:bit >Sensors ; bright** será o nome da variável que escreverá.

   Relembrar: **[menu] > Mais módulos > BBC micro:bit > Commands> while get_key() != "esc":**

   Não se esqueça de identar as duas últimas linhas de código, pois assim fazem parte do corpo do ciclo **while**.

3. Corra o programa e aponte o ecrã do micro:bit para a fonte de luz. O que for exibido no ecrã não é relevante. Aproxime e afaste o micro:bit e observe a variação dos valores no ecrã da TI-Nspire CX II. Deverá observar valores entre 0 e 255.
   Como seria de esperar, quanto mais longe o micro:bit se encontra da fonte de luz, menor será o nível de luminosidade. Agora vai adicionar código ao programa, de modo que este possa recolher os dados relativos ao nível de luz e a criar um gráfico de dispersão luz vs. tempo.

   Pressione **[esc]** para terminar o programa e voltar ao Editor.



4. Crie duas listas vazias antes do ciclo **while**:

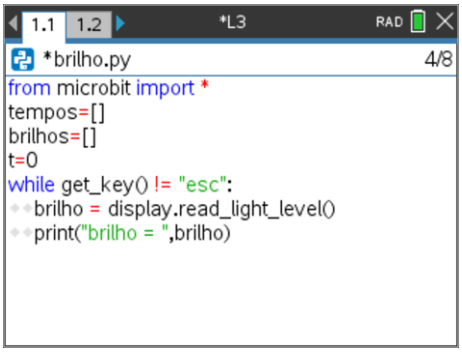
   tempos = [ ]
   brilhos = [ ]

   Encontre os parênteses retos no teclado **[ctrl] [left parenthesis]** ou no menu **[menu] > Planos integrados > Listas**

   Antes do ciclo while, adicione uma linha para iniciar um contador de tempo (**t**) em 0:

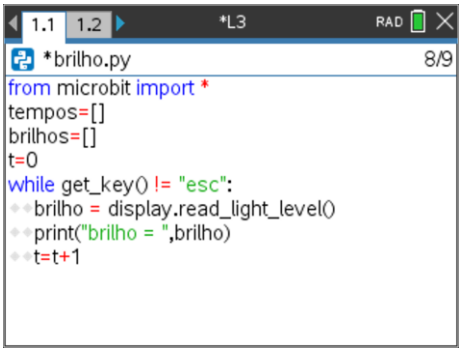
   t = 0

   Evite usar a palavra 'tempo' como variável porque já existe um modulo de tempo. Além disso, é boa prática pluralizar os nomes das listas visto que estas podem conter vários valores.



5. No corpo do ciclo, depois da linha de código **print**, adicione uma linha de código de modo a incrementar a variável **t**. Utilizaremos um segundo como intervalo de tempo entre leituras do nível de luminosidade, ou seja:

   t = t + 1



6. Adicione **(append)** os valores **brilho** e **t** às suas respetivas listas, utilizando os códigos:

   times.append(t)
   brights.append(brilho)

   .**append( )** pode ser encontrado em **[menu] > Planos integrados > Listas**

   Estes códigos adicionam o valor atual de luminosidade e o valor t (tempo) às listas.

7. De modo a controlar o intervalo de tempo da amostragem, adicione:

   ♦♦**sleep(1000)**

   a seguir aos dois comandos .**append**. Este commando pausa a recolha de dados durante um segundo entre amostras. **sleep( )** pode ser encontrado em:

   [menu] > BBC micro:bit > Commands

   É importante lembrar que quando usamos o micro:bit, **sleep( )** *utiliza milisegundos como argumento. Esta amostragem ocorre uma vez por segundo.*

8. Depois do corpo do ciclo (*não se identa mais!*), armazene (**store)** as duas **listas** de python em duas listas TI-Nspire, utilizando os mesmos nomes em ambos os ambientes de trabalho.

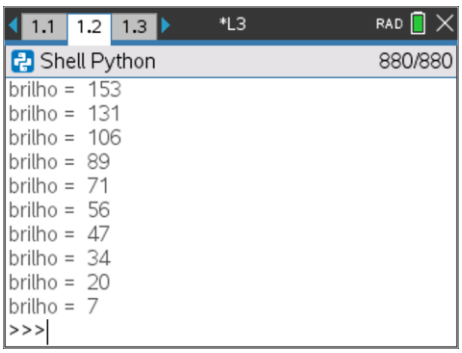
   Comece no início de uma nova linha (sem identação!).
   Encontra **store_list()** em **[menu] > BBC micro:bit > Commands**. São necessários dois argumentos: o "nome da lista TI-Nspire" entre aspas e o nome da lista python, sem aspas.

   Neste programa, as listas python são armazenadas sob a forma de listas TI-Nspire imediatamente antes do programa terminar, quando pressionas **[esc]** *para sair do ciclo.*

9. Corra o programa. Comece com o micro:bit numa posição próxima da fonte de luz. Uma lâmpada ou a lanterna de um smartphone funcionam bem. Lenta mas firmemente, afaste o micro:bit da fonte a uma velocidade constante, até que a luminosidade registada apresente um valor inferior a 10.
   Pressione **[esc]** para terminar o programa.

   Repita o processo até sentir que tem um "bom" conjunto de dados.

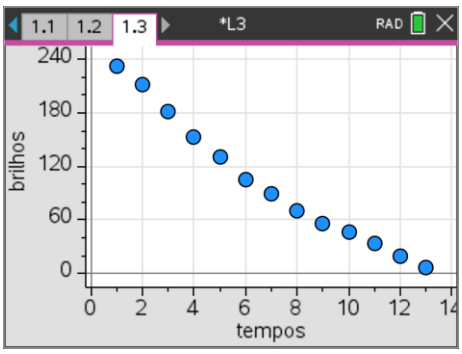
   Dica: A lanterna de um smartphone funciona lindamente!



10. Quando o programa terminar, adicione uma página ao documento pressionando **[ctrl] [doc]**. Selecione **'Adicionar Dados e Estatística'.** Deverá ver uma página semelhante à representada na imagem. Uma das nossas listas aparece sob a forma de dispersão. Agora, ten de organizar o gráfico…

11. Clique no espaço '*Clicar para adicionar variável*' no fundo do ecrã e selecione a lista **tempos**. Esta é a variável *independente*.

    Clique no espaço '*Clica para adicionar variável*' no lado esquerdo do ecrã e selecione a lista **brilhos**. Esta é a variável *dependente*. O gráfico da dispersão deverá aparecer como na figura.



    Poderá ter de correr o programa várias vezes de modo a obter uma "boa" curva como a apresentada na figura. Poderá ser também necessário ajustar a janela para dados diferentes.

    Poderá ser ainda necessário ajustar o intervalo de tempo entre amostras, assegurando sempre que editas o argumento da variável **sleep(1**000 **)** *e o valor do contador. (t = t +* ***1****).*

    Em que é que repara?

    - *Que padrão observa? Que modelo matemático se adequa ao comportamento físico?*
    - *Utilize as ferramentas de análise de dados da TI-Nspire para determinar um bom modelo matemático para o seu conjunto de dados.*

Dica para o Professor: Esta lição é baseada numa aproximação da lei do inverso do quadrado para uma fonte de luz pontual. Esta lei relaciona a distância à fonte de luz e a intensidade da luz, mas não o tempo.

intensidade = k / (distância2)**

Um afastamento cuidadoso e constante do micro:bit da fonte de luz, establece uma relação linear entre o tempo e a distância, sendo aplicável a proporção do inverso quadrado.

É possivel modificar esta atividade utilizando uma medida da distância entre a fonte de luz e o micro:bit, accionando uma tecla para recolha de dados em cada ponto. Por exemplo, afastar o micro:bit 10cm da fonte de luz a cada momento de recolha. Utilize **get_key(1)** para pausar o programa até que o micro:bit se encontra devidamente posicionado para recolha da amostra seguinte.