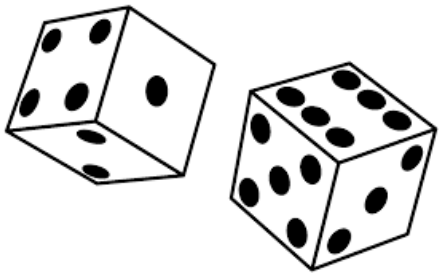
Unidade 8: micro:bit com Python **Aplicação: Lançar o dado**

Nesta Apliacação, deverá escrever um programa que permita recolher dados utilizando o micro:bit e executar o programa enquanto observa um gráfico de barras a crescer numa página em separado da TI-Nspire

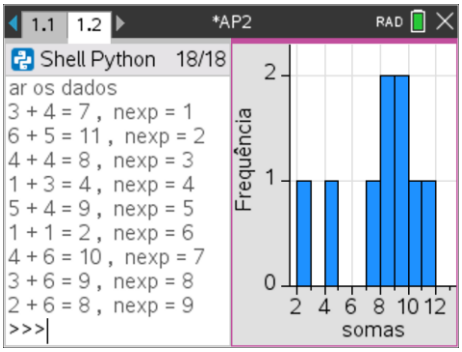
Objectives:

- Escrever um programa para recolha de dados no micro:bit
- Criar um gráfico dinâmico em Dados e Estatística

1. Este projeto requer uma compilação de todas as aptidões de micro:bit adquiridas nas últimas três atividades: escrever um programa que utilize um gesto, como "agitar" (ou o pressionar de uma tecla) para recolher dados, armazená-los numa lista sob a forma de variável TI-Nspire e…

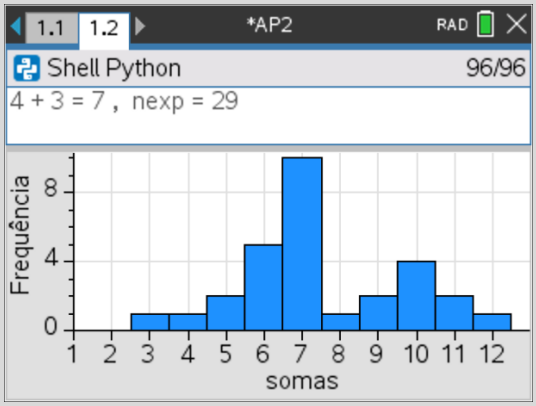


2. … criar uma página TI-Nspire que:
   - Corra o programa python num lado do ecrã (python Shell) e
   - Exiba um gráfico de pontos (ou histograma) dos dados recolhidos, em simultâneo com o funcionamento do programa python (utilizando a aplicação Dados e Estatística).



Dica para o Professor: O ecrã dividido da TI-Nspire demonstrado acima pode ser orientado vertical ou horizontalmente. Quando se utiliza o ctrl-4 para agrupar duas aplicações na mesma página, o modo base é vertical, como é possível observar na lição dos alunos. **Para mudar para um layout horizontal, pressione [doc] > Esquema de página > Selecionar esquema > Esquema 3.**

Também é possível ajustar a barra do separador para reduzir o tamanho da aplicação python Shell e aumentar o tamanho da aplicação Dados e Estatística.



Esta lição produz um gráfico de pontos e não um histograma. Instruções acerca de como converter o gráfico de pontos num histograma através da aplicação Dados e Estatística podem ser encontradas mais tarde neste documento, nas Dicas para o Professor.

3. Comece o seu programa micro:bit com os *imports* habituais, incluindo o módulo **random** e cria uma lista vazia com o nome **somas**:

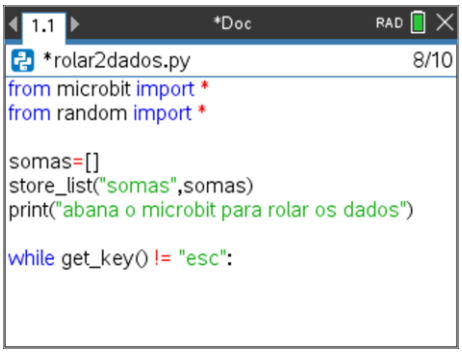
   somas = [ ]

   Armazene, *imediatamente,* a lista numa variável TI-Nspire (com o mesmo nome).

   store_list("somas", somas)

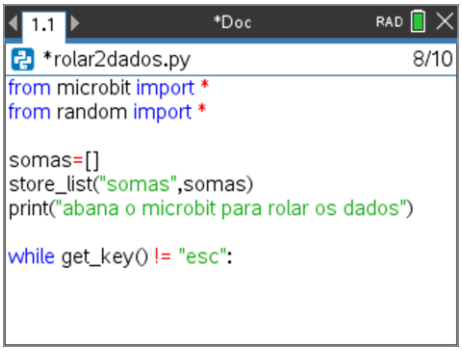
   de modo a que a lista TI-Nspire se encontre também vazia.

   print( ) algumas instruções para o utilizador, antes do ciclo começar. Vamos usar o gesto "agitar" para lançar os dados.



4. No corpo do ciclo **while**, use o gesto para
   - **Lançar** *dois d*ados (gerar dois inteiros aleatórios)
   - **Adicionar** os valores
   - **Adicionar** a soma à lista **somas**
   - **Exibir** os valores dos dois dados, a sua soma e o número do lançamento no ecrã da TI-Nspire. Dica: **len(somas)** é o número do lançamento.
   - **Exibir** ambos os valores do dado no display do micro:bit
   - **Armazenar** a **lista** numa variável da TI-Nspire

   Tente agora.

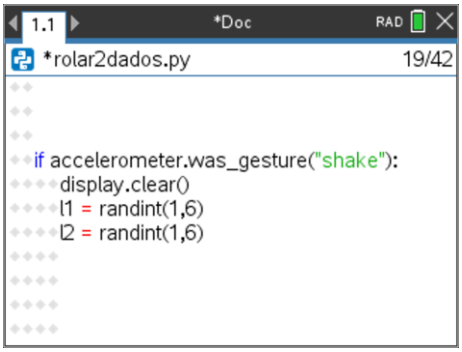


Dica para o Professor: Os alunos devem ter conhecimentos suficientes das lições 1, 2 e 3 para desenvolver este programa. Se se verificar alguma dificuldade com menus, encoraje-os a consultar as lições anteriores

5. Para lançar o dado, use um gesto ou o prima uma tecla:

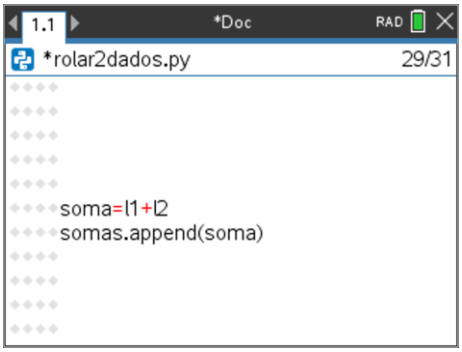
   if accelerometer.was_gesture("shake"):
   display.clear()
   l1 = randint(1,6)
   l2 = randint(1,6)

   Tome atenção às identações.



6. Adicione-os e anexe (.**append)** a **soma** à lista **sums**:

   sum = l1 + l2
   sums.append(soma)



7. Exiba os dados no *display* do micro:bit. Lembre-se que os dados podem ter o mesmo valor e como tal, é fundamental garantir que ambos aparecem.

```
display.clear()
display.show(I1)
sleep(250)
display.clear()
display.show(I2)
sleep(250)
```

Talvez prefira utilizar um intervalo mais longo no comando **sleep( )**.
Se tiver escrito o código de forma correta e na sequência apropriada, tente correr o programa e agitar o micro:bit. Deverá ver dois valores exibidos no micro:bit.

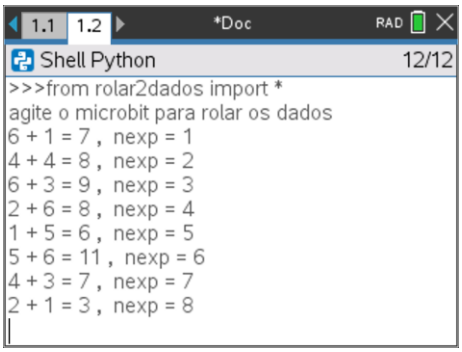
Dica do Professor: Outro **sleep( )** poderá ser útil no ciclo, fornecendo ao micro:bit uma chance para monitorizar o gesto.

8. Adicione código para exibir o dado, as somas e o número de lançamentos no ecrã da TI-Nspire. Para tal, podemos utilizar um simples comando **print()**:

```
    print (I1, "+", I2,"=",soma,", ","nexp =", len(somas))
```

Resultando nas linhas demonstradas na imagem.

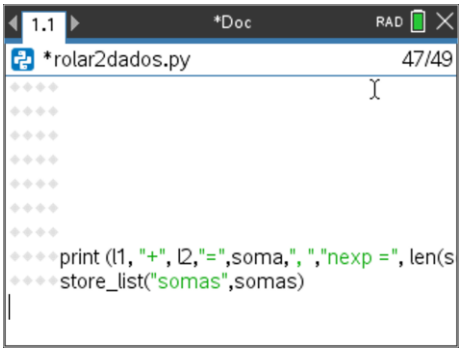
Cuidado com a identação!



9. Armazene a lista python **sums** numa lista TI-Nspire com o mesmo nome:

```
    store_list("somas", somas)
```

O comando **store_list( )** *encontra-se situado profundamente nos blocos while e if, de modo a que a lista TI-Nspire seja atualizada de cada vez que um novo par de dados é lançado.*

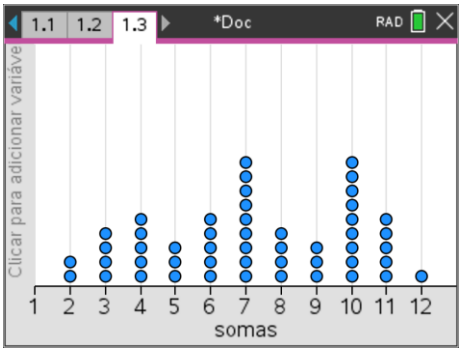


Dica para o Professor: Note a falta de uma variável 'contador' na lista de comandos acima. Esta não é necessária, visto que o programa utiliza **len(somas)** como número de lançamentos.

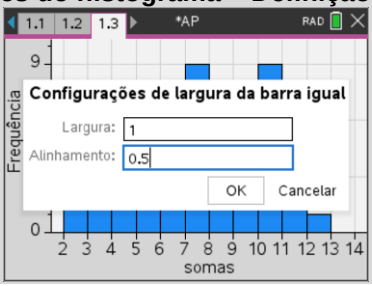
10. Quando estiver satisfeito com o funcionamento adequado do programa, encontra-se preparado para conectar o programa python às capacidades gráficas da TI-Nspire. Corra o programa e gere 50 lançamentos. Pressione **[esc]** para terminar o programa.

    Na python Shell (a seguir a **>>>**) pressione **[ctrl] [doc]** ou **[ctrl] [I]** para inserir uma página. Selecione a aplicação **Dados e Estatistica**. Deverá observar um ecrã semelhante àquele apresentado na imagem à direita. Os dados da soma encontram-se dispersos pelo ecrã.

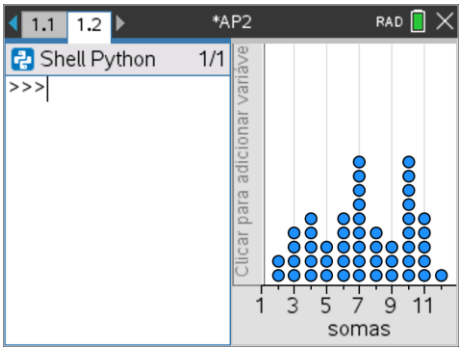
11. Clique em '*Clique para adicionar uma variável*' no fundo do ecrã e selecione a lista das variáveis soma **(sums)**. Os dados, previamente dispersos pelo ecrã, encontram-se agora organizados segundo o eixo dos xx, de acordo com o seu valor. A janela encontra-se ajustada ao conjunto de dados analisados. Este é o Gráfico de Pontos **(Dot Plot)**.

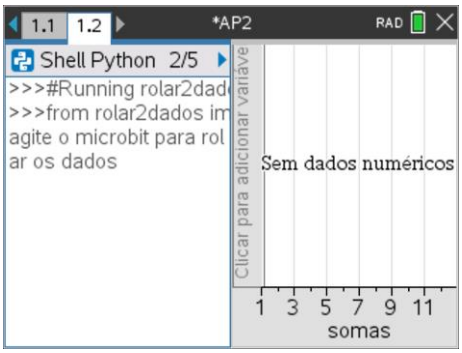


Dica para o Professor: *É possível transformar o gráfico de pontos num histograma: pressione* **[menu] > Tipo de gráfico > Histograma**. *É importante ajustar também o alinhamento das barras para 0.5, de modo a que as barras se encontrem centradas de acordo com os seus valores correspondentes no eixo dos xx. Pressione* **[menu] > Propriedades do gráfico > Propriedades do histograma > Definição das barras** *e defina* **Alinhamento** *para* **0.5**.



12. Volte para a página da aplicação python Shell (**[ctrl] [leftarrow]**) e pressione **[ctrl] [4]** para a 'agrupar' com a aplicação de Dados & Estatística, criando uma página de ecrã dividido, com o python Shell à esquerda e os Dados e Estatística à direita.





13. O Shell foi 're-iniciado' e como tal, pressionando **[ctrl] [R]** não irá correr novamente o programa. Volte para o editor python e pressione **[ctrl] [R]** para correr o programa. Este corre no meio-ecrã Shell, como se encontra representado na figura. À direita aparece a mensagem '<sem dados numéricos' porque o programa armazena uma lista vazia automaticamente.

    Á medida que vai recolhendo dados (agitar o micro:bit para lançar o dado), os valores das **somas** aparecem como pontos na aplicação Dados e Estatistica, à direita.

    Pressionando **[esc]** terminará o programa e pode fazer muitas coisas com a análide de dados do TI-Nspire.

    Agora, pressionando **[ctrl] [R]** novamente (no python Shell) irá correr novamente o programa.

    Dica: para limpar o Shell no início de cada recolha de dados adiciona o comando::

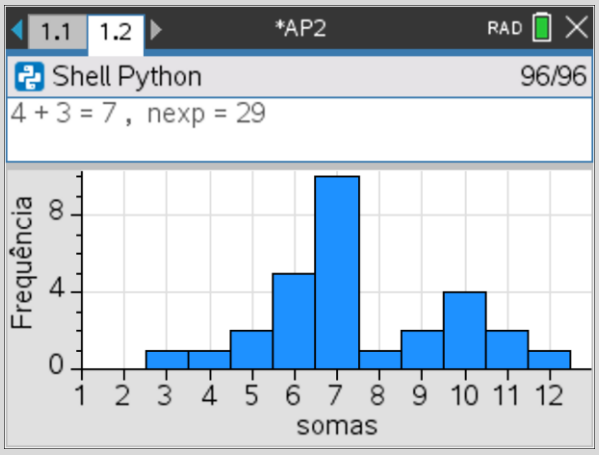
    clear_history()

    encontrado no **[menu] > Mais módulos > BBC micro:bit > Commands**

    no início do programa.

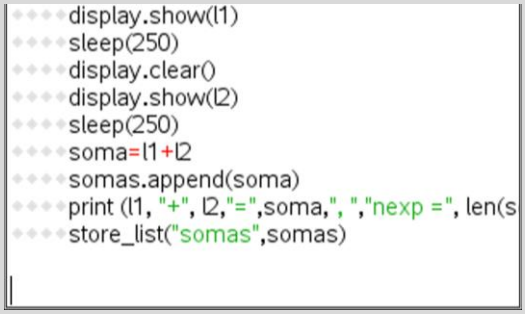
    Divirta-se e não se esqueças de gravar o documento!

Dica para o Professor: (repetido) O ecrã dividido da TI-Nspire demonstrado acima pode ser orientado vertical ou horizontalmente. Quando se utiliza o ctrl-4 para agrupar duas aplicações na mesma página, o modo default é vertical, como é possível observar na lição dos alunos. Para mudar para um layout horizontal pressione: [doc] > Esquema de página > Selecione esquema > Esquema 3. Também é possível ajustar a barra do separador para reduzir o tamanho da aplicação python Shell.



Para transformar o gráfico de pontos num histograma, pressione [menu] > Tipo de grágico e selecione 'Histograma'.

Solução:



Extensão Opcional: Exibir os pontos das faces do dado

É fácil desenhar imagens personalizadas para exibir no micro:bit. O código subsequente permite criar as seis faces do dado. Note que o **I** é maiúsculo em Imagem.

Escreva no primeiro bloco, one=Image(…, depois copie/cole/edite as seguintes cinco faces. Seguidamente, construa a lista de imagens do dado (dice_images), utilizando None para 0.

O Python intrepreta duas strings (conjunto de caracteres) escritas em linhas separadas sem delimitador (como por exemplo, uma vírgula) como sendo apenas uma única string, assim:

"aaa"
"bbb"

é igual a

"aaabbb"

No código abaixo, as 5 linhas do micro:bit encontram-se duplicadas na função Image( ), de modo a facilitar o design de uma imagem. O valor de cada digito na imagem encontra-se compreendido entre 0 e 9, de modo a controlar o brilho de cada LED.

```
#################################################################
from microbit import *
from random import *

one=Image(
"00000:"
"00000:"
"00900:"
"00000:"
"00000")
```

```
two=Image(
"00000:"
"09000:"
"00000:"
"00090:"
"00000")

three=Image(
"00000:"
"09000:"
"00900:"
"00090:"
"00000")

four=Image(
"00000:"
"09090:"
"00000:"
"09090:"
"00000")

five=Image(
"00000:"
"09090:"
"00900:"
"09090:"
"00000")

six=Image(
"00000:"
"09090:"
"09090:"
"09090:"
"00000")

faces=[None, one, two, three, four, five, six]
#     index:   0   1   2    3     4    5    6
print("running...")
while get_key() != "esc":
  if button_a.is_pressed():
    dado=randint(1,6)
    display.show(faces[dado])  # exibir uma das faces do dado usando pips
```