



**Unidade 5: Utilização da biblioteca TI System**

**Lição 1: Trabalhar com dados**

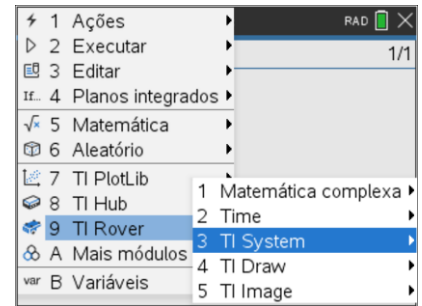
Nesta primeira lição da Unidade 5, pode aprender como usar a biblioteca **TI System** para importar ou exportar listas num programa em Python.

**Objetivos:**

- Importar e exportar listas.
- Revisitar os conceitos da Unidade 4 sobre representações gráficas.

A biblioteca ou módulo **TI System** usada sozinha ou associada a outras, permite comunicação bidirecional com a calculadora gráfica.

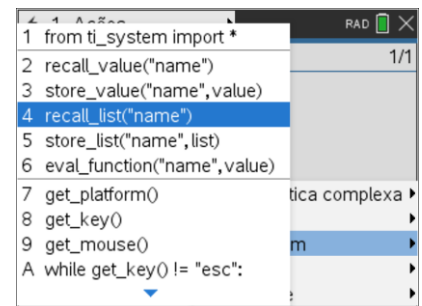
Para carregar esta biblioteca, pressionar **menu** depois **A Mais módulos** e finalmente **3 TI System**.



Nesta lição, vamos centrar a atenção sobre as instruções:

**4: var=recall\_list("name")** e **5: store\_list("name",list)**

As outras opções desta biblioteca serão abordadas nas próximas lições.

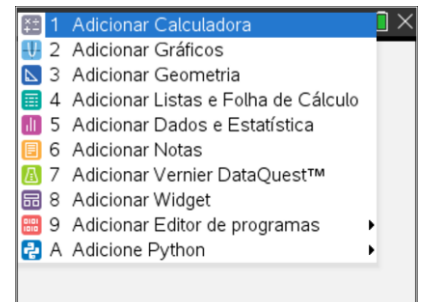


**1: Importar dados da calculadora.**

**a) Construir duas listas.**

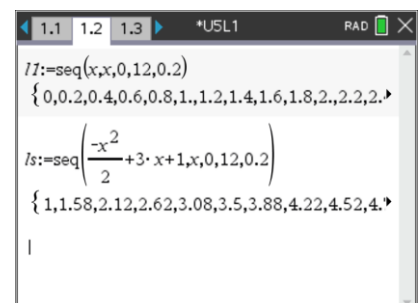
Em primeiro lugar, vamos criar apenas duas listas contendo os dados.

- Crie uma nova página de calculadora **ctrl** **1** depois escolha **1 Adicionar Calculadora**.
- O mesmo trabalho poderia ter sido realizado a partir da aplicação **Listas e Folha de Cálculo**.



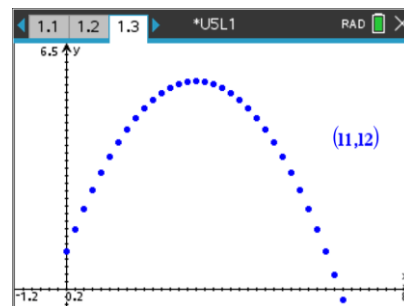
Crie uma lista **I<sub>1</sub>** com a sequência de números de 0 a 12, com passo de 0.2.

De seguida crie outra lista **I<sub>2</sub>**, das imagens de **x** (imagens dos dados da lista **I<sub>1</sub>**) pela função **f** definida por:  $x \mapsto -x^2/2 + 3x + 1$





- Adicione uma nova página com a aplicação **Gráficos**.
- Obtenha nessa página a representação gráfica da nuvem de pontos  $(I_1, I_2)$ , isto é, o diagrama de dispersão.



- Considere as seguintes definições da janela de visualização:  
 $X_{\min} = -1.2$  ;  $X_{\max} = 8$  ;  $Y_{\min} = -0.5$  e  $Y_{\max} = 6.5$ .

**b) Importar dados de um programa em Python.**

- Comece um novo script com o nome U5L1.
- A partir de importe a biblioteca **TI System**.
- Crie duas variáveis de listas (vazias) **abcs** e **orda**.
- Importe as bibliotecas **TI System** e **TI PlotLib** (não importa a ordem)
- Crie uma variável **abcs** e depois, a partir das opções da biblioteca **TI System**, escolha a opção **4: var=recall\_list("name")**. Como as abcissas estão na lista  $I_1$ , o campo "name" é preenchido com o nome da lista.
- Crie outra variável **orda** e proceda da mesma forma com a lista  $I_2$ .

```

from ti_system import *
import ti_plotlib as plt
abcs=[]
orda=[]
abcs=recall_list("I1")
orda=recall_list("I2")

```

**SUGESTÃO:**

A criação de listas vazias `abcs=[]` e `orda=[]` não é obrigatória, pois elas serão criadas quando as listas  $I_1$  e  $I_2$  forem chamadas. No entanto, é bom manter os bons hábitos aprendidos quando não se utilizar o módulo **TI System** que requer essa criação prévia.

- Execute o programa e depois verifique o conteúdo das variáveis **abcs** e **orda** pressionando a tecla .
- Chame a lista **abcs** e valide-a .
- Proceda da mesma forma com a lista das ordenadas (**orda**)

```

Shell Python 12/12
>>>from U5L1 import *
>>>abcs
[0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4, 3.6, 3.8, 4.0, 4.2, 4.4, 4.6, 4.8, 5.0, 5.2, 5.4, 5.6, 5.8, 6.0, 6.2, 6.4, 6.6, 6.8, 7.0, 7.2, 7.4, 7.6, 7.8, 8.0, 8.199999999999999, 8.4, 8.6, 8.800000000000001, 9.0, 9.199999999999999, 9.4, 9.6, 9.800000000000001, 10.0, 10.2, 10.4, 10.6, 10.8, 11.0, 11.2, 11.4, 11.6, 11.8, 12.0]
>>>

```



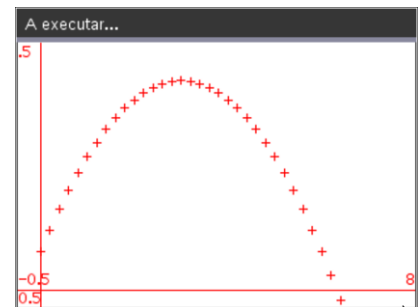


c) Representação gráfica.

Configure a representação conforme o ecrã ao lado.

```
*U5L1.py 13/13
abcs=[]
orda=[]
abcs=recall_list("1")
orda=recall_list("2")
# Representação gráfica
plt.cls()
plt.window(-0.5,8,-0.5,6.5)
plt.color(255,0,0)
plt.axes("on")
plt.scatter(abcs,orda,"+")
plt.show_plot()
```

Execute o programa **ctrl R**.



**2: Exportar dados.**

Crie um novo programa e designe-o por U5L11

**SUGESTÃO:**

Coloque o cursor no final duma linha e validar. Não importa a ordem da escrita da importação dos módulos.

- Crie uma função, nomeando-a por **data(a,b,n)**.
- Crie duas listas de dados a serem representados na forma de nuvem de pontos com valores no intervalo **[a; b]**, calculados com passo **n**.
- Na lista **y**, calcula-se a raiz quadrada dos valores da lista **x**.
- Para criar estas listas de dados podemos utilizar um ciclo **FOR**, naturalmente depois de criar duas listas vazias.

```
U5L11.py 10/11
from ti_system import *
from math import *
def data(a,b,n):
    x=[]
    y=[]
    for i in range(a,b,n):
        x.append(i)
        y.append(sqrt(x[i]))
    store_list("x",x)
    store_list("y",y)
```

**SUGESTÃO:**

A criação de duas listas vazias permite evitar uma mensagem de erro durante a execução do programa.



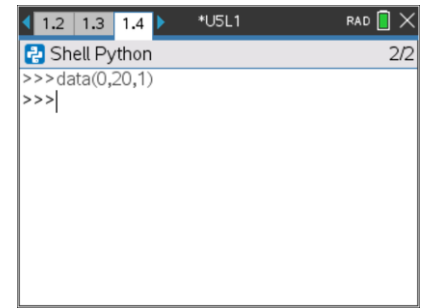


**NOTA:**

Atenção à indentação, as instruções `store_list` não podem estar dentro do ciclo.

Utilizar `del` para suprimir níveis de deslocamento.

- Execute o programa. Tomamos aqui 20 valores, de 0 a 20 com passo 1
- Saia do ambiente Python e exiba a representação gráfica das listas `lx` e `ly` (nuvem de pontos na aplicação Gráficos, por exemplo).



```
Shell Python 2/2
>>> data(0,20,1)
>>>|
```

**SUGESTÃO:**

A exportação de listas para a calculadora poderá ser particularmente interessante para representar dados recolhidos com sensores no **TI – Innovator & TI-Rover**.

