Unidade 5: Utilização da biblioteca TI System — Lição 1: Trabalhar com dados

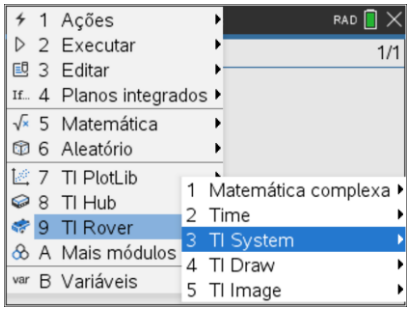
Nesta primeira lição da Unidade 5, pode aprender como usar a biblioteca **TI System** para importar ou exportar listas num programa em Python.

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.

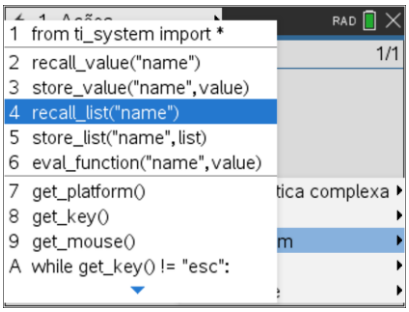
Para carregar esta biblioteca, pressionar menu depois **A Mais módulos** e finalmente **3 TI System**.



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.

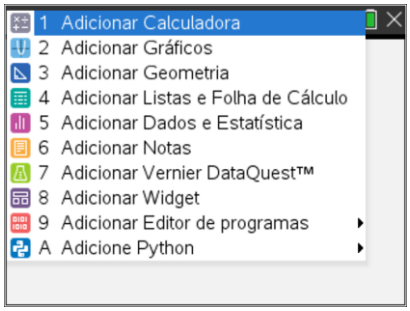


1: Importar dados da calculadora.

a) Construir duas listas.

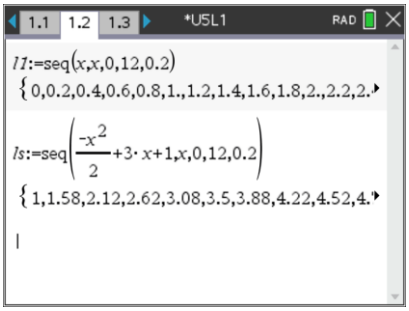
Em primeiro lugar, vamos criar apenas duas listas contendo os dados.

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



Crie uma lista l_1 com a sequência de números de 0 a 12, com passo de 0.2.

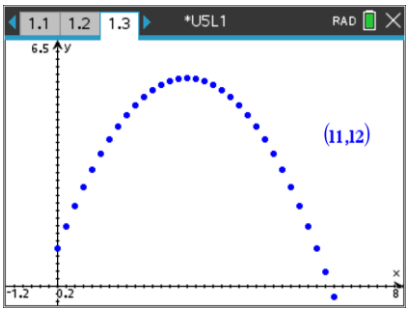
De seguida crie outra lista l_2, das imagens de x (imagens dos dados da lista l_1) pela função f definida por: $x \mapsto -x^2/2 + 3x + 1$



Disponibilizado sob a Licença Pública Creative Commons: Atribuição – Não Comercial – Compartilha Igual 3.0 Portugal.
https://creativecommons.org/licenses/by-nc-sa/3.0/pt/

- Adicione uma nova página com a aplicação **Gráficos**.

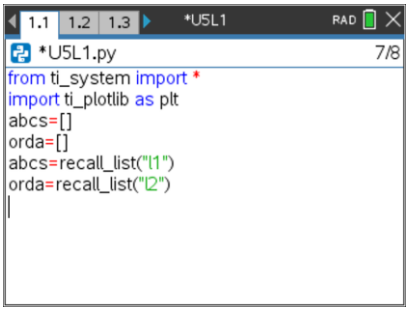
- Obtenha nessa página a representação gráfica da nuvem de pontos (l_1, l_2), isto é, o diagrama de dispersão.



- 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.

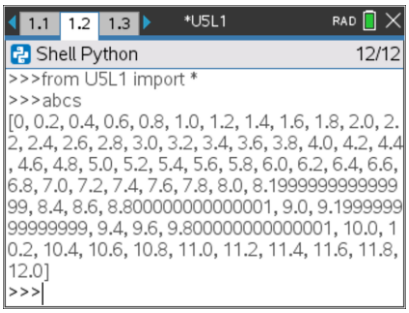


- Comece um novo script com o nome U5L1.
- A partir de menu importe a biblioteca **TI System**.
- Crie duas variáveis de listas (vazias) **abcs** e **orda**.
- Importe as bibliotecas **TI System** e **TI PlotLib** (não importa a ordem)
- Crie uma variável **abcs** e depois, a partir das opções da biblioteca **TI System**, escolha a opção **4: var=recall_list("name")**. Como as abcissas estão na lista l_1, o campo "name" é preenchido com o nome da lista.
- Crie outra variável **orda** e proceda da mesma forma com a lista l_2.

SUGESTÃO:

A criação de listas vazias abcs=[ ] e orda=[ ] não é obrigatória, pois elas serão criadas quando as listas l_1 e l_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 var.
- Chame a lista **abcs** e valide-a enter.
- Proceda da mesma forma com a lista das ordenadas (**orda**)

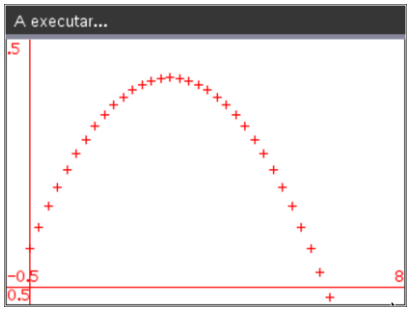


Disponibilizado sob a Licença Pública Creative Commons: Atribuição – Não Comercial – Compartilha Igual 3.0 Portugal.
https://creativecommons.org/licenses/by-nc-sa/3.0/pt/
© 2020 Texas Instruments

c) Representação gráfica.

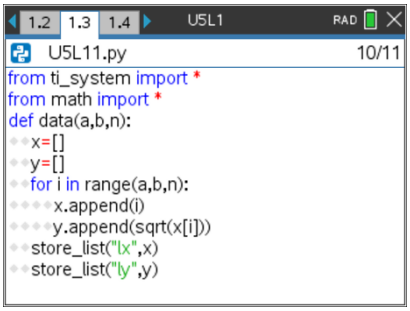
Configure a representação conforme o ecrã ao lado.

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.

SUGESTÃO:

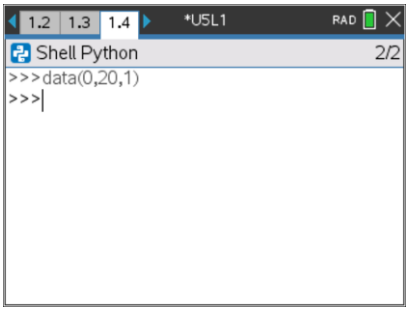
A criação de duas listas vazias permite evitar uma mensagem de erro durante a execução do programa.

Disponibilizado sob a Licença Pública Creative Commons: Atribuição – Não Comercial – Compartilha Igual 3.0 Portugal.
https://creativecommons.org/licenses/by-nc-sa/3.0/pt/

NOTA:

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

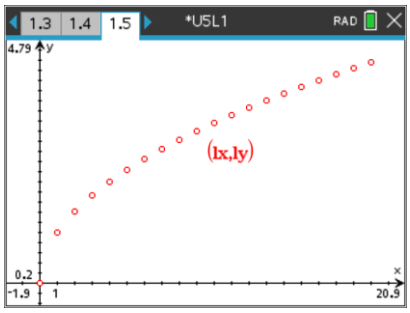
Utilizar del para suprimir níveis de deslocamento.



- 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).

SUGESTÃO:

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



Disponibilizado sob a Licença Pública Creative Commons: Atribuição – Não Comercial – Compartilha Igual 3.0 Portugal.
https://creativecommons.org/licenses/by-nc-sa/3.0/pt/

Unidade 5: Utilização da biblioteca TI System | **Lição 2: Modelação**

Nesta segunda lição da unidade 5, pode aprender como importar os resultados de um modelo utilizando a biblioteca **TI System**.

Objetivos:

- Efetuar modelação linear.
- Importar os resultados desta modelação num programa em Python.

Nesta lição trabalhará com um modelo linear a partir de dados inseridos previamente nas listas da calculadora. De seguida, vai escrever um programa para importar os dados desta modelação para os utilizar numa representação gráfica, numa interpolação, numa extrapolação …

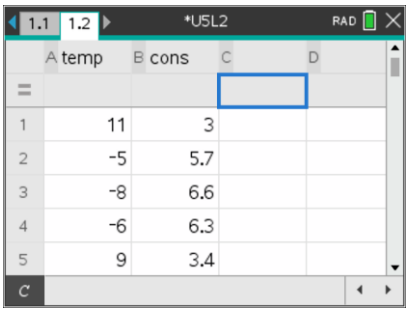
O PROBLEMA:

Num determinado dia, o pico de consumo de eletricidade é atingido por volta das 19 h. A nível nacional, é registado um pico de consumo de 96 350 megawatts às 19h02 do dia 15 de dezembro de 2019, uma quarta feira. Pretende fazer uma previsão dos picos de consumo no fim de semana seguinte para a zona abrangida pela central elétrica.

Para estabelecer essa previsão, tem dez leituras de consumo feitas às 19h, em função da temperatura, conforme apresentado na tabela abaixo.

T(°C)	11	-5	-8	-6	9	14	4	-1	-12	3
MW	3	5.7	6.6	6.3	3.4	2.7	4	5.1	7.1	4.6

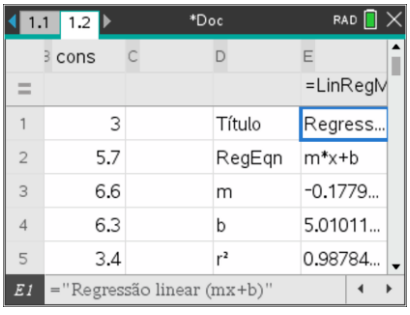
- Os dados são inseridos em listas da calculadora, a temperatura em **temp** e o consumo em MW em **cons**.
- Para tal, abra uma aplicação de **Listas e Folha de Cálculo**, e de seguida introduza os dados.



NOTA:

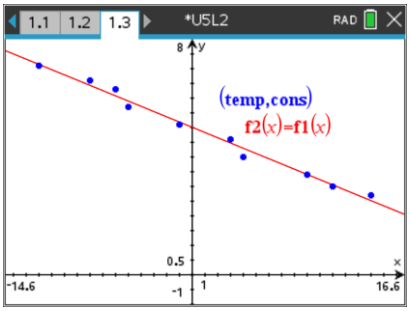
É também possível utilizar a aplicação **Dados e Estatística**, mas também a definição de listas numa aplicação **Calculadora**.

- Efetuar uma regressão linear do tipo mx+b (menu depois **4 Estatística**).



Disponibilizado sob a Licença Pública Creative Commons: Atribuição – Não Comercial – Compartilha Igual 3.0 Portugal.
https://creativecommons.org/licenses/by-nc-sa/3.0/pt/

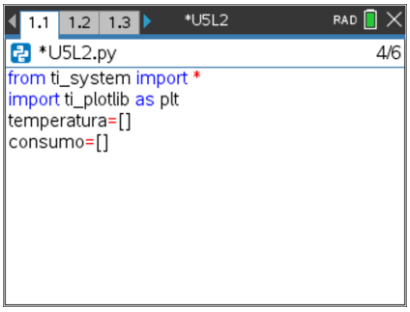
A fórmula permite prever o consumo em função da temperatura, que é $C = -0.18 \times t + 5.01$.



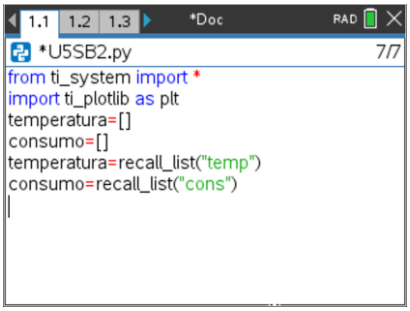
Representar numa aplicação **Gráficos**, a nuvem de pontos, bem como a reta de regressão.

Utilização dos resultados da modelação num programa em Python.

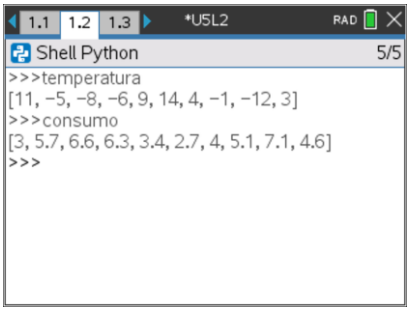
- Começar um novo programa, nomeando-o como U5L2.
- Incorporar o menu **TI System** e **TI PlotLib**.
- Criar duas listas vazias, **temperatura** e **consumo**.



Atribuir o conteúdo das listas **temp** e **cons** nas respetivas listas novas.
A instrução **var=recall_list("name")** está acessível no menu do módulo **TI System** (ver lição 1 da unidade 5).



Testar o programa e solicitar a exibição das variáveis pressionando a tecla var, escolhendo depois a variável pretendida.

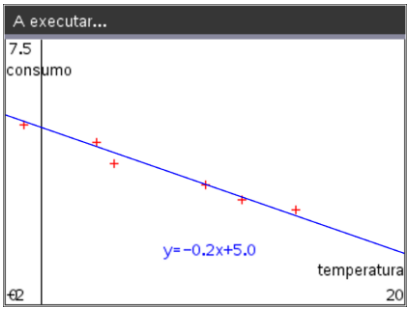


Para concluir esta lição e revisitar as lições da unidade 4, pode efetuar a representação gráfica das medições, bem como o modelo de regressão obtido.

Disponibilizado sob a Licença Pública Creative Commons: Atribuição – Não Comercial – Compartilha Igual 3.0 Portugal.
https://creativecommons.org/licenses/by-nc-sa/3.0/pt/

SUGESTÃO:

A instrução **lin_reg(xliste, yliste, "marca", row)** comporta um parâmetro suplementar **row** dando a possibilidade de colocar sobre outra linha a equação de regressão. Por defeito, é colocada na linha 11.



MAIS ALÉM:

Estimar o consumo de energia para uma determinada temperatura.

Irá recuperar os coeficientes da reta **m** e **b** para os utilizar numa função que permitirá realizar uma estimativa do consumo de eletricidade quando se sabe a temperatura.

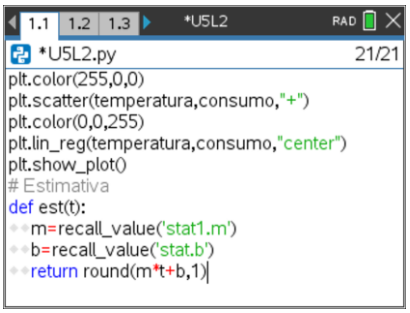
Definir uma função **est(t)** que dará o consumo estimado para uma temperatura **t**.

Executar o programa e efetuar alguns testes com diferentes temperaturas.

Lembre-se que o modelo apresentado representa uma estimativa de consumo às 19h02 em função da temperatura.

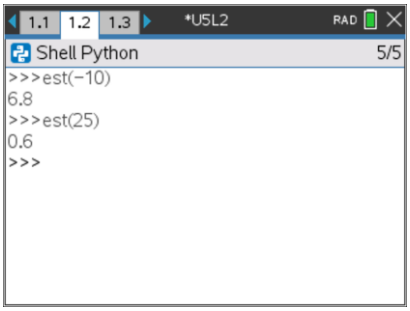
Pode então determinar os limites de validade do modelo.

- A instrução **recall(value, "name")** permite recuperar um programa, uma variável definida ou declarada numa aplicação.
- Atenção: as variáveis estatísticas têm um nome do tipo **statn°.nome.** Para as encontrar, pressione a tecla var quando a aplicação estiver ativa.



Exemplo:

Estimar o consumo de eletricidade em MW a uma temperatura de -10°C depois de 25°C.



Disponibilizado sob a Licença Pública Creative Commons: Atribuição – Não Comercial – Compartilha Igual 3.0 Portugal.
https://creativecommons.org/licenses/by-nc-sa/3.0/pt/
© 2020 Texas Instruments

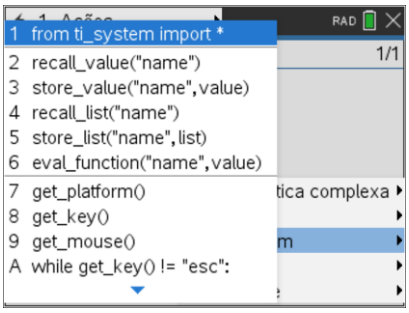
Unidade 5: Utilização da biblioteca TI System — Lição 3: Exibição e temporização

Nesta terceira lição da unidade 5, poderá aprender como utilizar as opções de exibição e de temporização da biblioteca **TI System** do Python.

Objetivos:

- Compreender o funcionamento das instruções **disp…**
- Utilizar estas instruções num programa, para além dos outros módulos.

Nas primeiras duas lições desta unidade aprendeu como importar/exportar listas de dados e a trabalhar com uma equação de regressão.
A biblioteca TI System tem outras opções eventualmente úteis, que nos propomos descobrir aqui.

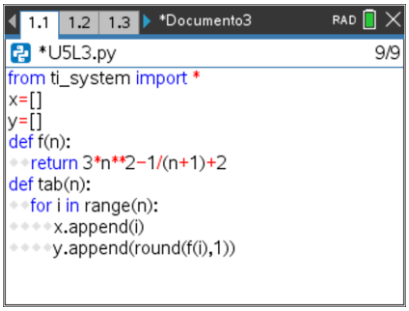


1: A instrução clear_history

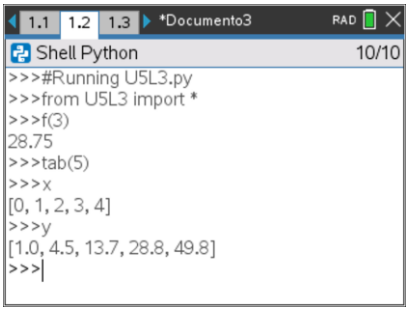
- Iniciar um novo programa com o nome U5L3.
- Importar o módulo **TI System**.

- Escrever um programa que defina uma tabela de uma função.

$$x \to 3x^2 - \frac{1}{x+1} + 2$$

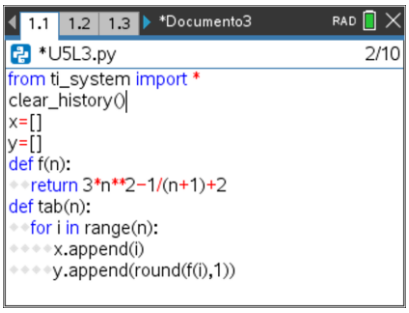


- Lembrar o programa U5L3 e executar. Deverá obter o ecrã ao lado.

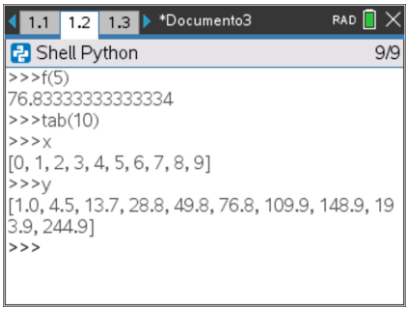


Disponibilizado sob a Licença Pública Creative Commons: Atribuição – Não Comercial – Compartilha Igual 3.0 Portugal.
https://creativecommons.org/licenses/by-nc-sa/3.0/pt/
© 2020 Texas Instruments

A instrução **clear_history** limpa a tela e coloca o cursor para início no topo do ecrã. Obtém assim uma nova tela, sem os resultados de execuções anteriores.



Assim, a execução do programa torna-se mais legível.

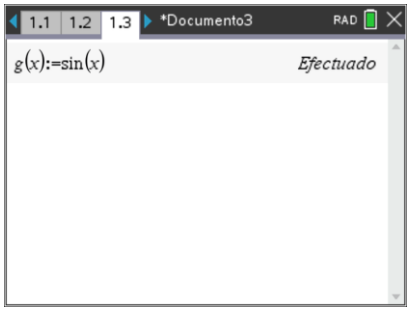


2: A instrução eval_function()

- Inserir uma página de Calculadora [ctrl] [I] e definir a função:

$$g: x \to sin(x)$$

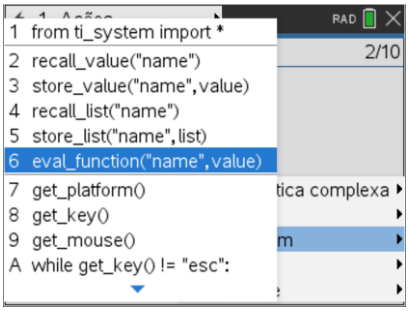
- A instrução **eval_function()** permitirá a utilização num programa em Python de uma instrução previamente definida noutra aplicação da TI-Nspire™ (Calculadora, Gráficos, Listas e Folha de Cálculo…).



Retome o programa e modifique-o como no ecrã para poder tabelar a função g: e depois representar graficamente a nuvem de pontos correspondente ao intervalo $[0\ ;\ 2\pi]$.

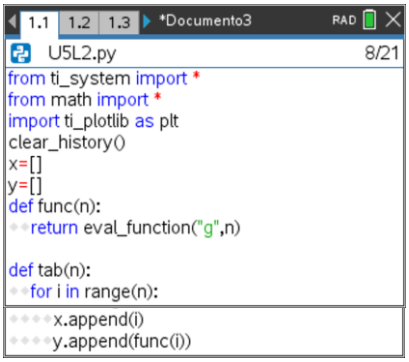
A instrução **eval_function** encontra-se no menu **TI System**, mas atenção à sintaxe.

eval_function("name",value) : o argumento name será aqui **g** e não g(x)

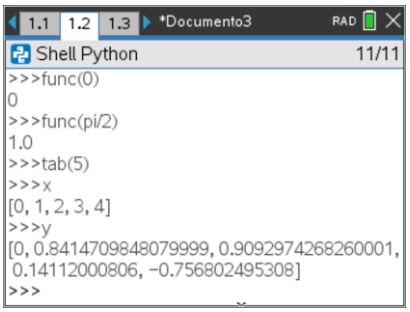


Disponibilizado sob a Licença Pública Creative Commons: Atribuição – Não Comercial – Compartilha Igual 3.0 Portugal.
https://creativecommons.org/licenses/by-nc-sa/3.0/pt/
© 2020 Texas Instruments

- Completar e modificar o programa anterior.



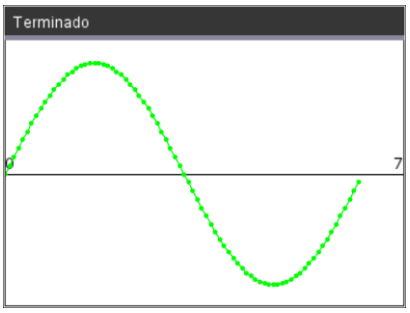
- Executar o programa. Atenção, a unidade é, por defeito, o radiano.
- Veja as opções da biblioteca Maths para uma expressão da medida dos ângulos em graus ou uma conversão. (Unidade 1, Lição 1).



- Modificar novamente o programa para obter as coordenadas de n pontos de abcissa no intervalo $[0\,;\,2\pi]$. Pode revisitar aqui o que aprendeu na unidade 4, lição 2.

- Terminar o programa, incluindo uma função **graf( )**.

- Verifique o funcionamento do programa.



NOTA:
A instrução **recall_value("name")** permitirá também colocar numa variável de um programa em Python o conteúdo de uma variável utilizada noutra aplicação da TI-Nspire™.

Disponibilizado sob a Licença Pública Creative Commons: Atribuição – Não Comercial – Compartilha Igual 3.0 Portugal.
https://creativecommons.org/licenses/by-nc-sa/3.0/pt/

Unidade 5: Utilização das bibliotecas TI System & TI PlotLib | **Aplicação: Estudo da queda livre**

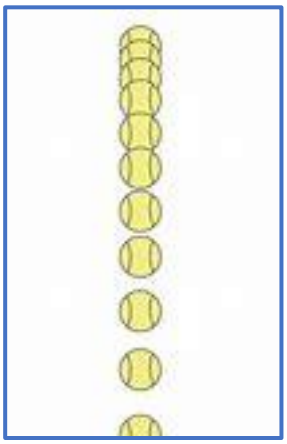
Nesta aplicação da unidade 5, pode revisitar as noções tratadas nas unidades 4 e 5 para criar um simulador que permita descrever um movimento.

Objetivos:

- Efetuar uma simulação de uma foto estroboscópica.
- Exportar os resultados para listas na calculadora.

Propõe-se, nesta aplicação relativa à unidade 5, utilizar a linguagem Python para criar um simulador de um fenómeno físico. Trataremos do estudo da queda livre:

- O programa deverá permitir a descrição de um movimento;
- A representação gráfica dos dados sob a forma de uma foto estroboscópica;
- Exportação dos dados para listas da calculadora.



O PROBLEMA:

Uma bola é lançada, sem velocidade inicial, de uma altura h.

As fotos são tiradas a cada 60 ms.

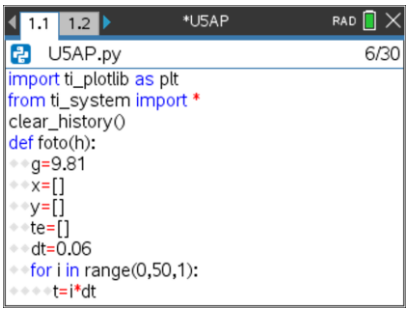
Vai escrever um programa que permita calcular, ao longo do tempo, a posição da bola, e depois fazer uma representação gráfica.

a) Cálculo das posições sucessivas

Durante a queda livre, a partir de uma posição inicial, e origem da marcação do tempo (y>0), a posição da bola pode ser calculada a partir da fórmula $y = -\frac{g}{2} \times t^2 + h_0$.

Note-se que: $g \approx 9{,}81$ m.s^{-2}**.**

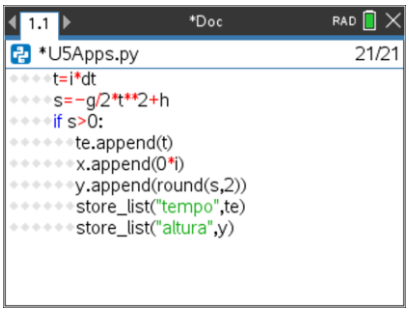
Criar um novo programa com o nome U5AP.



- Importar os módulos **TI System** e **TI PlotLib.**
- Limpar o ecrã.
- Criar uma função **foto(h)**, com a altura inicial como parâmetro, que forneça a posição da bola em função do tempo.
- Criar três listas vazias, abcissa **x**, ordenada **y** e tempo **te.**
- A ordenada é calculada a cada 60 ms.
- Os valores são guardados nas listas se y >0 (a bola não passa do chão).

Disponibilizado sob a Licença Pública Creative Commons: Atribuição – Não Comercial – Compartilha Igual 3.0 Portugal.
https://creativecommons.org/licenses/by-nc-sa/3.0/pt/

Criar um ciclo for para calcular a altitude da bola em função do tempo. Os resultados são expressos com precisão de 10^{-2} e armazenados nas respetivas listas. Como se pretende uma representação gráfica na forma de foto estroboscópica correspondente à realidade, a lista de abcissas será sempre 0. Finalmente, as listas **te** e **y** são exportadas respetivamente para as listas de **tempo** e de **altura** da calculadora.

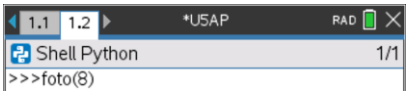


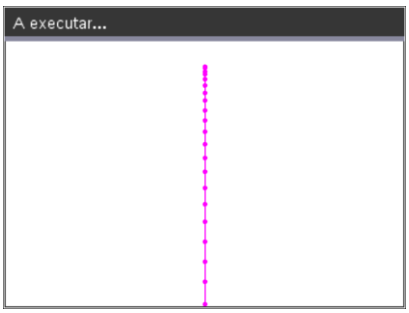
b) Representação gráfica

Configurar:

- Limpar o ecrã **plt.cls( )**.
- Exibir uma grelha de unidade 2 **plt.grid(xscl, yscl, type,(r,v,b))**.
- Definir uma janela para a representação **plt.window(x_{min}, x_{max}, y_{min}, y_{max})**.
- Definir a cor do ponto como magenta **plt.color(255,0,255)**.
- Representar a nuvem de pontos **plt.plot(x-list, y-list, mark)**.
- Exibir o gráfico **plt.show_plot( )**.

Executar o programa e pressionar a tecla [var], depois fornecer um valor para a altura inicial (8 m por exemplo).





A foto estroboscópica está representada.

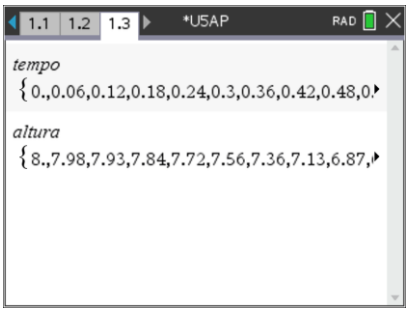
SUGESTÃO:

A partir da lista de posições da bola, pode calcular-se a velocidade da bola e exibir os respetivos vetores.

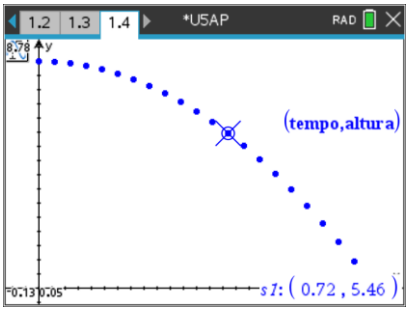
Disponibilizado sob a Licença Pública Creative Commons: Atribuição – Não Comercial – Compartilha Igual 3.0 Portugal.
https://creativecommons.org/licenses/by-nc-sa/3.0/pt/

c) Visualização dos dados exportados

- Sair do editor Python e exibir as listas.
- Inserir uma aplicação **Calculadora**.
- Obter as listas **tempo** e **altura**.



- Inserir uma aplicação **Gráficos** e representar a nuvem de pontos (**tempo**, **altura**).
- Utilizar a ferramenta **Traçar** para explorar a representação gráfica.



Disponibilizado sob a Licença Pública Creative Commons: Atribuição – Não Comercial – Compartilha Igual 3.0 Portugal.
https://creativecommons.org/licenses/by-nc-sa/3.0/pt/