

Atividades

Laboratório de Educação Digital



Guião 2 – Descobrindo... o HUB

Explorar, Aprender e Inspirar: O Poder dos LED e da TI-Nspire™ CX II-T

- Duração: 50 min;
- Dificuldade: Básico;
- Tema(s): STEM; Pensamento Computacional; Características do som; Altura e frequência do som; Relação entre música e matemática; Progressão geométrica.

1. Objetivos

- Realizar programas simples em Python;
- Desenvolver competências inerentes ao trabalho STEM;
- Compreender a relação entre frequência e altura de um som;
- Reconhecer regularidades matemáticas nas frequências das notas musicais.

2. Enquadramento

Projetos STEM com programação em Python e interfaces de entrada e saída promovem eficazmente o Perfil dos Alunos à Saída da Escolaridade Obrigatória. Através da programação, é possível explorar conceitos abstratos de forma concreta, aplicando matemática, física e lógica para resolver problemas reais.

A interatividade do TI-Innovator™ Hub torna a aprendizagem mais envolvente, despertando curiosidade e preparando os alunos para o uso prático, como manipulação de dados e controle de dispositivos físicos, incluindo sensores, alto-falantes e microcontroladores.

Neste caso, um projeto que gera sons com frequências programadas pode ajudar a compreender os fundamentos da acústica e da “matemática da música”, enquanto promove a experimentação e o pensamento crítico, competências cada vez mais essenciais na resolução de desafios tecnológicos com que se defrontam os profissionais da atualidade.

3. Material necessário

- Calculadora TI-Nspire™ CX II-T
- TI-Innovator™ Hub¹

¹ Para instruções detalhadas de configuração e atualização iniciais do equipamento, consultar <https://education.ti.com/pt/produtos/micro-controller/ti-innovator-rover/getting-started>



Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons

4. Para começar...

Vamos começar com a criação de um programa em Python, muito simples, que apenas vai produzir um som de frequência 440 Hz, durante 2 segundos.

Certifica-te que a calculadora está devidamente conectada ao Hub, utilizando o cabo mini USB-A / mini USB-B, tal como mostrado na figura 1:

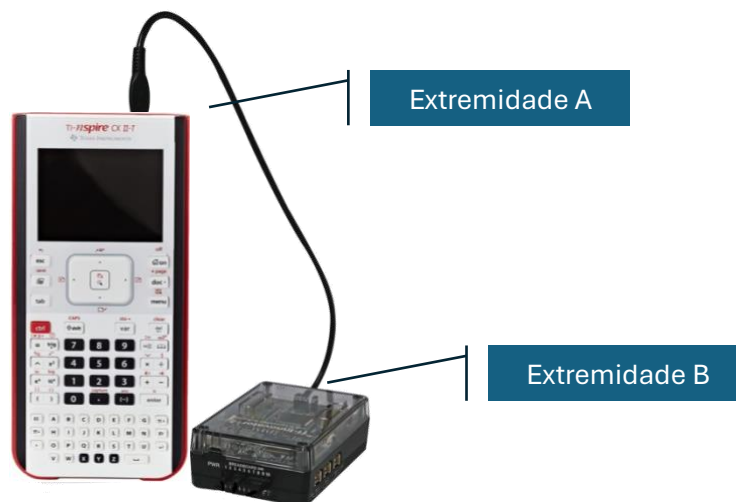
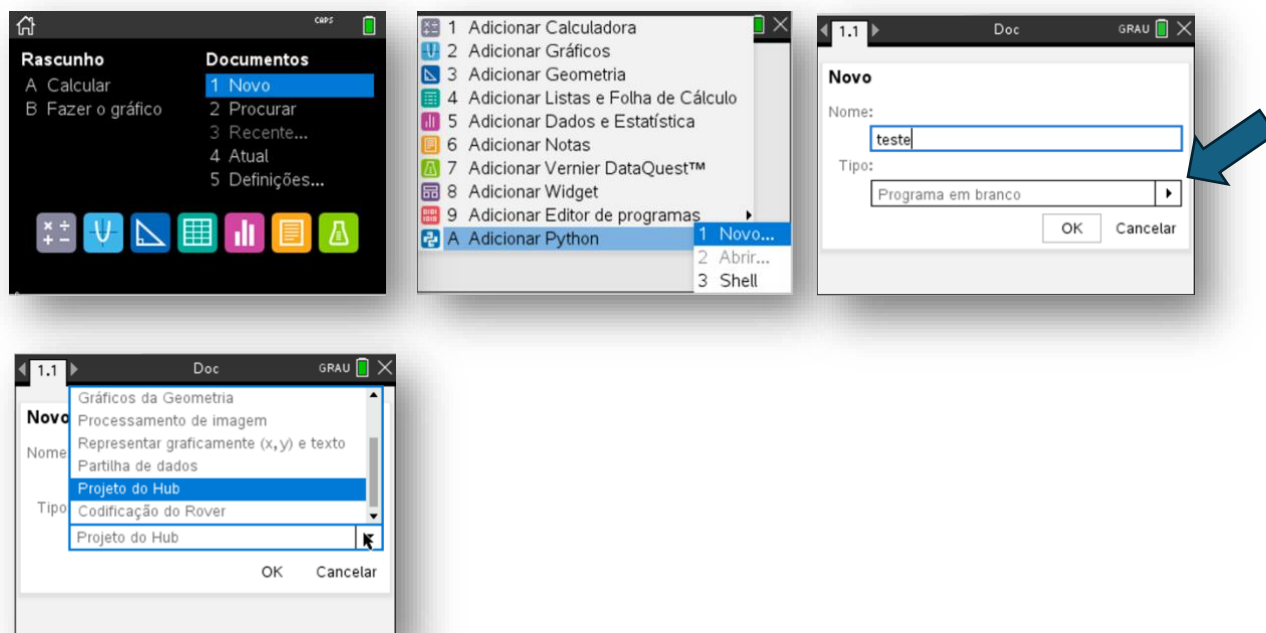


Figura 1 - Conexão do TI Innovator™ Hub à calculadora

Começando por pressionar o botão , adiciona uma página de Python, pré-configurada para projetos de Hub, utilizando os passos apresentados de seguida:



Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons

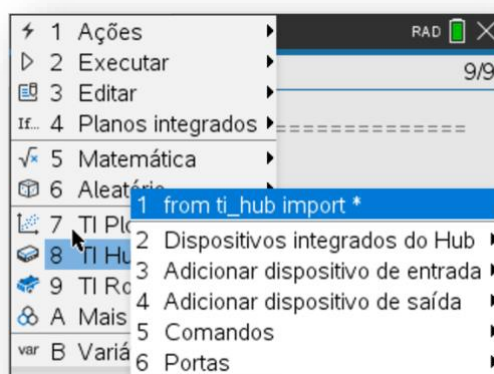
Deverás ter obtido um ecrã semelhante ao da figura:

```

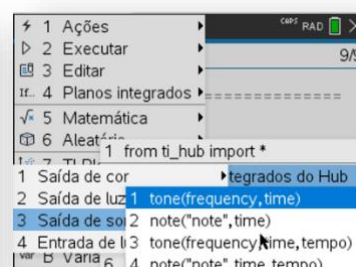
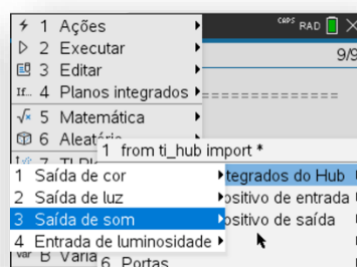
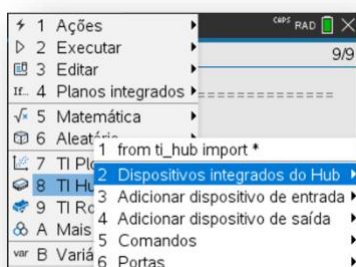
1.1 *Doc RAD 9/9
*teste.py
# Hub Project
#=====
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key
#=====

```

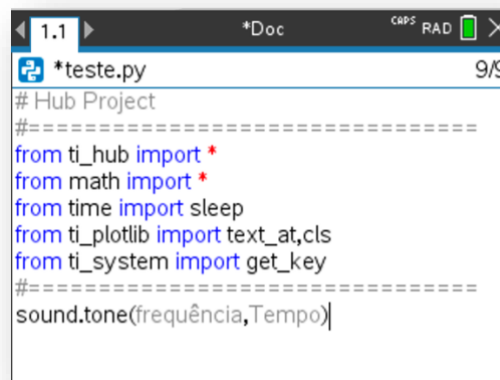
Todos os comandos necessários à construção de programas com o Hub podem ser facilmente acedidos a partir de  | 8 (TI Hub) :



Dado que o nosso primeiro programa tem como objetivo produzir um som de uma determinada frequência, e a emissão de som faz parte integrante do Hub (não sendo necessário conectar nenhum dispositivo adicional), podemos encontrar esse comando escolhendo 2 (Dispositivos integrados do Hub) | 3 (Saída de som) | 1 (tone(frequency,time)):



Deverás ter obtido um ecrã semelhante ao da figura:



```
1.1 *Doc CAPS RAD 9/9
*teste.py
# Hub Project
#=====
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at, cls
from ti_system import get_key
#=====
sound.tone(frequência,Tempo)|
```

Resta introduzir um valor numérico para a frequência (neste caso, 440), e para o tempo (neste caso, 2), e executar o programa utilizando **ctrl** **R**. O Hub deverá produzir um som de 440 Hz, durante 2 segundos.

Acabaste de programar um dispositivo eletrónico para produzir a nota musical (Lá) utilizando o Python!

Todos os outros programas podem ser realizados de forma semelhante, adicionando comandos adicionais que podes encontrar a partir de **menu** (não te esqueças de regressar à página anterior (1.1), clicando em **ctrl** seta de cursor para a esquerda.)

Os próximos desafios irão convidar-te a fazer isso.

5. Desafios

1. Agora que sabes programar o Hub, para produzir um som, consegues alterar o programa para que o som seja mais agudo, por exemplo, de frequência 4400 Hz?

Volta para a página de edição do programa (podes navegar para a página 1.1 clicando **ctrl** e seta para a esquerda).

Altera o programa para a frequência desejada e testa o programa com **ctrl** **R**.

Consegues perceber a diferença entre este som e o som inicial?

2. Modifica o programa para tocares sons de frequências diferentes (atenção, o máximo é 8000 Hz!) e testa várias vezes, para perceberes as diferenças entre sons graves (baixas frequências) e agudas (altas frequências).

3. E que tal tocar um pequeno trecho musical?

Vamos começar por um caso simples, as primeiras notas do Hino da Alegria, popularizado por Beethoven no séc. XIX.

As primeiras notas são:

si si dó ré ré dó si lá sol sol



Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons



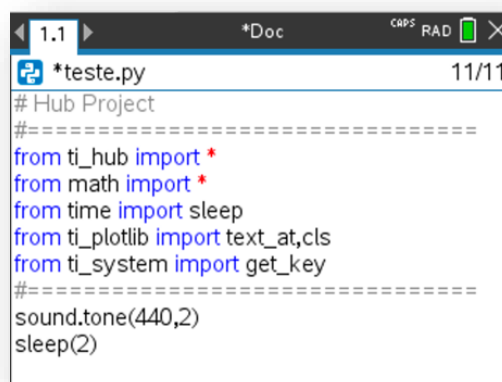
Considera que as notas que aparecem em duplicado podem ser tocadas apenas uma vez, com uma duração dupla das restantes notas, que aparecem só uma vez. Utiliza a seguinte tabela de frequências das notas musicais, para saberes as frequências dos sons que devem ser reproduzidos:

Nota	Frequência (Hz)
SOL	392,00
SOL#	415,30
LÁ	440,00
LA#	466,16
SI	493,88
DÓ	523,25
DÓ#	554,37
RE	587,33
RE#	622,25
MI	659,26
FÁ	698,46
FÁ#	739,99
SOL	783,99
SOL#	830,61
LÁ	880,00

Tabela 1 - Frequência das notas musicais

Nota: Para que a instrução de produção do som (nota musical) seguinte só seja enviado ao Hub após a produção do som correspondente à nota musical anterior, torna-se necessário adicionar um comando de espera (sleep()) com um tempo, em segundos, igual à duração da nota.

Por exemplo, a seguir ao comando `sound.tone(440,2)` deve ser adicionado, na linha seguinte, o comando `sleep(2)`. Este comando pode ser digitado diretamente no teclado ou acedido a partir de  | 8 (TI Hub) | 5 (Comandos) | 1 (sleep(seconds)). Neste caso seria obtido um ecrã semelhante ao seguinte:



```
1.1 | *Doc | CAPS RAD | X
*teste.py | 11/11
# Hub Project
#=====
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at, cls
from ti_system import get_key
#=====
sound.tone(440,2)
sleep(2)
```

Constrói agora de forma semelhante toda a sequência de notas (e respetivas frequências) correspondentes à música pretendida. Para facilitar o trabalho, podes selecionar as linhas que se repetem (premindo, sem soltar a tecla  e a seta de cursor para selecionar), copiar ( ) e colar ( )



Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons

4. Divide a frequência de cada nota que aparece na tabela 1 pelo valor correspondente à frequência da nota anterior (por exemplo, 415,30/392,00). Notas alguma regularidade? Discute com o teu professor o seu significado.

6. Conclusão

A relação constante entre as frequências de notas musicais consecutivas, que verificaste no ponto anterior, significa que a frequência das notas segue uma progressão geométrica.

Investiga como poderias modificar os teus programas para calcular a frequência de qualquer nota musical, usando uma expressão matemática. Esta expressão deverá ter como única variável um número inteiro, n , que representa o número de teclas de afastamento em relação à nota Lá (440 Hz) no teclado de um piano.



Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons

7. Para o professor

Sugestões de resposta aos desafios:

1.  *teste.py 10/11

```
# Hub Project
#=====
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key
#=====
sound.tone(4400,2)
```

2.  *teste.py 11/13

```
# Hub Project
#=====
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key
#=====
sound.tone(110,2)
|
```

3.  *teste1.py 9/23

```
#=====
sound.tone(493.88,2)
sleep(2)
sound.tone(523.25,1)
sleep(1)
sound.tone(587.33,2)
sleep(2)
sound.tone(523.25,1)
sleep(1)
sound.tone(493.88,1)
sleep(1)
sound.tone(440,1)
sleep(1)
sound.tone(392,1)
```

4. O aluno deverá notar que as frequências das notas musicais obedecem a uma progressão geométrica, de razão $\sqrt[12]{2}$.

Conclusão

$$f = 440 \times 2^{\frac{n}{12}}$$



Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons