

## Atividades

# Laboratório de Educação Digital



# Guião 5 – Descobrindo... Pack Placas de Ensaio

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

- Duração: 100 min;
- Dificuldade: Intermédio;
- Tema(s): STEM; Pensamento Computacional; Circuitos elétricos em série e em paralelo; Introdução à eletrónica; Síntese aditiva de cor;

### 1. Objetivos

- Desenvolver competências inerentes ao trabalho STEM;
- Aplicar conhecimentos de circuitos elétricos em série e em paralelo;
- Desenvolver competências básicas de circuitos eletrónicos;
- Desenvolver programas em Python para controlo de circuitos eletrónicos.

### 2. Enquadramento

A eletrónica é uma área fascinante e rica em possibilidades, onde conceitos teóricos se encontram com aplicações práticas, proporcionando uma compreensão mais profunda da tecnologia que nos rodeia. Um projeto didático e envolvente para quem se está a iniciar na eletrónica poderá ser a construção de um sistema de controlo de um LED RGB, utilizando resistências, uma placa de ensaios (*breadboard*) e um controlador eletrónico programado em Python, neste caso o TI Innovator™ Hub.

O LED RGB é composto por três LEDs (vermelho, verde e azul) que, quando combinados, podem produzir uma vasta gama de cores. A utilização de resistências é essencial para limitar a corrente e proteger os LEDs, enquanto a placa de ensaios facilita a montagem de vários circuitos de teste, dado que não seria adequado usar solda para unir os componentes.

Com algumas linhas de código em Python, será possível definir a intensidade de cada LED, explorando a interatividade entre o código e o circuito físico. Este tipo de projeto não só proporciona uma iniciação à eletrónica apoiada pela programação em Python, como também incentiva a criatividade, a resolução de problemas e as competências inerentes ao trabalho em metodologia STEM.



Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons

### 3. Material necessário

- Calculadora TI-Nspire™ CX II-T
- TI-Innovator™ Hub<sup>1</sup>
- Pack Placas de ensaio:

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

### 4. Para começar...

Este projeto vai utilizar a TI-Nspire™ CX-II T e o TI-Innovator™ Hub, conectado por portas eletrônicas a uma placa de ensaios (*breadboard*). Vamos utilizar alguns componentes eletrônicos, tais como resistências, LED's e fios de ligação, para uma verdadeira iniciação ao fascinante mundo da eletrônica.

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

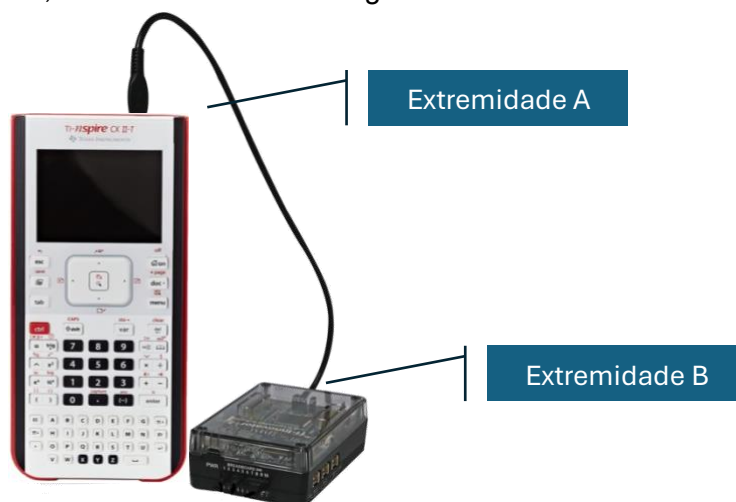


Figura 1 - Conexão do TI Innovator™ Hub à calculadora

O Hub possui vários pinos eletrônicos especialmente dedicados a testes de circuitos eletrônicos. Neste projeto, vamos usar portas de I/O digital (BB1 a BB10 na figura 2, linha de pinos superior), e portas de terra (linha de pinos inferior, os 8 mais à esquerda):

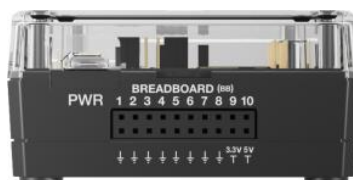


Figura 2 – Pinos eletrônicos do Hub,

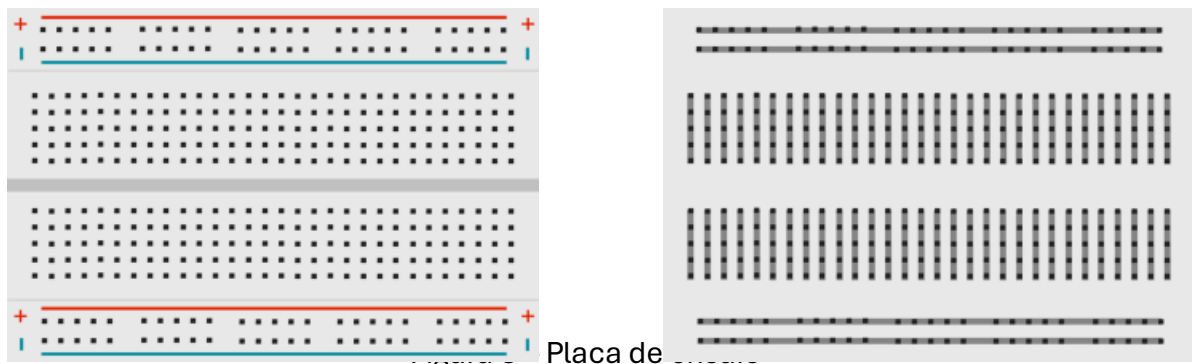


Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons



Para facilitar a conexão dos componentes eletrônicos pode usar-se uma placa de ensaio, permitindo ligar de forma organizada vários elementos, sem necessidade de uso de solda ou outro meio adicional de fixação. Podem ser usados cabos de ligação direta entre o Hub e a placa de ensaio, para alimentar componentes da placa de ensaio e controlar ou monitorizá-los usando um programa em Python.

Os pinos da placa de ensaio estão dispostos em grupos de cinco, conectados eletricamente entre si (na parte traseira da placa). Os trilhos de corrente nas partes superior e inferior estão assinalados a vermelho (+) e azul (-). A figura 3 mostra um esquema de uma placa de ensaio (à esquerda) e a interconexão entre os pinos (à direita)



Vamos começar por um circuito muito simples, constituído apenas por um LED, que vamos ligar muito brevemente e com baixa intensidade, já que não usaremos uma resistência para o proteger de um eventual excesso de corrente, como faremos de seguida.

Coloca um LED verde na placa de ensaio, tendo o cuidado de verificar que o pino mais comprido (+, ânodo) é ligado (através da placa) à porta BB1 do Hub e o mais curto (-, cátodo) é ligado (através da placa) a uma das portas terra do Hub. Repara que a placa poderia, neste caso, ser dispensada, ligando o LED diretamente aos pinos do Hub.

O teu primeiro circuito deverá assemelhar-se à figura 4 (nota que a diferente cor dos cabos tem como único propósito a sua fácil distinção):

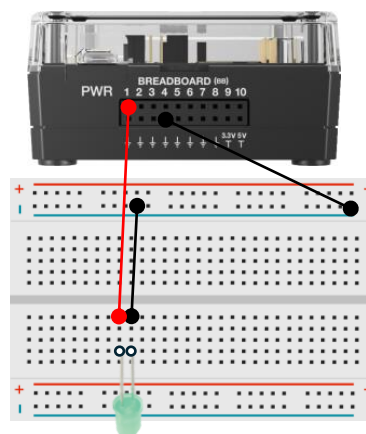


Figura 4– Esquema de montagem

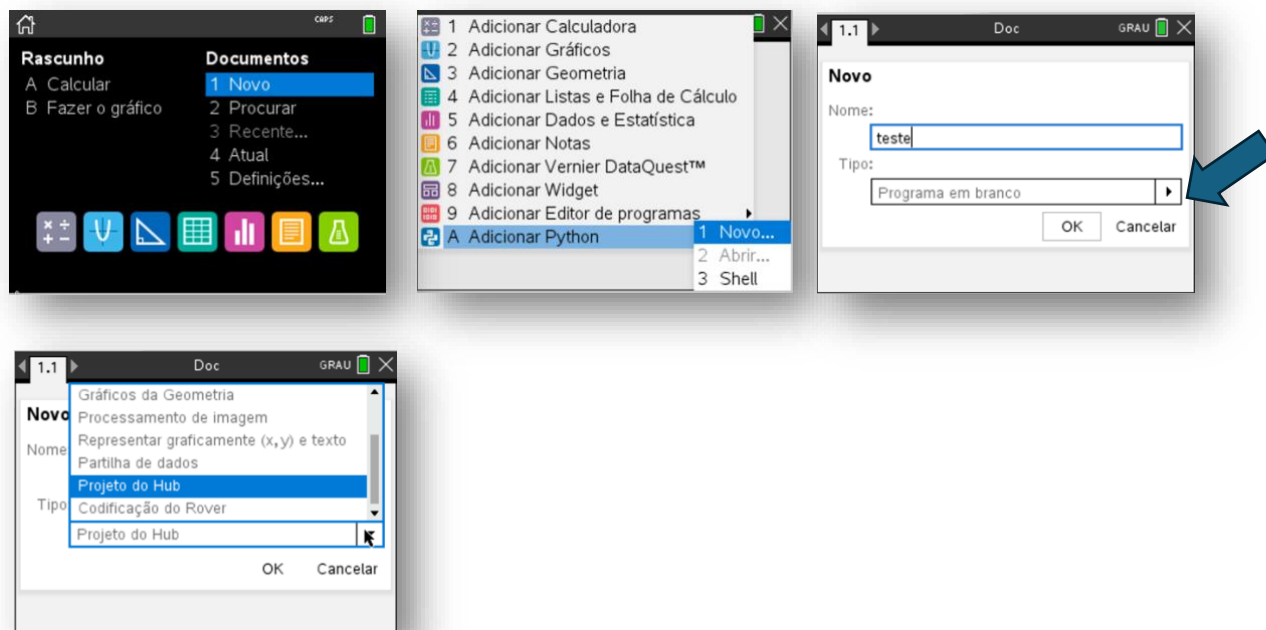


Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons

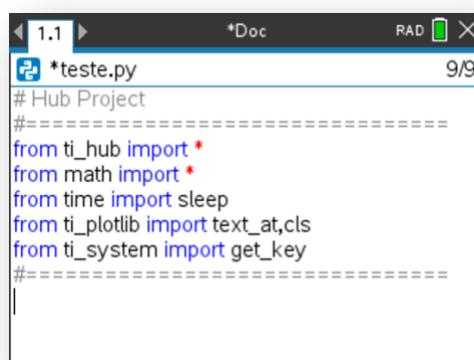


Vamos agora criar uma página em Python, pré-configurada para projetos com o Hub, que será utilizada para controlar o circuito eletrónico.

A partir de , segue os passos descritos de seguida:

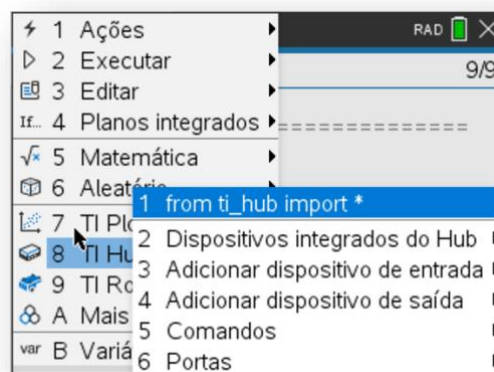


Deverás ter obtido um ecrã semelhante ao da figura:



Todos os comandos necessários à construção do podem ser facilmente acedidos a partir de

 | 8 (TI Hub) :



Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons

Para ativarmos o LED, vamos fazer corresponder o comando “saída analógica” ao pino ao qual o mesmo está conectado no Hub (neste caso, BB1, tal como na figura 4), selecionando 4 (Adicionar dispositivo de saída) | 7 (Analog Out).

Se optarmos por designar este objeto de led1, (após inserir a linha anterior) ao digitar led1 podemos escolher do menu de opções *set(value)* e colocar o valor 100, ou 200 (inferior ao máximo, que seria 255). Apesar desta precaução, para evitar queimar o LED, é conveniente desligá-lo ao fim de pouco tempo, adicionando um comando *sleep(1)* e *led1.off()*.

Deverás ter obtido um ecrã semelhante ao seguinte:

Testa o programa ( **ctrl** **R** ), e verifica se funciona como previsto. Caso o LED não acenda, é muito provável que o tenhas ligado ao contrário, bastando inverter os pinos + e -.

Acabaste de construir um circuito eletrónico, controlado por Python!

Os passos seguintes irão desafiar-te a fazer algo um pouco mais complexo.

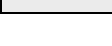


Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons

## 5. Desafios

1. O uso de um código de cores para resistências é uma prática essencial e altamente eficaz na identificação rápida e precisa do valor da resistência de um componente. Esta metodologia de codificação oferece várias vantagens, especialmente na produção e manutenção de dispositivos eletrônicos. Uma das causas está relacionada com o facto das resistências serem, muitas vezes, muito pequenas para que números ou letras sejam legíveis. As cores, por outro lado, são facilmente visíveis e interpretáveis, independentemente do tamanho do componente. Este sistema torna possível identificar resistências num relance, sem a necessidade de equipamentos adicionais ou leituras detalhadas.

A tabela seguinte mostra a codificação de cores utilizada na classificação das resistências.

|                                                                                     | Cor       | 1ª Faixa | 2ª Faixa | Multiplicador | Tolerância |
|-------------------------------------------------------------------------------------|-----------|----------|----------|---------------|------------|
|    | Preto     | 0        | 0        | ×1            |            |
|    | Castanho  | 1        | 1        | ×10           |            |
|    | Vermelho  | 2        | 2        | ×100          |            |
|    | Laranja   | 3        | 3        | ×1.000        |            |
|   | Amarelo   | 4        | 4        | ×10.000       |            |
|  | Verde     | 5        | 5        | ×100.000      |            |
|  | Azul      | 6        | 6        | ×1.000.000    |            |
|  | Violeta   | 7        | 7        | ×10.000.000   |            |
|  | Cinzentos | 8        | 8        |               |            |
|  | Branco    | 9        | 9        |               |            |
|  | Dourado   |          |          |               | ±5%        |
|  | Prateado  |          |          |               | ±10%       |

Para determinar o valor de uma resistência utilizando a tabela de cores, devem ser identificadas as faixas de cores na mesma. Estas faixas de cores geralmente representam os dígitos significativos da resistência e o multiplicador. As duas primeiras faixas representam os dígitos significativos, enquanto a terceira faixa é o multiplicador. A última faixa indica a tolerância da resistência.

Por exemplo, se uma resistência tem as cores castanho, preto e vermelho, o valor da resistência é calculada da seguinte forma: Castanho representa 1, preto representa 0, e vermelho representa o multiplicador de 100. Portanto, a resistência é 10 (dígitos significativos) vezes 100 (multiplicador), resultando em 1.000 ohms ou 1 kΩ. Se a quarta faixa for dourada, a tolerância é de ±5%.

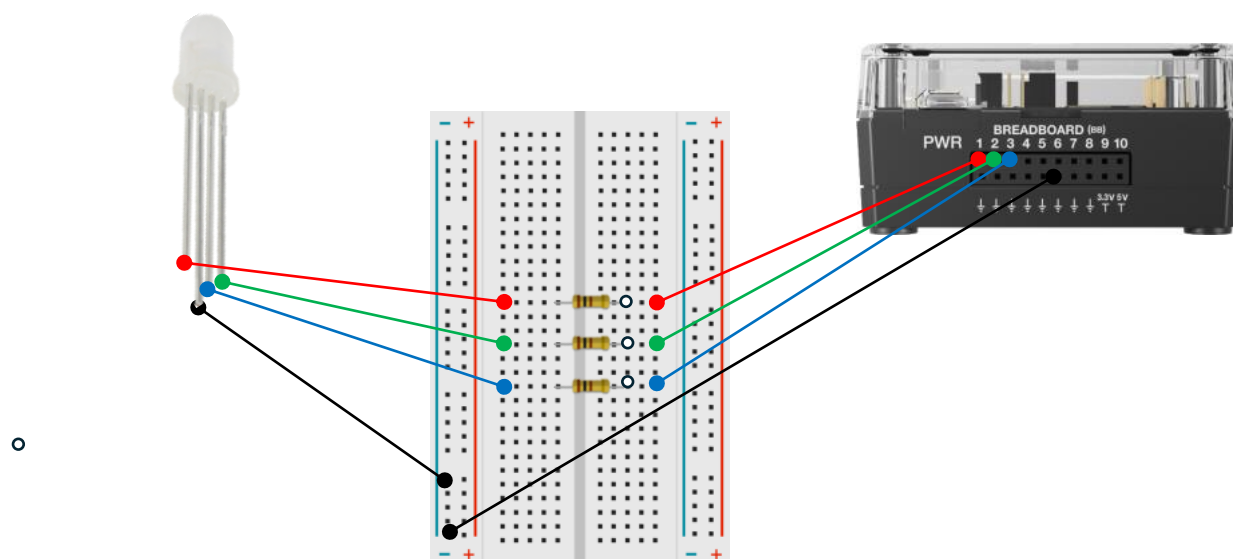
Consegues agora identificar, no pack, as resistências de 100 Ohm? Vamos precisar de 3.



Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons



2. Vamos agora usar um LED RGB (efetivamente composto por um LED vermelho, um verde e um azul, que iremos controlar de forma individual), para produzir luz de diferentes cores. Vamos ligar cada um dos pinos do LED (correspondentes a uma cor) a um pino BB diferente do Hub. O pino mais longo do LED deve ser ligado a uma porta terra no Hub. Vamos também proteger cada um destes LED por uma resistência de 100 Ohm, ligada em série com estes. Efetua as ligações conforme indicado na figura:



Procede de forma semelhante ao desafio anterior e adapta o programa em Python para configurar a ligação individual de cada um dos LED's do LED RGB, aos pinos BB1, BB2 e BB3 e ligá-los, de forma independente e sequencial, durante 2 segundos cada um.

## 6. Conclusão

No âmbito deste projeto de verdadeira iniciação à eletrónica, foi desenvolvido um sistema que utilizou um LED RGB, várias resistências e um microcontrolador programado em Python. A sua implementação envolveu a criação de circuitos simples, em série e em paralelo, com resistências para limitar a corrente, e a programação do TI-Innovator™ Hub para gerar diferentes padrões de iluminação.

A síntese aditiva de cor com um LED RGB baseia-se na combinação de luzes de diferentes cores primárias (vermelho, verde e azul) para criar uma ampla gama de cores. Num LED RGB, cada um desses díodos emissores de luz pode ser controlado de forma independente, permitindo que a intensidade de cada cor seja ajustada. Quando as cores vermelho, verde e azul são combinadas em várias proporções, diferentes cores são produzidas, incluindo



Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons

branco quando todas as três cores são misturadas na mesma intensidade. Esse princípio é amplamente utilizado em telas de dispositivos eletrônicos, como televisões e monitores, para reproduzir imagens coloridas de forma bastante precisa. A este processo denomina-se *síntese aditiva de cor*<sup>1</sup>.

Com base na informação anterior, consulta uma tabela de codificação de cor RGB e altera o programa para que o LED RGB fique iluminado com qualquer cor, à escolha do utilizador.

---

<sup>1</sup> Em contraste com a *síntese subtrativa de cor* que se obtém pela mistura de tintas.



Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons

## 7. Para o professor

Sugestões de resposta aos desafios:

1. Uma resistência de 100 Ohm pode apresentar uma risca castanha (1) uma preta (0) e uma castanha (x10), semelhante à da figura (a última risca – dourada - está relacionada com a tolerância, que para já não vamos considerar). 

2.

```
*teste.py 8/28
#=====
ledR=analog_out("BB 1")
ledG=analog_out("BB 2")
ledB=analog_out("BB 3")

ledR.on()
sleep(2)
ledR.off()
ledG.on()
sleep(2)
ledG.off()
ledB.on()
sleep(2)
ledB.off()
```

## Conclusão

```
*LED5_RGB 9/27
1.1
ledR=analog_out("BB 1")
ledG=analog_out("BB 2")
ledB=analog_out("BB 3")
# por exemplo, para a cor "laranja-outono"
ledR.set(255)
ledG.set(102)
ledB.set(51)
sleep(2)
ledR.off()
ledG.off()
ledB.off()
```



Este recurso pedagógico é disponibilizado com a respetiva licença Creative Commons