



Neste **guião** vais continuar a aprender **Arduino**. Inicialmente, vamos utilizar **botões** para fazer **ligar** ou **desligar** um **LED** e posteriormente iremos utilizar o **teclado** do **computador** para **ligar** determinados **LEDs**.

Todas as prints/recortes pedidos ao longo deste guião devem ser **colados** no **documento Diário de Bordo**. Adiciona um **título** ao **final** do teu **documento** de nome: **Guião 6 botões**

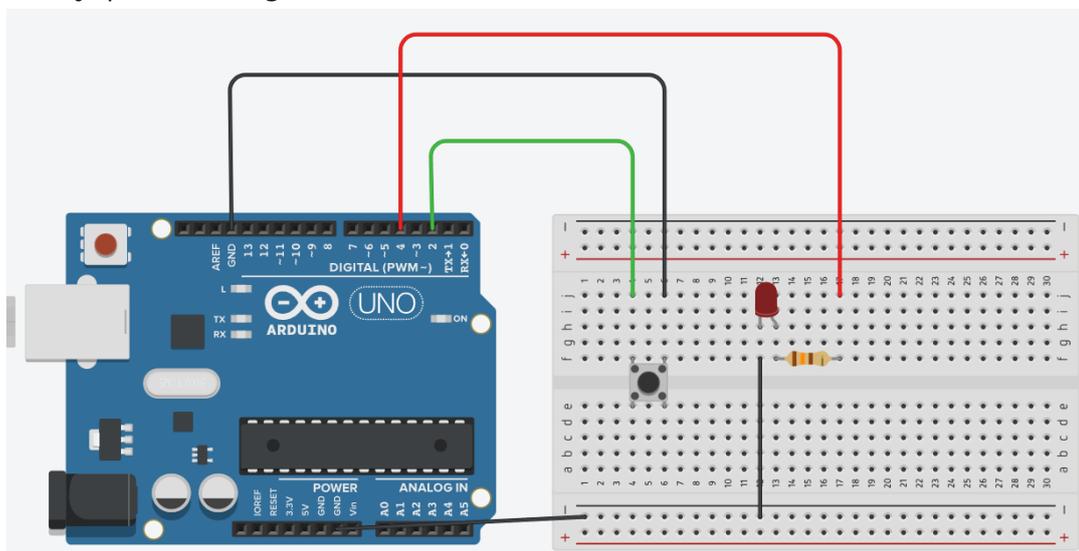
Accede a <https://www.tinkercad.com/> e **entra** na tua **conta**.

Na tua **conta** **escolhe**, no menu do lado **esquerdo**:

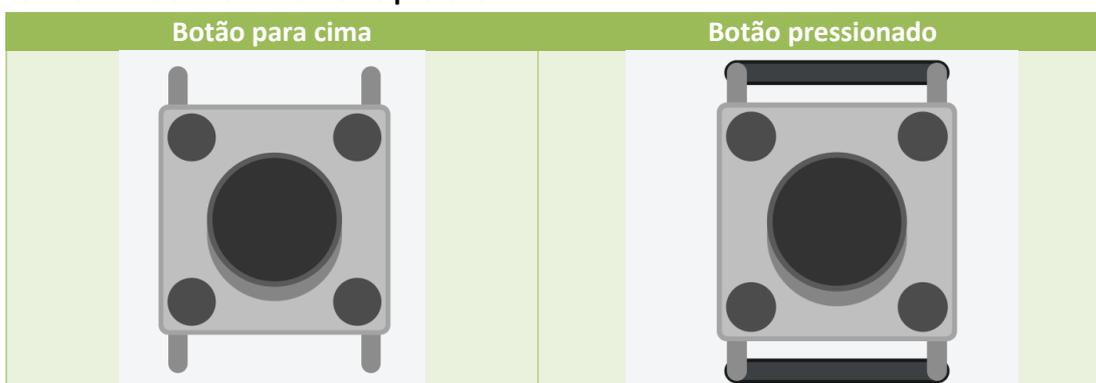


Para **criares** um novo **circuito**.

Começa por **criar** o seguinte **circuito**:



Como **funciona** este **botão** de **pressão**?



Como podes ver ao **pressionar** o **botão** são os **pinos** que estão **lado a lado** que **curto circuitam**.

Vamos ao **pretendido** para o **exercício**. Neste caso ao **pressionar** o **botão** o **LED** deve **apagar** e em **situação normal** (sem estar **pressionado**) deve **estar ligado**. Parece **estranho** por estar ao **contrário**, mas **pensa** num **alarme** de **casa**. Quando a **porta fecha** **pressiona** o **botão** e a



alarme não liga. Mas se alguém abrir a porta o botão deixa de estar pressionado pela porta e toca o alarme.



Abre o .

Com o que aprendeste até agora define os pinos utilizados no circuito chamando ao pino do LED, `ledPin` e ao pino do botão `btnPin`

Na função `setup()` terás de indicar se os pinos definidos são **INPUT** ou **OUTPUT**. Para o LED terás de ser tu a escrever a linha de código, mas para o botão deixo-te o que deves escrever:

```
pinMode(btnPin, INPUT_PULLUP);
```

INPUT_PULLUP?? Estranho! Pois o que acontece aqui é que o botão ao ser pressionado como está ligado ao **GND** passará o estado **LOW** para o pino digital 2 ou seja um zero. Mas que valor envia ele quando não está pressionado?? Pois esse é o problema, nenhum valor em concreto. Para que ele tenha um valor nessas situações ele tem de estar definido a **INPUT_PULLUP**.

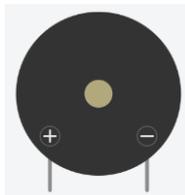
Para confirmares o que expliquei anteriormente, deves inicializar o monitor série, ainda dentro da função `setup()`. De seguida na função `loop()` acrescenta o código que te permite ver no monitor série o valor que está a ser recebido pelo pino digital 2 quando o botão está pressionado ou não.

Ajuda: deves utilizar a função `digitalRead(btnPin)` para leres do pino 2 o valor. Até agora escrevíamos nos pinos porque eram OUTPUTS, mas como o botão é INPUT nós queremos ler e não escrever.

Inicia a simulação e verifica que no monitor série aparece um 1 quando o botão não está pressionado e um zero quando o pressionas.

Agora terás de utilizar uma estrutura de decisão **if...else** para que caso o botão não esteja pressionado o LED vermelho esteja ligado e caso contrário se desligue.

Desafio: Adiciona um Buzzer que aqui tem o nome de Piezo.



Para tocar deve ser ligado entre o **GND** e uma porta digital, neste caso o pino 7.

No código define o pino 7 como `buzzerPin` e considera-o como **OUTPUT**.



TIN
KER
CAD Vamos ao seu **funcionamento**. Ele **toca tons**. Neste caso vamos tocar um **Dó** que tem o código **262**. Podes **conseguir** isso com a **seguinte instrução**:

```
tone(buzzerPin,262);
```

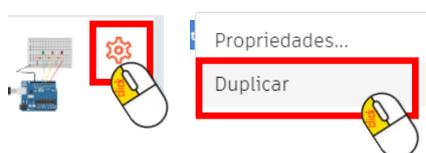
TIN
KER
CAD Podes **fazer** com que **pare de tocar** com a **instrução**:

```
noTone(buzzerPin);
```

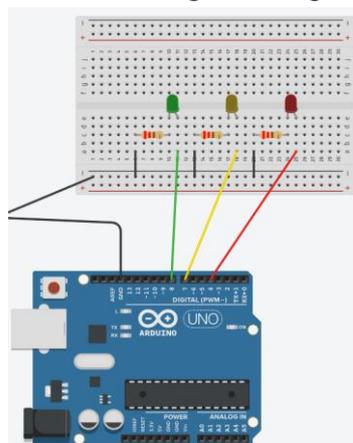
TIN
KER
CAD Agora deves **programar** o **buzzer** para que **toque** quando o **botão não** está **pressionado** (**porta aberta**) e para que se **desligue** quando a **porta estiver fechada**.

TIN
KER
CAD **Faz um print/recorte** que apanhe todo o **esquema elétrico** e ainda outro do **código** e **cola** no teu **diário de bordo**.

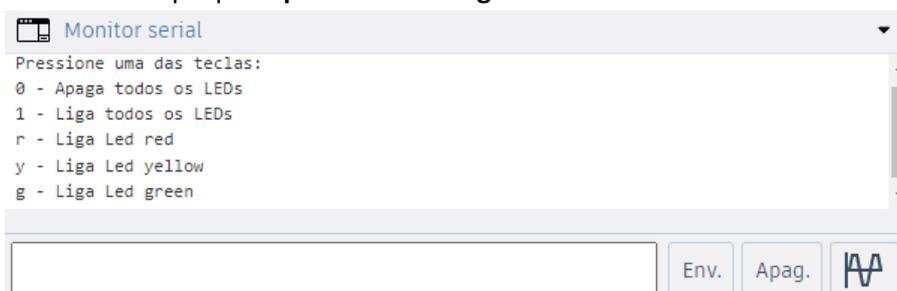
TIN
KER
CAD Clica em **TIN**
KER
CAD e na tua **conta** faz uma **cópia** do circuito do **semáforo**. Para isso por cima do **respetivo projeto** clica sobre a **roda dentada** e **escolhe a opção duplicar**.



Vamos **utilizar** este **circuito** para **acender os LEDs** conforme as **teclas** que **pressionamos** no nosso **teclado**. **Verifica** se tens o teu **circuito igual** ao seguinte antes de **prosseguir**:



TIN
KER
CAD **Aproveita** o **código existente**, neste caso a **definição de pinos** e a função **setup()**. **Dentro** da função **setup()** utiliza o **monitor série** para **imprimir as seguintes linhas** e assim **indicar ao utilizador** que pode **pressionar as seguintes teclas**.



TIN
KER
CAD Para já **ainda não funcionam** as teclas. **Vou ajudar-te** com uma **delas** e **terás de fazer** as **restantes opções**. Vamos abordar a **tecla 1**.



Começa por definir uma **variável**, de nome **key**, para **guardar a tecla** que se **vai pressionar**.

Deves fazê-lo no **início** junto da **definição** dos **pinos** a utilizar:

```
#define redPin 4
#define yellowPin 7
#define greenPin 8
char key;
```



Como podes ver **criou-se** uma **variável** de nome **Key** do tipo **char**, ou seja irá permitir **guardar um carater**.



De seguida **acrescenta** o seguinte **código** dentro da função **loop()**

```
if(Serial.available()>0){//código para não ler valores sem que as teclas sejam pressionadas
  key= Serial.read();
  Serial.print(key);
  switch(key){
    case '1':
      digitalWrite(redPin,HIGH);
      digitalWrite(yellowPin,HIGH);
      digitalWrite(greenPin,HIGH);
      break;
  }
}
```

Este **código** permite **inicialmente verificar** o que vai **sendo escrito** na **porta série**, mas **apenas** se as **teclas forem pressionadas**.

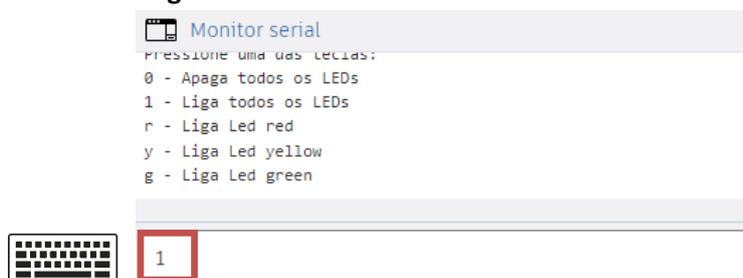
De seguida a **variável** criada (**key**) será **igualada** ao **Serial.read()**, ou seja, a tecla pressionada será **lida** pelo **Serial.read()** e ficará **guardada** na variável **key**.

Esta **variável** vai ser **imprimida** no **monitor série** para **termos a certeza** que está a ser **lida corretamente** e passará por uma **estrutura de decisão**, neste caso **Switch...case**. Para já apenas temos a **opção 1 (case 1:)** onde os **pinos dos leds, vermelho, amarelo e verde** são colocados a **alto, ligando todos** os LEDs.



Testa o código.  e **verifica** que **todos os LEDs acendem**.

Para **veres** os LEDs a **ligar** deves **escrever 1** no **monitor série** e **terminar** com **Enter**:



Completa o código, para as **restantes teclas**. Relembrando as **opções disponíveis**:

```
0 - Apaga todos os LEDs
1 - Liga todos os LEDs
r - Liga Led red
y - Liga Led yellow
g - Liga Led green
```



Faz um **print/recorte** que **apanhe** todo o **esquema elétrico** e ainda outro do **código** e cola no teu **diário de bordo**.



Chama o teu **professor** para **avaliar**.