

## Prática Arduino

### Roteiro de Aula Prática

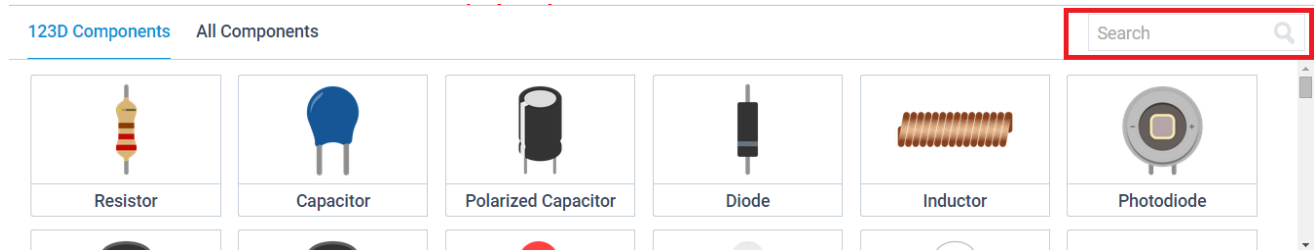
Nome: \_\_\_\_\_ Matric.: \_\_\_\_\_

#### Iniciando

Após efetuar o login na sua conta, crie um novo projeto do tipo “Electronics Lab”.  
Clique em “The Unnamed Circuit” e mude o nome do projeto pra Blink + Button.

#### Montando nosso circuito exemplo

Clique no botão “+ Components” na parte superior da tela. Um menu como esse será exibido:



Na barra de pesquisa (marcada com um quadrado), digite Resistor. Depois de encontrado, clique na figura do resistor e adicione dois resistores.

Após isso, adicione um LED, uma placa Arduino Uno e um Pushbutton, utilizando a barra de pesquisa para encontrar o componente.

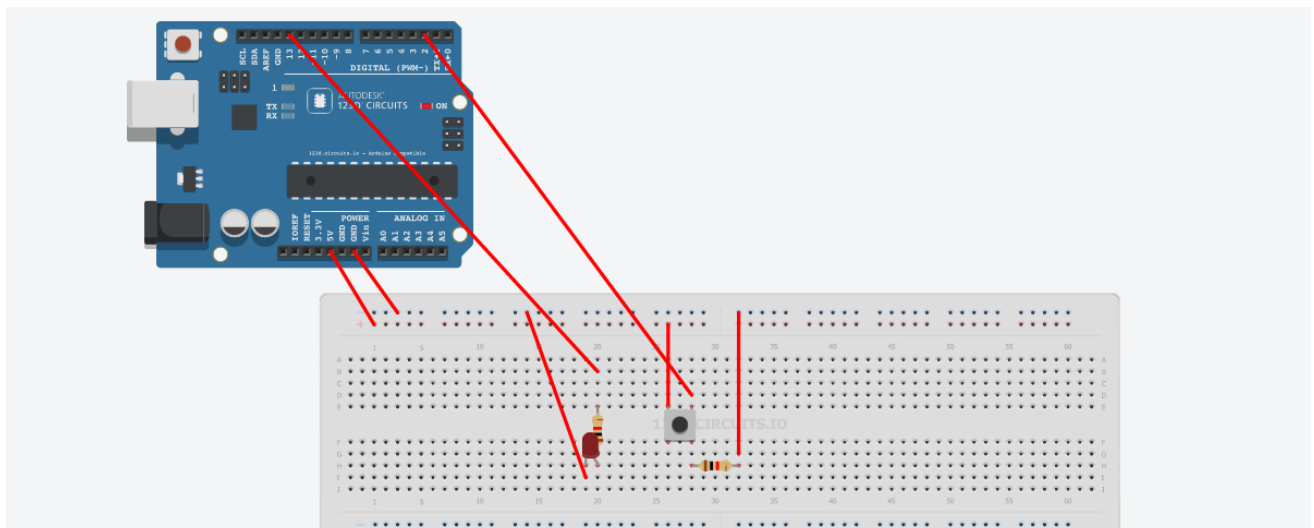
Neste ponto, você terá em sua tela: uma Protoboard; dois resistores; um LED, um pushbutton e uma placa Arduino Uno.

#### PushButton



O pushbutton internamente é uma chave que fecha o circuito quando está pressionado, e deixa o circuito aberto quando não está pressionado.

Monte o circuito exatamente como na figura abaixo (link com uma resolução melhor: <http://bit.do/piscabotao> )



**Importante:** O lado positivo do LED deve estar na mesma coluna do resistor.

O LED deverá estar conectado no pino 13 e o pushbutton no pino 2.

Clique no botão “Code Editor”. Será aberta uma janela, com um código já pronto.

Baixe o código exemplo em <http://bit.do/ExemploPiscabotao> . Copie o código e cole no editor de código do simulador.

Clique em Upload and Run. Caso tenha sucesso, a simulação começará a rodar. O que acontece quando você clica no pushbutton?

### Entendendo o código

A única linha que ainda não vimos na aula anterior foi a seguinte:

```
EstadoBotao = digitalRead(Botao); /*novo estado do botão  
vai ser igual ao que  
  Arduino ler no pino onde está o botão.  
  Poderá ser ALTO (HIGH) se o botão estiver  
  Pressionado, ou BAIXO (LOW), se o botão  
  estiver solto */
```

A função `digitalRead` lê e retorna o estado de um pino de INPUT. O estado pode ser HIGH ou LOW. Para o caso do botão, quando o mesmo está pressionado, o seu estado é HIGH, e quando está solto o seu estado é LOW. A sintaxe da função é:

#### **Função `digitalRead()`**

Lê o valor HIGH ou LOW do pino digital

#### **Sintaxe:**

`digitalRead(pino)`

#### **Parâmetros**


*Pino* – o número do pino o qual você deseja ler

**Retorno:** HIGH ou LOW

### Faça você mesmo

#### **Atividade 1**

Nossa primeira atividade desta prática será a criação de um sinal de trânsito, desta vez com o acionamento via botão. Nosso objetivo é simular a ação de um pedestre apertando o botão, para que o sinal feche para os carros, semelhante aos semáforos que existem em frente a universidade e na avenida PH Rolfs.

Clique no ícone  que está no canto superior esquerdo da tela (do lado do nome do seu projeto).

Clique no botão “+ New Eletronics Lab”. Um novo projeto irá abrir.

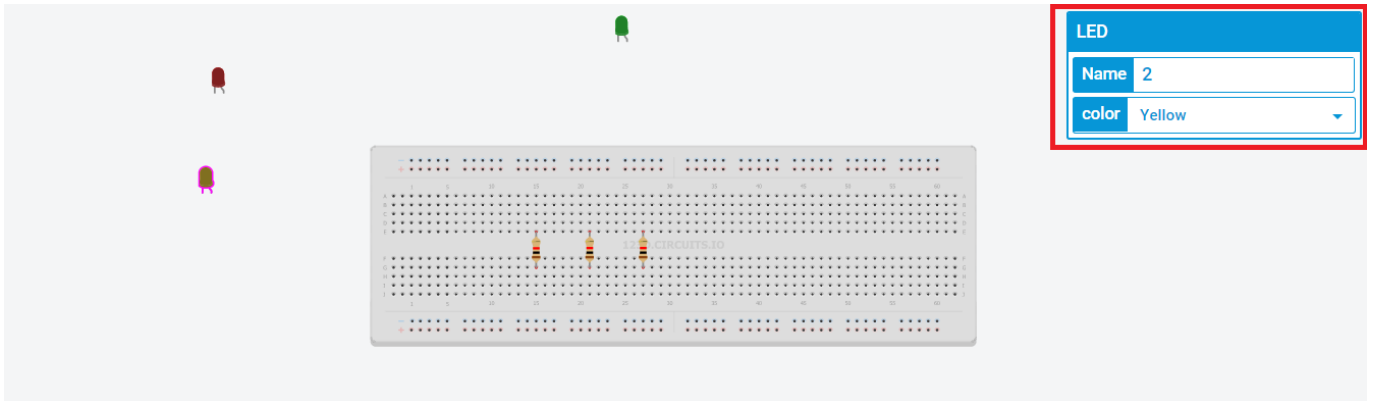
Renomeie o projeto para “Sinal de Transito V2”.

Adicione no projeto:

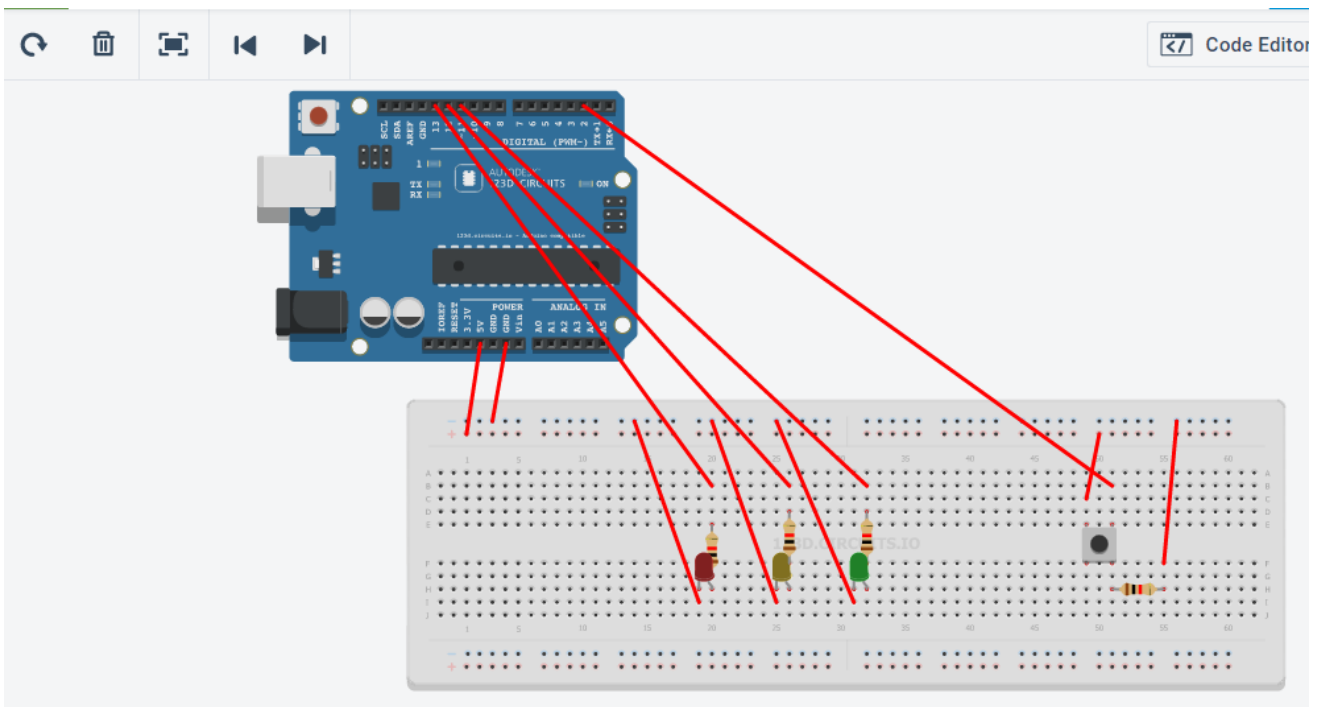
- 1 placa Arduino Uno
- 4 resistores
- 3 Leds.
- 1 Pushbutton

Mude as cores dos Leds para Vermelho, Amarelo e Verde.

Para mudar a cor, clique no LED. No canto direito aparecerá uma janela como a da figura abaixo:



Monte o Circuito como o da figura abaixo: (Link em maior resolução: <http://bit.do/sinalv2> )



A ligação dos leds é:

- Vermelho: Pino 13
- Amarelo: Pino 12
- Verde: Pino 11

O pushbutton deverá ser ligado no pino 2.


Baixe o código exemplo em <http://bit.do/Exemplosinalv2> . Copie o código e cole no editor de código do simulador.

Clique em Upload and Run. Caso tenha sucesso, a simulação começará a rodar. Os leds irão acender quando o botão estiver pressionado.

A sua tarefa é simular um sinal de trânsito que inicialmente fica com a luz verde acesa. Quando alguém pressiona o botão, a luz amarela se acende por 1, para que a luz vermelha se acenda por 5 segundos. Após isso a luz vermelha se apaga e a verde fica acesa novamente. Quando o botão é pressionado novamente, esse ciclo se repete.

## Atividade 2

Nesta atividade, além do sinal para carros, vamos adicionar o sinal para pedestres. Ambos serão controlados pelo pressionamento do botão.

Clique no ícone  que está no canto superior esquerdo da tela (do lado do nome do seu projeto).

Clique no botão “+ New Eletronics Lab”. Um novo projeto irá abrir.

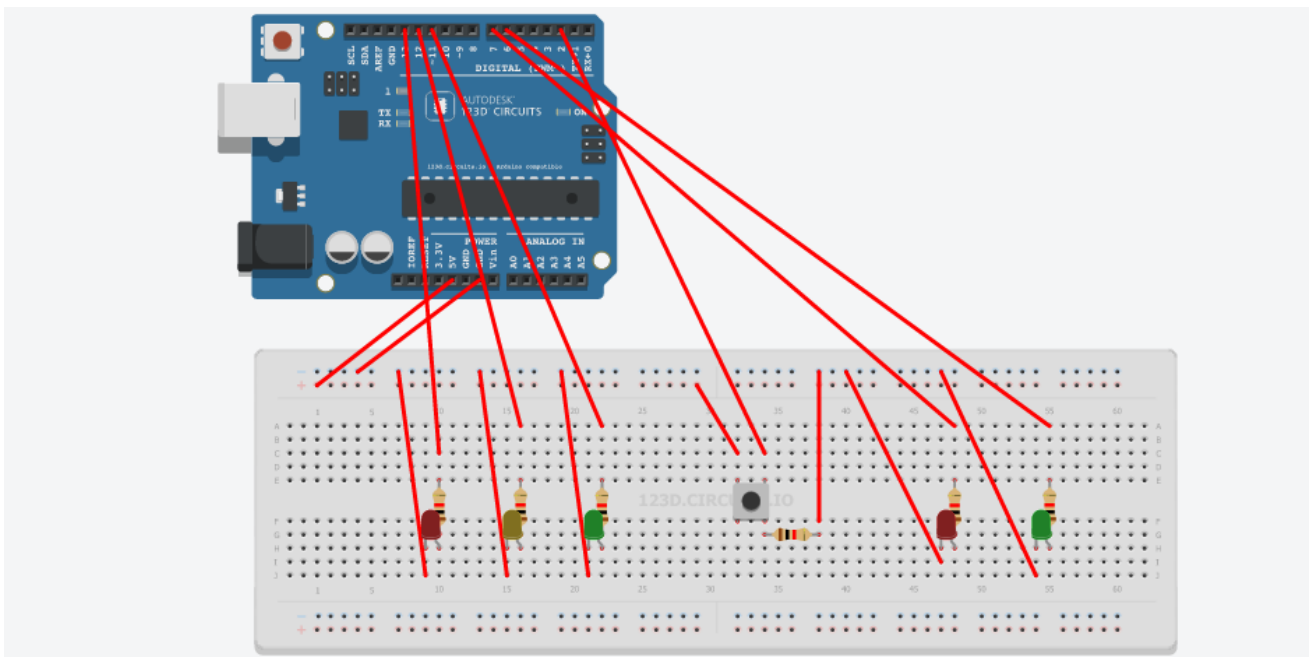
Renomeie o projeto para “Sinal de Transito V2”.

Adicione no projeto:

- 1 placa Arduino Uno
- 6 resistores
- 5 Leds.
- 1 Pushbutton

Mude as cores dos Leds para Vermelho (2 leds), Amarelo(1 led) e Verde (2 leds).

Monte o Circuito como o da figura abaixo: (Link em maior resolução: <http://bit.do/sinalv3> )



A ligação dos leds é:

- Vermelho: Pino 13 e pino 7
- Amarelo: Pino 12
- Verde: Pino 11 e pino 6
- 

O pushbutton deverá ser ligado no pino 2

Baixe o código exemplo em <http://bit.do/Exemplosinalv3> . Copie o código e cole no editor de código do simulador.

Clique em Upload and Run. Caso tenha sucesso, a simulação começará a rodar. Os leds irão acender quando o botão estiver pressionado.

A sua tarefa é simular um sinal de trânsito que inicialmente fica com a luz verde acesa para os carros e vermelha para os pedestres. Quando alguém pressiona o botão, a luz amarela acende por 1 segundo, e logo depois o sinal fecha para os carros. No mesmo momento que o sinal fecha para os carros, ele abre para os pedestres. O sinal de pedestre fica aberto por 5 segundos e depois fecha, para que o sinal de carros abra novamente. Quando o botão é pressionado novamente, este ciclo se repete.

### DESAFIO

Fazer com que a luz vermelha para pedestre pisque 3 vezes antes de o sinal de pedestre fechar e o sinal de carros abrir.