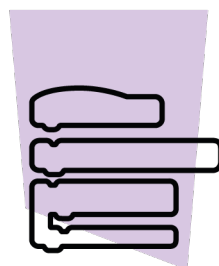
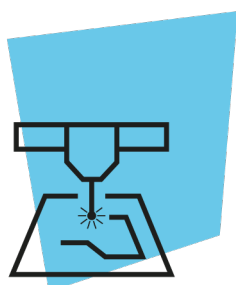
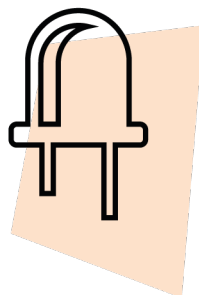
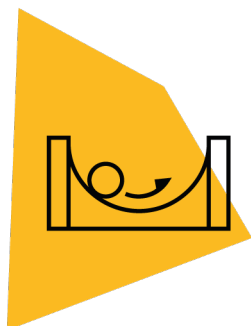
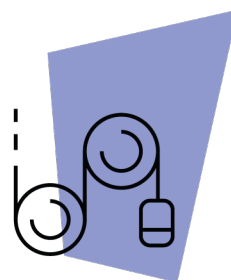
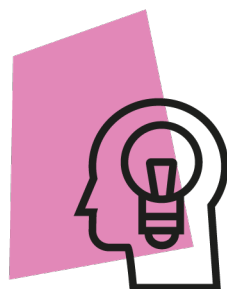


# ROBÓTICA

Módulo 1



**Semáforo**  
**Módulo Semáforo**  
**[Pedestres]**

AULA

11

\*Disponível no kit distribuído em 2023

**GOVERNADOR DO ESTADO DO PARANÁ**

Carlos Massa Ratinho Júnior

**SECRETÁRIO DE ESTADO DA EDUCAÇÃO**

Roni Miranda Vieira

**DIRETOR DE TECNOLOGIA E INOVAÇÃO**

Claudio Aparecido de Oliveira

**COORDENADOR DE TECNOLOGIAS EDUCACIONAIS**

Marcelo Gasparin

**Produção de Conteúdo**

Darice Alessandra Deckmann Zanardini

**Revisão Textual**

Orlando de Macedo Junior

**Projeto Gráfico e Diagramação**

Edna do Rocio Becker

**Ilustração**

Jocelin Vianna (ASCOM)

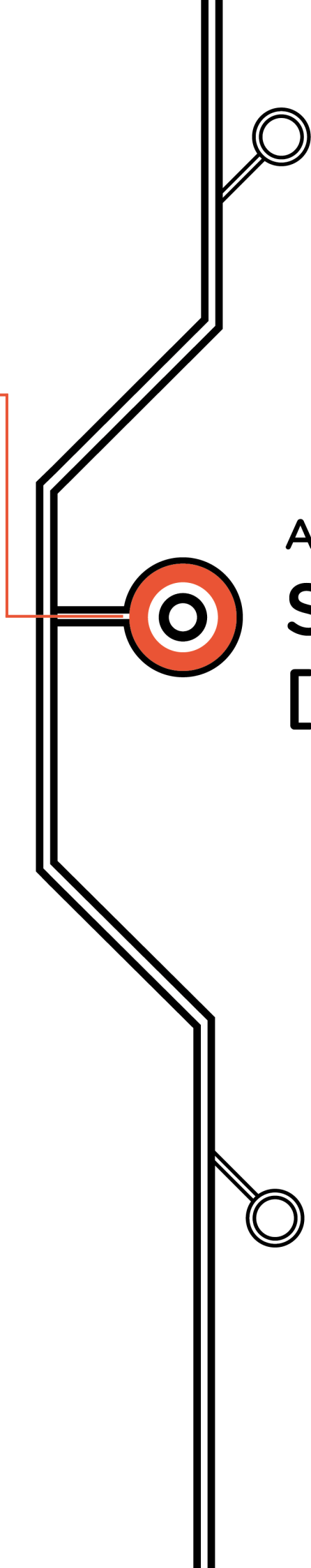
**2023**

- Aula 01 Por Que Robótica?
- Aula 02 Tensão, Corrente e Resistência
- Aula 03 Kit de Robótica
- Aula 04 Arduino Uno R3
- Aula 05 Softwares Arduino IDE e mBlock
- Aula 06 Portas Digitais
- Aula 07 Circuito Elétrico
- Aula 08 LED e Resistor
- Aula 09 Semáforo [Carros]
- Aula 10 Semáforo [Cruzamento Carros]
- Aula 11 Semáforo [Pedestres]
- Aula 12 Semáforo [Cruzamento Carros + Pedestres]
- Aula 13 Push Button
- Aula 14 Feedbacks + Inventário I
- Aula 15 Semáforo [Carros + Pedestres com Botão]
- Aula 16 Display 7 Segmentos
- Aula 17 Fonte DC + Plug P4
- Aula 18 Portas PWM
- Aula 19 LED Fade-In
- Aula 20 LED Fade-Out
- Aula 21 Super Máquina 80's
- Aula 22 Super Máquina 2008
- Aula 23 Potenciômetro
- Aula 24 Buzzer Passivo
- Aula 25 LED RGB
- Aula 26 Arco-Iris
- Aula 27 Sensor LDR
- Aula 28 Feedbacks + Inventário II
- Aula 29 Sensor de Temperatura
- Aula 30 Sensor de Obstáculo IR
- Aula 31 Controle Motor DC
- Aula 32 Kit Chassi 2WD Robô
- Aula 33 Seguidor de Linha
- Aula 34 Sensor de Distância
- Aula 35 Sensor de Estacionamento
- Aula 36 Display LCD 16x2
- Aula 37 Trena Digital
- Aula 38 Robô Sumô [Estrutura]
- Aula 39 Robô Sumô [Programação + Treinamento I]
- Aula 40 Robô Sumô [Programação + Treinamento II]
- Aula 41 Disputa de Sumôs
- Aula 42 Feedbacks + Inventário III

Aula 10  
Semáforo  
[Cruzamento Carros]

Aula 11  
**Semáforo**  
**[Pedestres]**

Aula 12  
Semáforo  
[Cruzamento Carros  
+ Pedestres]



# Sumário

Introdução	2
Objetivos desta aula	2
Competências gerais previstas na BNCC	3
Habilidades do século XXI a serem desenvolvidas	4
Lista de materiais	4
Roteiro da aula	5
1. Contextualização	5
2. Montagem e programação	6
3. Feedback e finalização	12
Videotutorial	13
Referências	14



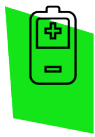
## Introdução

Como você viu na **Aula 10 Semáforo [Cruzamento Carros]** os semáforos são dispositivos formados por sistemas de luzes com cores (vermelho, amarelo e verde) criados para organizar as travessias de veículos por ruas e avenidas nas grandes cidades, evitando assim as colisões e acidentes. Contudo, nas vias urbanas não circulam somente veículos, mas também pessoas, pensando nisso, foram feitos semáforos para pedestres, outra forma de contribuir com a organização do trânsito. Nesta aula, programaremos um semáforo para pedestres para compreender como ocorre esses sincronismos de tempo entre o fluxo de automóveis e a travessias de pedestres, compreendendo assim o funcionamento deste sistema nas organizações do trânsito da cidade.



## Objetivos desta aula

- Abordar conceitos algorítmicos;
- Compreender conceitos da lógica booleana;
- Trabalhar com sincronias e técnicas de programação;
- Realizar sincronismo de sinais com o uso da lógica booleana;
- Utilizar a função `for()`;
- Prototipar com Arduino;
- Programar por blocos ou código.



## Competências gerais previstas na BNCC

**[CG02]** - Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas.

**[CG04]** - Utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital –, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo.

**[CG05]** - Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

**[CG09]** - Exercitar a empatia, o diálogo, a resolução de conflitos e a cooperação, fazendo-se respeitar e promovendo o respeito ao outro e aos direitos humanos, com acolhimento e valorização da diversidade de indivíduos e de grupos sociais, seus saberes, identidades, culturas e potencialidades, sem preconceitos de qualquer natureza.

**[CG10]** - Agir pessoal e coletivamente com autonomia, responsabilidade, flexibilidade, resiliência e determinação, tomando decisões com base em princípios éticos, democráticos, inclusivos, sustentáveis e solidários.



## Habilidades do século XXI a serem desenvolvidas

- Pensamento crítico;
- Afinidade digital;
- Resiliência;
- Resolução de problemas;
- Colaboração;
- Comunicação.



## Lista de materiais

- 1 placa Arduino Uno R3;
- 1 cabo USB;
- 1 módulo semáforo
- 1 computador;
- Software mBlock ou Arduino IDE.



## Roteiro da aula

### 1. Contextualização (15min)

Qual a importância de organizarmos o sincronismo de semáforos e o tempo destinado à travessia? Imagine a seguinte situação: o entorno da escola possui grande movimentação de veículos, especialmente nos horários de chegada e saída de alunos, comprometendo a segurança de todos. Mesmo com lombada na rua, muitos motoristas não reduzem a velocidade de seus veículos nem respeitam a movimentação dos alunos. Que tal projetar e programar o protótipo de um semáforo de pedestres como alternativa à dificuldade de segurança na movimentação da comunidade escolar?

Figura 1 - Ilustração Semáforo de Pedestres



Fonte: projetado pelo [Freepik](#)

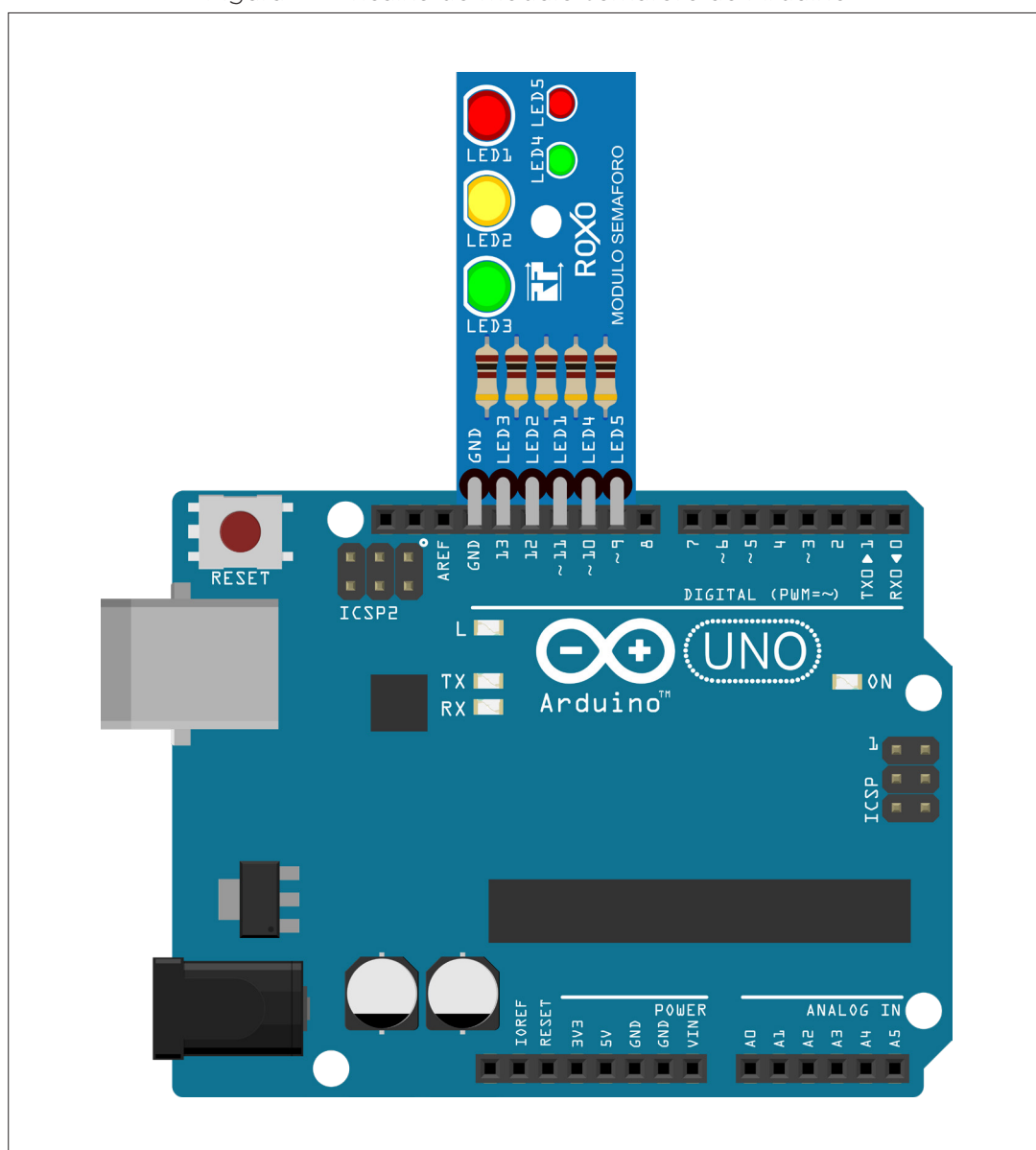
Agora, chegou o momento de organizar os componentes necessários a esta aula. Para isso, confira a lista de peças acima e separe o que você precisará.



## 2. Montagem e Programação (60min)

Encaixe o módulo semáforo diretamente na placa Arduino observando a posição do pino GND do módulo (pino à esquerda) que deverá ser encaixado à porta GND do Arduino (figura 2).

Figura 2 - Encaixe do módulo semáforo ao Arduino



Fonte: Fritzing

Com o módulo semáforo devidamente encaixado na placa Arduino, vamos programar, por codificação e por blocos, o protótipo de um semáforo de pedestres.

### i. Linguagem de programação por código

Para iniciar a programação, conecte a placa Arduino ao computador, através de um cabo USB, para que ocorra a comunicação entre a placa microcontroladora e o software Arduino IDE.

No software IDE, crie um sketch e lembre-se de selecionar a porta que o computador atribuiu ao Arduino; então, escreva ou copie e cole o código-fonte de programação (quadro 1).

Quadro 1 - Código-fonte da programação na linguagem do Arduino (Wiring)

```
/* Declaração de variáveis */
int led_vermelho_ped = 9;
int led_verde_ped = 10;
int led_vermelho_car = 11;
int led_amarelo_car = 12;
int led_verde_car = 13;

/* Configurando os pinos dos LEDs como saída */
void setup() {
  pinMode(led_vermelho_ped, OUTPUT);
  pinMode(led_verde_ped, OUTPUT);
  pinMode(led_vermelho_car, OUTPUT);
  pinMode(led_amarelo_car, OUTPUT);
  pinMode(led_verde_car, OUTPUT);
}

/* Bloco Principal onde está toda a execução das ações do programa */
void loop() {

  /* Primeiro estágio: Aberto para os carros e fechado para os
pedestres */
  digitalWrite(led_vermelho_ped, 1);
  digitalWrite(led_verde_car, 1);
  delay(5000);

  /* Atenção para os carros (LED amarelo ligado por 1 segundo) */
  digitalWrite(led_amarelo_car, 1);
  digitalWrite(led_verde_car, 0);
  delay(1000);
```

```
/* Segundo estágio: Aberto para os pedestres e fechado para os
carros */
digitalWrite(led_vermelho_car, 1);
digitalWrite(led_amarelo_car, 0);
digitalWrite(led_vermelho_ped, 0);
digitalWrite(led_verde_ped, 1);
delay(5000);
/* Desliga o LED verde dos pedestres para iniciar o fechamento do
sinal */
digitalWrite(led_verde_ped, 0);
/* Atenção para os pedestres (LED vermelho pisca indicando que irá
fechar) */
for (int count = 0; count < 5; count++) {
digitalWrite(led_vermelho_ped, 1);
delay(250);
digitalWrite(led_vermelho_ped, 0);
delay(250);
}

/* Apaga o LED vermelho dos carros e repete a sequência */
digitalWrite(led_vermelho_car, 0);
}
```

Com o código-fonte inserido no Arduino IDE, compile o programa pressionando o botão **Verificar** para averiguar se não há erros de sintaxe. Estando o código correto, o próximo passo é realizar a transferência do programa para o Arduino. Pressione o botão **Carregar**, para realizar upload do programa para o Arduino.

Após a transferência do programa para o Arduino, os LEDs presentes no módulo semáforo devem acender conforme a sequência programada, simulando o funcionamento do semáforo de pedestres.

## ii. Linguagem de programação por blocos

Outra forma de simular o funcionamento do semáforo é por meio da linguagem de programação que utiliza blocos de funções prontas, os quais representam comandos de programação. Vamos utilizar o software mBlock.



Para conectar o mBlock ao Arduino, você deve clicar no ícone **Adicionar**, localizado no campo **Dispositivos**, e selecionar o Arduino, na biblioteca de dispositivos do mBlock, clicando, na sequência, no botão **OK**.

Uma vez selecionado, o Arduino Uno é visualizado no campo **Dispositivos** do mBlock e já é possível iniciar a programação em blocos.

Monte os blocos, arrastando e soltando, de acordo com a programação de funcionamento do semáforo (figura 3).

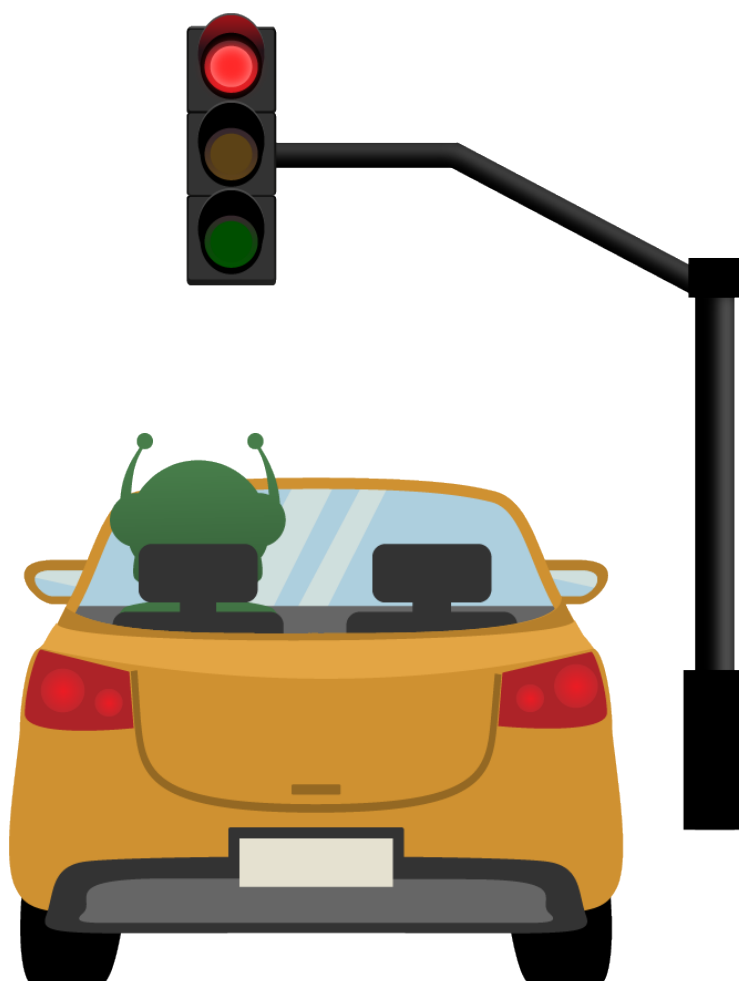


Figura 3 - Programação em blocos para funcionamento de semáforos

```
quando o Arduino Uno começar
  definir a saída do pino digital 9 como baixo
  definir a saída do pino digital 10 como baixo
  definir a saída do pino digital 11 como baixo
  definir a saída do pino digital 12 como baixo
  definir a saída do pino digital 13 como baixo
  repetir para sempre
    definir a saída do pino digital 9 como alto
    definir a saída do pino digital 13 como alto
    definir a saída do pino digital 10 como baixo
    definir a saída do pino digital 11 como baixo
    definir a saída do pino digital 12 como baixo
    esperar 5 segundo(s)
    definir a saída do pino digital 13 como baixo
    definir a saída do pino digital 12 como alto
    esperar 3 segundo(s)
    definir a saída do pino digital 9 como baixo
    definir a saída do pino digital 11 como alto
    definir a saída do pino digital 12 como baixo
    definir a saída do pino digital 10 como alto
    esperar 5 segundo(s)
    definir a saída do pino digital 10 como baixo
    repetir 5
      definir a saída do pino digital 9 como alto
      esperar 0.25 segundo(s)
      definir a saída do pino digital 9 como baixo
      esperar 0.25 segundo(s)
    definir a saída do pino digital 11 como baixo
```

Fonte: site mBlock oficial.

Assim que os blocos estiverem montados, clique no botão **Conectar** para iniciar a comunicação entre o software mBlock com a placa Arduino Uno. Ao clicar sobre o botão **Conectar**, aparecerá um *tooltip* solicitando a confirmação da conexão entre os dois dispositivos.

Uma vez realizada a conexão entre os dispositivos, será ativado, na interface do mBlock, o botão **Upload**, o qual, ao ser clicado, o software irá verificar se não há erros na estrutura do programa e, então, compilará para enviar o programa à placa Arduino.

Com a transferência do código para o dispositivo Arduino Uno, inicia-se o funcionamento do semáforo, ou seja, os LEDs começam a acender e a apagar, de acordo com a ordem e o tempo definido na programação em blocos.



## Desafios

- i. Que tal alterar os valores de delay pra dar uma variada no tempo que os pedestres têm pra atravessar? Bora encarar esse desafio!
- ii. Que tal também projetar a construção de seu semáforo fora da placa de prototipagem, simulando o cruzamento de pedestres?

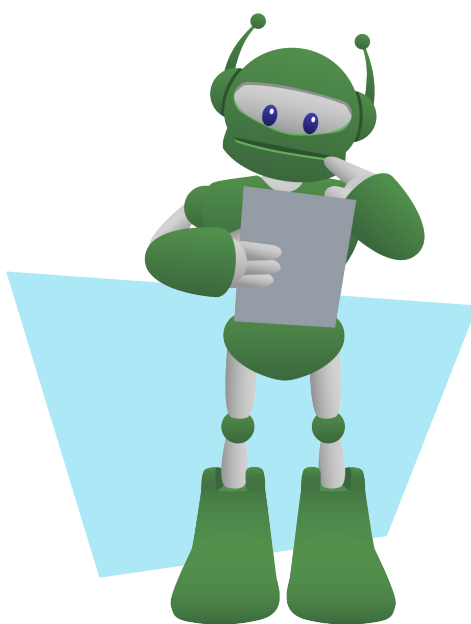


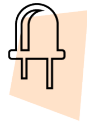
## E se...

- i. O projeto não funcionar?
  1. Verifique se o módulo semáforo foi corretamente conectado à placa Arduino;
  2. Verifique se a programação está adequada a cada porta digital;

### 3. Feedback e Finalização (15min)

- a.** Confira, compartilhando seu projeto com os demais colegas, se o objetivo foi alcançado.
- b.** Analise seu projeto desenvolvido, de modo a atender aos requisitos para funcionamento de um semáforo: sequência correta do acendimento das cores verde, amarelo e vermelho para os veículos e, de modo simultâneo, das cores vermelho e verde para pedestres.
- c.** Reflita se as seguintes situações ocorreram:
  - i.** Colaboração e cooperação: você e os membros de sua equipe interagiram entre si, compartilhando ideias que promoveram a aprendizagem e o desenvolvimento deste projeto?
  - ii.** Pensamento crítico e resolução de problemas: você conseguiu identificar os problemas, analisar informações e tomar decisões de modo a contribuir para o projeto desenvolvido?
- d.** Reúna todos os componentes utilizados nesta aula e os organize novamente, junto aos demais, no kit de robótica.





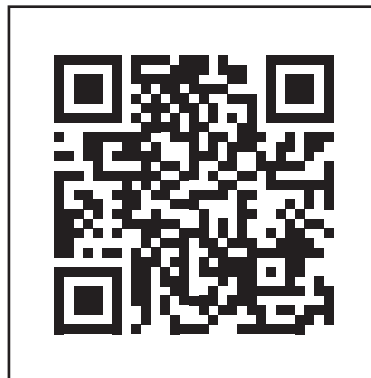
## Videotutorial

Com o intuito de auxiliar na montagem e na programação desta aula, apresentamos um videotutorial, disponível em:



<https://rebrand.ly/a11roboticamod>

Acesse, também, pelo QRCode:







## Referências

ARDUINO. Site oficial. **Ambiente de Programação do Arduino**. Disponível em: <https://create.arduino.cc/editor>. Acesso em: 15 out. 2021.

ARDUINO. Site oficial. **Downloads**. Disponível em: [www.arduino.cc/en/Main/Software](http://www.arduino.cc/en/Main/Software). Acesso em: 15 out. 2021.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Brasília, 2018. Disponível em: [http://basenacionalcomum.mec.gov.br/images/BNCC\\_EI\\_EF\\_110518-versaofinal\\_site.pdf](http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518-versaofinal_site.pdf). Acesso em: 18 out. 2021.

MAKEBLOCK. mBlock. **Download mBlock**. Disponível em: <https://mblock.makeblock.com/en-us/download/>. Acesso em: 15 out. 2021.

MAKEBLOCK. mBlock. **Programação em blocos**. Disponível em: <https://ide.mblock.cc/>. Acesso em: 15 out. 2021.

FREEPIK. **Vetor**. Download de imagem semáforo travessia de pedestre. Disponível em: [https://br.freepik.com/vetores-gratis/semaforo-e-faixa-de-pedestres-na-rua-da-cidade\\_6993818.htm#query=semaforo%20pedestre&position=0&from\\_view=search](https://br.freepik.com/vetores-gratis/semaforo-e-faixa-de-pedestres-na-rua-da-cidade_6993818.htm#query=semaforo%20pedestre&position=0&from_view=search). Acesso em: 05 jan. 2022.

DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)  
COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

**EQUIPE ROBÓTICA PARANÁ**

Andrea da Silva Castagini Padilha  
Cleiton Rosa  
Darice Alessandra Deckmann Zanardini  
Edgar Cavalli Júnior  
Edna do Rocio Becker  
José Feuser Meurer  
Marcelo Gasparin  
Michele Serpe Fernandes  
Michelle dos Santos  
Orlando de Macedo Júnior  
Roberto Carlos Rodrigues

Os materiais, aulas e projetos da “Robótica Paraná” foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação do Paraná (SEED), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica. Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.



Este trabalho está licenciado com uma Licença  
Creative Commons – CC BY-NC-SA  
[Atribuição - NãoComercial - Compartilha Igual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

