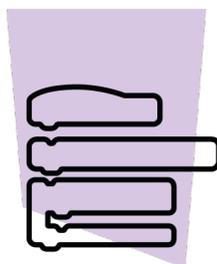
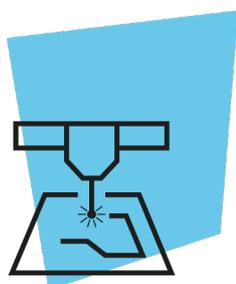
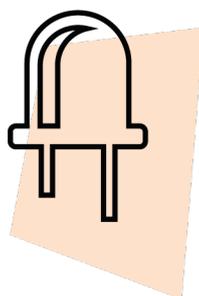
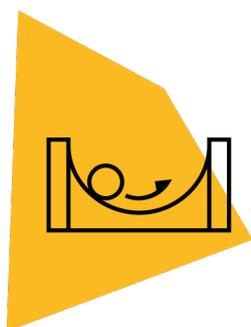
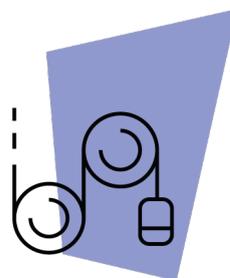
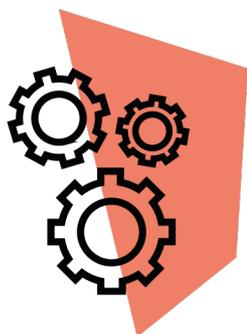


# ROBÓTICA

Módulo 1



## Semáforo

### Módulo Semáforo

### [Cruzamento Carros]

AULA 10

\*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**

Cleiton Rosa

Simone Sinara de Souza

**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 09  
Semáforo [Carros]

# Aula 10 Semáforo [Cruzamento Carros]

Aula 11  
Semáforo [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	5
3. Feedback e finalização	12
Videotutorial	13
Referências	14



### Introdução

Em uma cidade é comum observarmos, em cruzamentos de ruas e avenidas, a circulação constante de veículos. Para evitar conflito entre os condutores por quererem circular simultaneamente por estas vias e, possíveis riscos de colisões, são instalados dispositivos de sinalização, ou semáforos, para controlar o direito de passagem dos veículos.

Nesta aula, você terá a oportunidade de prototipar um semáforo de cruzamento de veículos e entender o funcionamento deste.



### Objetivos desta aula

- Compreender a importância da presença dos semáforos de veículos em cruzamentos de vias urbanas;
- Prototipar dois semáforos de veículos, simulando a presença destes em cruzamento de ruas e avenidas.
- Entender, via programação, o funcionamento dos semáforos de veículos dispostos em cruzamentos de vias públicas.



## 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.



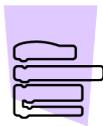
## Habilidades do século XXI a serem desenvolvidas

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



## Lista de Materiais

- 1 placa Arduino Uno R3;
- 1 cabo USB;
- 2 módulos semáforo;
- Software mBlock ou Arduino IDE.



## Roteiro da aula

### 1. Contextualização (15min)

Qual a importância de organizarmos o sincronismo de semáforos e o tempo destinado a cruzamentos de estradas?

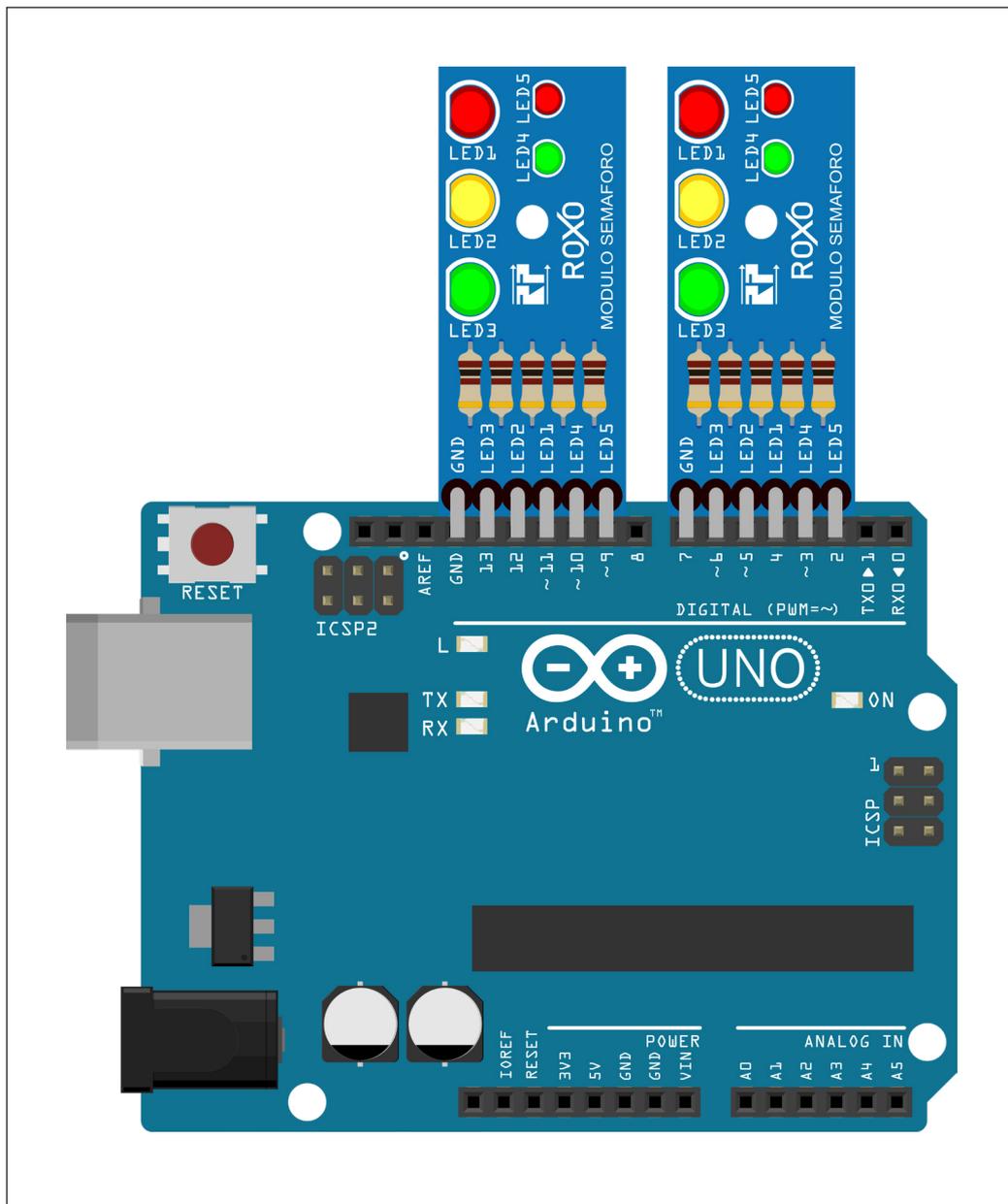
Imagine a seguinte situação: no centro de uma grande cidade, um cruzamento de duas avenidas está causando congestionamentos, são muitos carros que precisam passar por ali diariamente. A norma de trânsito que estabelece regras de controle no direito de passagem, quando o que chegar primeiro tem a preferência pela travessia no cruzamento, não funciona na situação levantada. Nesse caso, é recomendado a instalação de dispositivos de sinalização para controlar o direito de passagem dos veículos. Os semáforos, quando sincronizados nos cruzamentos, permitem a fluidez do trânsito e, conseqüentemente, a redução no risco de colisões.

### 2. Montagem e programação (60min)

Encaixe os módulos semáforo na placa Arduino observando a posição correta dos pinos GND dos módulos semáforo 1 e 2 nas portas do Arduino (GND e digital 7 respectivamente) (figura 1).

Pode parecer meio estranho conectar o pino GND do módulo de semáforo em uma porta digital do Arduino, não é? Mas com o jeito certo de programar, a gente consegue manter a porta digital 7 com um nível lógico BAIXO, ou seja, em 0V, igualzinho ao comportamento de um pino GND.

Figura 1 - Encaixe dos módulos semáforo ao Arduino



Fonte: SEED/DTI/CTE

Com os componentes eletrônicos montados, vamos programar, por codificação e por blocos, o protótipo de um semáforo cruzamento de veículos.

### i. Linguagem de programação por código

Para iniciar a programação, conecte a placa Arduino ao computador, com 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)

```
/* Código de Funcionamento para um Semáforo de Cruzamento */

/* Define as portas dos LED's do semáforo 1 */
#define LED_Vermelho_1 11
#define LED_Amarelo_1 12
#define LED_Verde_1 13

/* Define as portas dos LED's do semáforo 2 */
#define LED_Vermelho_2 4
#define LED_Amarelo_2 5
#define LED_Verde_2 6
#define GND_2 7

/* Define todos os pinos dos LED's como SAÍDA */
void setup() {
  pinMode(LED_Vermelho_1, OUTPUT);
  pinMode(LED_Amarelo_1, OUTPUT);
  pinMode(LED_Verde_1, OUTPUT);
  pinMode(LED_Vermelho_2, OUTPUT);
  pinMode(LED_Amarelo_2, OUTPUT);
  pinMode(LED_Verde_2, OUTPUT);
  pinMode(GND_2, OUTPUT);
  digitalWrite(GND_2, LOW);
}

void loop() {
  /* Primeiro estágio: Semáforo 1 ABERTO e Semáforo 2 FECHADO (5 segundos) */
  digitalWrite(LED_Vermelho_1, LOW);
  digitalWrite(LED_Amarelo_1, LOW);
  digitalWrite(LED_Verde_1, HIGH);
  digitalWrite(LED_Vermelho_2, HIGH);
  digitalWrite(LED_Amarelo_2, LOW);
  digitalWrite(LED_Verde_2, LOW);
  delay(5000);
}
```

```
    /* Segundo estágio: Semáforo 1 ATENÇÃO E Semáforo 2 FECHADO (3
segundos) */
    digitalWrite(LED_Vermelho_1, LOW);
    digitalWrite(LED_Amarelo_1, HIGH);
    digitalWrite(LED_Verde_1, LOW);
    digitalWrite(LED_Vermelho_2, HIGH);
    digitalWrite(LED_Amarelo_2, LOW);
    digitalWrite(LED_Verde_2, LOW);
    delay(3000);

    /* Terceiro estágio: Semáforo 1 Fechado e Semáforo 2 Aberto (5
segundos) */
    digitalWrite(LED_Vermelho_1, HIGH);
    digitalWrite(LED_Amarelo_1, LOW);
    digitalWrite(LED_Verde_1, LOW);
    digitalWrite(LED_Vermelho_2, LOW);
    digitalWrite(LED_Amarelo_2, LOW);
    digitalWrite(LED_Verde_2, HIGH);
    delay(5000);

    /* Quarto estágio: Semáforo 1 FECHADO e Semáforo 2 ATENÇÃO (3
segundos) */
    digitalWrite(LED_Vermelho_1, HIGH);
    digitalWrite(LED_Amarelo_1, LOW);
    digitalWrite(LED_Verde_1, LOW);
    digitalWrite(LED_Vermelho_2, LOW);
    digitalWrite(LED_Amarelo_2, HIGH);
    digitalWrite(LED_Verde_2, LOW);
    delay(3000);
}
```

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 nos módulos semáforo devem acender conforme a sequência programada, simulando o funcionamento do cruzamento de semáforos.



## 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.

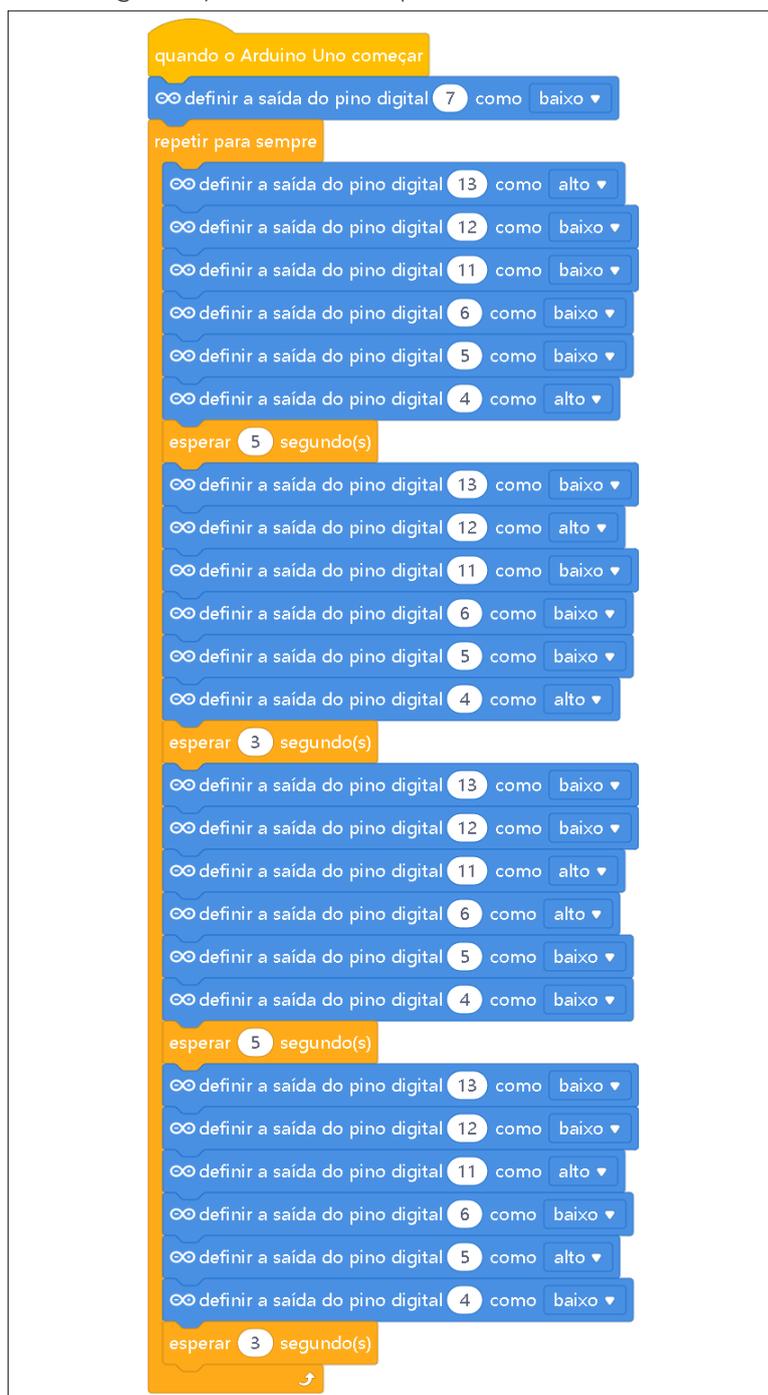
Para conectar o mBlock ao Arduino, você deve clicar no ícone **Adicionar**, localizado no campo **Dispositivos**, e selecionar o Arduino, na biblioteca de dispositivos do mBlock, clicando, na sequência, no botão **OK**.

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 2).

Figura 2 - Programação em blocos para funcionamento de semáforos



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 (figura 3).

Figura 3 - Confirmar conexão entre mBlock e Arduino Uno



Fonte: site mBlock oficial.

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 delays dos estagios dos semáforos para deixar mais tempo pros carros na rua mais congestionada?
- ii. Que tal também projetar a construção de seu semáforo diretamente em uma maquete, simulando o cruzamento de duas ruas?

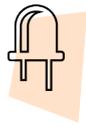


### E se...

- i. O projeto não funcionar?
  1. Verifique se os módulos semáforos foram conectados corretamente nas portas digitais do Arduino;

### 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, para os dois semáforos.
- 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/a10roboticamod>

Acesse, também, pelo QRCode:



# AULA 10



## Referências

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.

**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 - Compartilhalgual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

