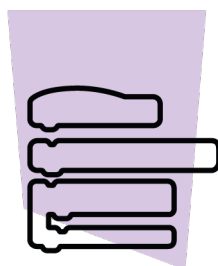
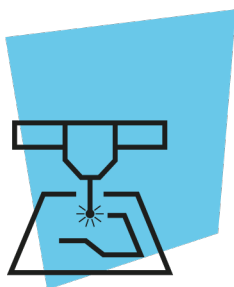
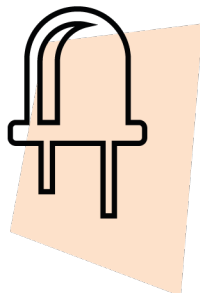
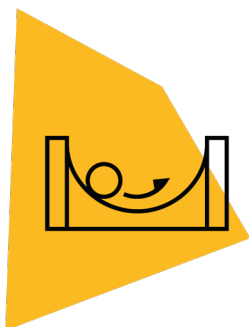
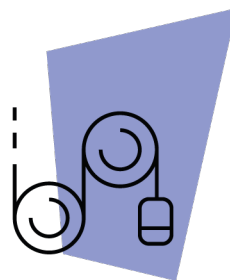


ROBÓTICA

Módulo 1



Semáforo

Módulo Semáforo

[Cruzamento
Carros + Pedestres]

AULA

12

*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

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 11
Semáforo
[Pedestres]

Aula 12
Semáforo
[Cruzamento
Carros
+ Pedestres]

Aula 13
Push Button

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	15
Videotutorial	16
Referências	17



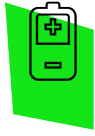
Introdução

Conforme visto nas aulas anteriores, os semáforos são muito importantes para a organização do trânsito das cidades, eles ajudam a evitar acidentes e não formar congestionamento. Nesta aula, compreenderemos como funciona o sistema de um semáforo que trabalha de forma conjunta para a circulação de carros e travessia de pedestres em cruzamentos.



Objetivos desta aula

- Compreender sobre circuitos;
- Apropriar-se de conceitos básicos de eletrônica;
- Perceber a linguagem visual como conceito mais universal;
- Prototipar com Arduino;
- Programar por blocos ou código;
- 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.



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.



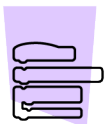
Habilidades do século XXI a serem desenvolvidas

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



Lista de materiais

- 1 placa Arduino Uno R3;
- 1 cabo USB;
- 2 módulos semáforo
- 1 notebook;
- 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 um grande congestionamento, são muitos carros que precisam passar por ali diariamente, além de pedestres que também precisam atravessar as avenidas de um lado para outro. Que tal projetar e programar o protótipo de semáforos de veículos e de pedestres como alternativa para amenizar esse problema?

Figura 1 - Ilustração



Fonte: SEED/DTI/CTE

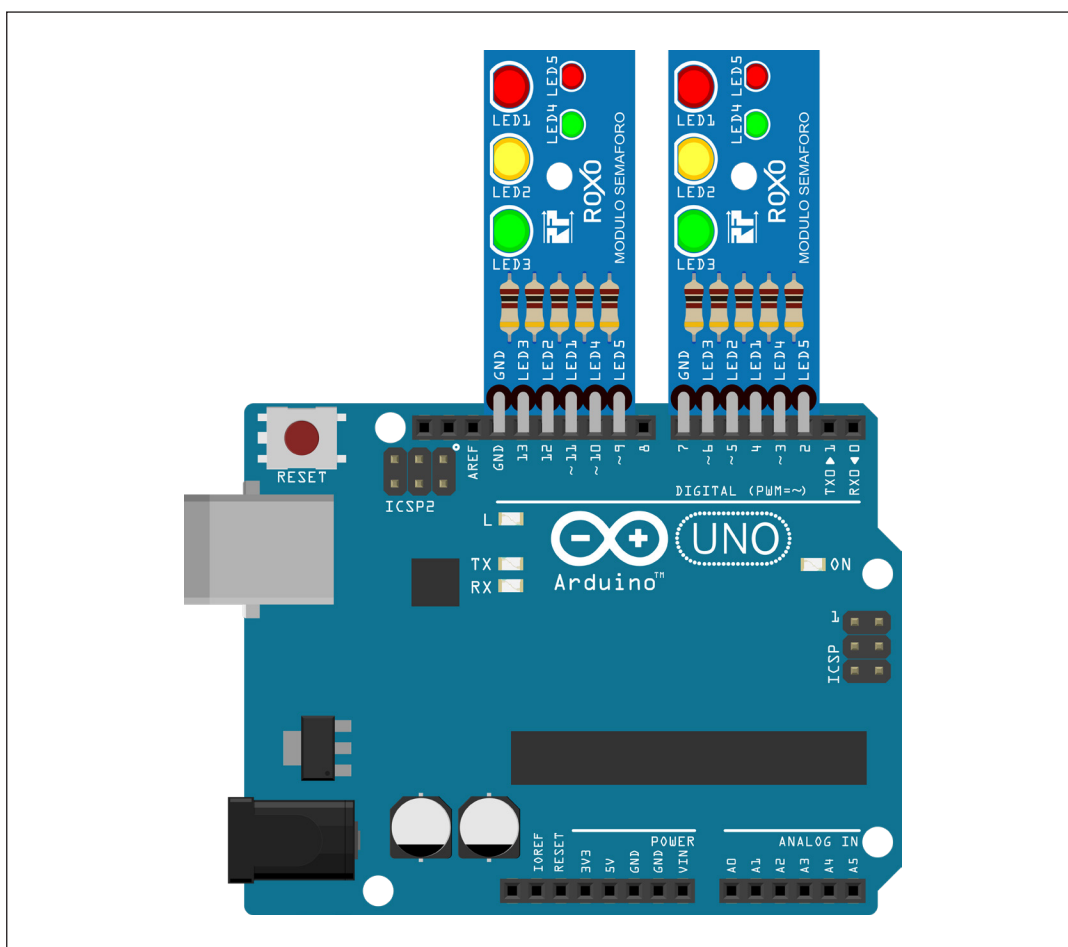
Agora, chegou o momento de organizar os componentes necessários a esta aula. Para isso, confira a lista de peças acima e separe o que você precisará.

2. Montagem e programação (60min)

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

Se você ainda estiver estranhando a conexão do pino GND do módulo de semáforo em uma porta digital do Arduino, lembre-se que na **Aula 10 - Semáforo Cruzamento Carros** a gente já realizou este procedimento, ou seja, com o jeito certo de programar, conseguimos manter a porta digital 7 com um nível lógico BAIXO, ou seja, em 0V, igualzinho ao comportamento de um pino GND.

Figura 2 - Encaixe dos módulos semáforo ao Arduino













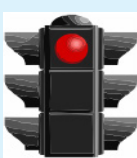





Fonte: Fritzing

Com os módulos semáforo conectados ao Arduino, vamos programar, por codificação e por blocos, o protótipo de um semáforo de cruzamento de veículos e pedestres.

Vale destacar que a programação será baseada em quatro estágios (tabela 1).

Tabela 1 - Estágios de acionamento do semáforo

Estágio	Sem. Car. 1	Sem. Ped. 1	Sem. Car. 2	Sem. Ped. 2
1				
	5 Segundos			
2				 Piscando
	3 Segundos			
3				
	5 Segundos			
4		 Piscando		
	3 Segundos			

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, 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 com pedestres */

/* Define as portas de todos os LEDs */
#define LED_Carro_Verde_1 13
#define LED_Carro_Amarelo_1 12
#define LED_Carro_Vermelho_1 11
#define LED_Pedestre_Verde_1 10
#define LED_Pedestre_Vermelho_1 9
#define GND_2 7
#define LED_Carro_Verde_2 6
#define LED_Carro_Amarelo_2 5
#define LED_Carro_Vermelho_2 4
#define LED_Pedestre_Verde_2 3
#define LED_Pedestre_Vermelho_2 2

void setup()
{
  /* Define os pinos de todos os LEDs como SAÍDA */
  pinMode(LED_Carro_Verde_1, OUTPUT);
  pinMode(LED_Carro_Amarelo_1, OUTPUT);
  pinMode(LED_Carro_Vermelho_1, OUTPUT);
  pinMode(LED_Pedestre_Verde_1, OUTPUT);
  pinMode(LED_Pedestre_Vermelho_1, OUTPUT);
  pinMode(LED_Carro_Verde_2, OUTPUT);
  pinMode(LED_Carro_Amarelo_2, OUTPUT);
  pinMode(LED_Carro_Vermelho_2, OUTPUT);
  pinMode(LED_Pedestre_Verde_2, OUTPUT);
  pinMode(LED_Pedestre_Vermelho_2, OUTPUT);
  pinMode(GND_2, OUTPUT);
  digitalWrite(GND_2, LOW);
}
```

```

void loop()
{
  /* Inicia o loop com todos os LEDs desligados */
  digitalWrite(LED_Carro_Verde_1, LOW);
  digitalWrite(LED_Carro_Amarelo_1, LOW);
  digitalWrite(LED_Carro_Vermelho_1, LOW);
  digitalWrite(LED_Pedestre_Verde_1, LOW);
  digitalWrite(LED_Pedestre_Vermelho_1, LOW);
  digitalWrite(LED_Carro_Verde_2, LOW);
  digitalWrite(LED_Carro_Amarelo_2, LOW);
  digitalWrite(LED_Carro_Vermelho_2, LOW);
  digitalWrite(LED_Pedestre_Verde_2, LOW);
  digitalWrite(LED_Pedestre_Vermelho_2, LOW);

  /* PRIMEIRO ESTÁGIO (5 SEGUNDOS).
  LED_Carro_Verde_1:      LIGADO
  LED_Carro_Amarelo_1:   DESLIGADO
  LED_Carro_Vermelho_1:  DESLIGADO
  LED_Pedestre_Verde_1:  DESLIGADO
  LED_Pedestre_Vermelho_1: LIGADO
  LED_Carro_Verde_2:     DESLIGADO
  LED_Carro_Amarelo_2:   DESLIGADO
  LED_Carro_Vermelho_2:  LIGADO
  LED_Pedestre_Verde_2:  LIGADO
  LED_Pedestre_Vermelho_2: DESLIGADO */
  digitalWrite(LED_Carro_Verde_1, HIGH);
  digitalWrite(LED_Carro_Amarelo_1, LOW);
  digitalWrite(LED_Carro_Vermelho_1, LOW);
  digitalWrite(LED_Pedestre_Verde_1, LOW);
  digitalWrite(LED_Pedestre_Vermelho_1, HIGH);
  digitalWrite(LED_Carro_Verde_2, LOW);
  digitalWrite(LED_Carro_Amarelo_2, LOW);
  digitalWrite(LED_Carro_Vermelho_2, HIGH);
  digitalWrite(LED_Pedestre_Verde_2, HIGH);
  digitalWrite(LED_Pedestre_Vermelho_2, LOW);
  delay(5000);

  /* SEGUNDO ESTÁGIO (3 SEGUNDOS).
  LED_Carro_Verde_1:      DESLIGADO
  LED_Carro_Amarelo_1:   LIGADO
  LED_Carro_Vermelho_1:  DESLIGADO
  LED_Pedestre_Verde_1:  DESLIGADO
  LED_Pedestre_Vermelho_1: LIGADO
  LED_Carro_Verde_2:     DESLIGADO

```

```
    LED_Carro_Amarelo_2:    DESLIGADO
    LED_Carro_Vermelho_2:   LIGADO
    LED_Pedestre_Verde_2:   DESLIGADO
    LED_Pedestre_Vermelho_2: PISCANDO */
digitalWrite(LED_Carro_Verde_1, LOW);
digitalWrite(LED_Pedestre_Verde_2, LOW);
digitalWrite(LED_Carro_Amarelo_1, HIGH);
for (int i = 1; i <= 3; i++) {
    digitalWrite(LED_Pedestre_Vermelho_2, HIGH);
    delay(500);
    digitalWrite(LED_Pedestre_Vermelho_2, LOW);
    delay(500);
}

/* TERCEIRO ESTÁGIO (5 SEGUNDOS).
LED_Carro_Verde_1:        DESLIGADO
LED_Carro_Amarelo_1:     DESLIGADO
LED_Carro_Vermelho_1:    LIGADO
LED_Pedestre_Verde_1:    LIGADO
LED_Pedestre_Vermelho_1: DESLIGADO
LED_Carro_Verde_2:       LIGADO
LED_Carro_Amarelo_2:     DESLIGADO
LED_Carro_Vermelho_2:    DESLIGADO
LED_Pedestre_Verde_2:    DESLIGADO
LED_Pedestre_Vermelho_2: LIGADO */
digitalWrite(LED_Carro_Verde_1, LOW);
digitalWrite(LED_Carro_Amarelo_1, LOW);
digitalWrite(LED_Carro_Vermelho_1, HIGH);
digitalWrite(LED_Pedestre_Verde_1, HIGH);
digitalWrite(LED_Pedestre_Vermelho_1, LOW);
digitalWrite(LED_Carro_Verde_2, HIGH);
digitalWrite(LED_Carro_Amarelo_2, LOW);
digitalWrite(LED_Carro_Vermelho_2, LOW);
digitalWrite(LED_Pedestre_Verde_2, LOW);
digitalWrite(LED_Pedestre_Vermelho_2, HIGH);
delay(5000);
```

```

/* QUARTO ESTÁGIO (3 SEGUNDOS).
LED_Carro_Verde_1:      DESLIGADO
LED_Carro_Amarelo_1:    DESLIGADO
LED_Carro_Vermelho_1:   LIGADO
LED_Pedestre_Verde_1:   DESLIGADO
LED_Pedestre_Vermelho_1: PISCANDO
LED_Carro_Verde_2:      DESLIGADO
LED_Carro_Amarelo_2:    LIGADO
LED_Carro_Vermelho_2:   DESLIGADO
LED_Pedestre_Verde_2:   DESLIGADO
LED_Pedestre_Vermelho_2: LIGADO */
digitalWrite(LED_Carro_Verde_2, LOW);
digitalWrite(LED_Pedestre_Verde_1, LOW);
digitalWrite(LED_Carro_Amarelo_2, HIGH);
for (int i = 1; i <= 3; i++) {
    digitalWrite(LED_Pedestre_Vermelho_1, HIGH);
    delay(500);
    digitalWrite(LED_Pedestre_Vermelho_1, LOW);
    delay(500);
}
}

```

Com o código-fonte inserido no Arduino IDE, compile o programa pressionando o botão **Verify** (botão com sinal de tique) para verificar se não há erros de sintaxe. Estando o código correto, o próximo passo é realizar a transferência do programa para o Arduino. Para tal, pressione o botão **Upload** (botão com uma seta apontando para a direita).

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 cruzamento de semáforos 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 do cruzamento de semáforos

```
quando o Arduino Uno começar
  definir a saída do pino digital 7 como baixo
  repetir para sempre
    definir a saída do pino digital 13 como alto
    definir a saída do pino digital 12 como baixo
    definir a saída do pino digital 11 como baixo
    definir a saída do pino digital 10 como baixo
    definir a saída do pino digital 9 como alto
    definir a saída do pino digital 6 como baixo
    definir a saída do pino digital 5 como baixo
    definir a saída do pino digital 4 como alto
    definir a saída do pino digital 3 como alto
    definir a saída do pino digital 2 como baixo
    esperar 5 segundo(s)
    definir a saída do pino digital 13 como baixo
    definir a saída do pino digital 12 como alto
    definir a saída do pino digital 3 como baixo
    repetir 3
      definir a saída do pino digital 2 como baixo
      esperar 0.5 segundo(s)
      definir a saída do pino digital 2 como alto
      esperar 0.5 segundo(s)
    definir a saída do pino digital 12 como baixo
    definir a saída do pino digital 11 como alto
    definir a saída do pino digital 10 como alto
    definir a saída do pino digital 9 como baixo
    definir a saída do pino digital 6 como alto
    definir a saída do pino digital 4 como baixo
    esperar 5 segundo(s)
    definir a saída do pino digital 10 como baixo
    definir a saída do pino digital 6 como baixo
    definir a saída do pino digital 5 como alto
    repetir 3
      definir a saída do pino digital 9 como baixo
      esperar 0.5 segundo(s)
      definir a saída do pino digital 9 como alto
      esperar 0.5 segundo(s)
```

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 cruzamento de semáforos, 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 mudar os valores dos delays para dar uma variada no tempo que os pedestres têm para atravessar? Vamos encarar esse desafio e ver quais resultados conseguimos!

ii. Que tal também projetar a construção de seu semáforo fora da placa de prototipagem, simulando o cruzamento de duas ruas?



E se...

i. O projeto não funcionar?

1. Verifique se os módulos semáforo foram corretamente conectados 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.

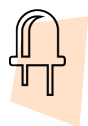
b. Analise seu projeto desenvolvido, de modo a atender aos requisitos para funcionamento de um semáforo: sequência correta do acendimento das cores verde, amarelo e vermelho para os veículos e pedestres.

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/a12roboticamod>

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

