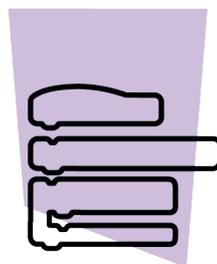
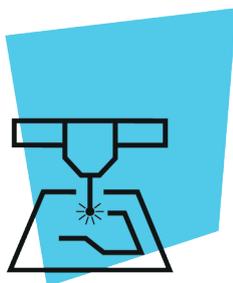
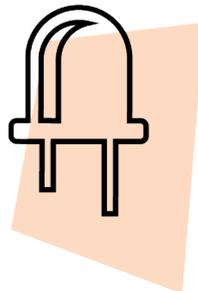
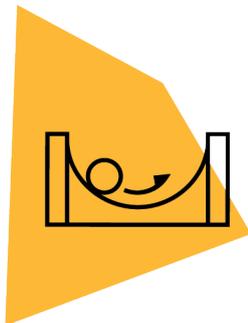
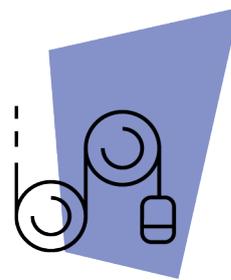
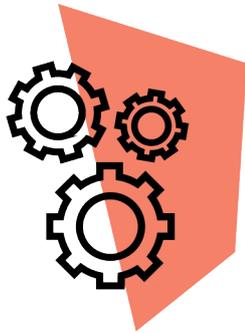


# ROBÓTICA

Módulo 1



## Sensor de Estacionamento

AULA 35

## **GOVERNADOR DO ESTADO DO PARANÁ**

Carlos Massa Ratinho Júnior

## **SECRETÁRIO DE ESTADO DA EDUCAÇÃO**

Renato Feder

## **DIRETOR DE TECNOLOGIA E INOVAÇÃO**

Andre Gustavo Souza Garbosa

## **COORDENADOR DE TECNOLOGIAS EDUCACIONAIS**

Marcelo Gasparin

### **Produção de Conteúdo**

Cleiton Rosa

Michelle Santos

Simone Sinara de Souza

### **Revisão Textual**

Adilson Carlos Batista

### **Projeto Gráfico e Diagramação**

Edna do Rocio Becker

2021



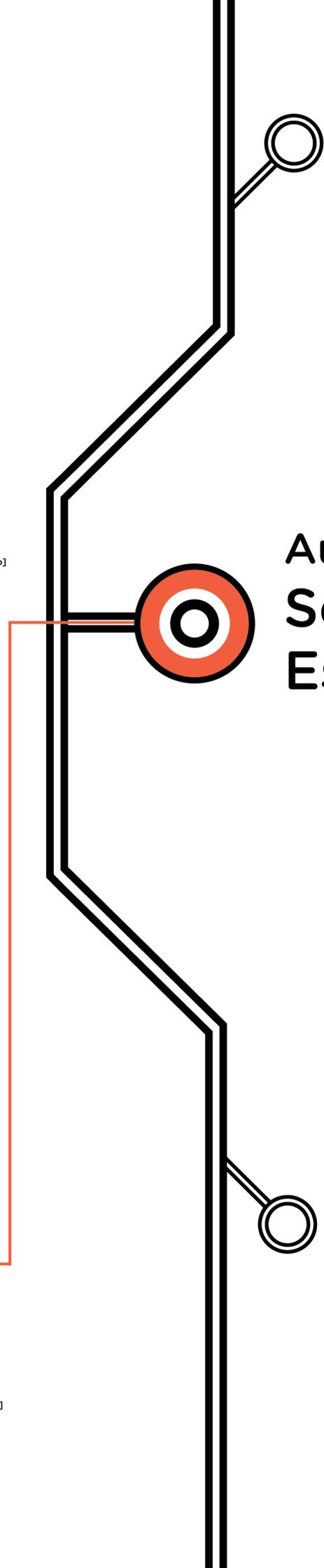
Este trabalho está licenciado com uma Licença Creative Commons  
Atribuição NãoComercial - Compartilha Igual 4.0 Internacional

	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-Íris	
	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 34  
Sensor de Distância

Aula 35  
Sensor de Estacionamento

Aula 36  
Display LCD 16x2



# Sumário

<b>Introdução</b>	<b>2</b>
<b>Objetivos desta Aula</b>	<b>2</b>
<b>Competências Gerais Previstas na BNCC</b>	<b>3</b>
<b>Habilidades do Século XXI a Serem Desenvolvidas</b>	<b>4</b>
<b>Lista de Materiais</b>	<b>4</b>
<b>Roteiro da Aula</b>	<b>5</b>
1. Contextualização	5
2. Montagem e Programação	6
3. Feedback e Finalização	12
<b>Videotutorial</b>	<b>14</b>

# 35 SENSOR DE ESTACIONAMENTO



## Introdução

Um dos equipamentos que mais ganhou destaque no meio automotivo ultimamente é o sensor de estacionamento, também conhecido como sensor de ré. Este dispositivo sinaliza, através de alertas sonoros, a proximidade de obstáculos facilitando ao condutor a realização de manobras, tanto para estacionar, como na percepção de objetos que estejam atrás dos veículos reduzindo o risco de colisão, proporcionando mais conforto e segurança ao motorista.



## Objetivos desta Aula

- Rever o funcionamento do sensor de distância ultrassônico associado a um Buzzer passivo;
- Programar o sensor de distância ultrassônico com a placa Arduino com nova configuração;
- Entender as novas possibilidades de aplicação do sensor de distância em programações com Arduino para estacionar veículos.



## 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.



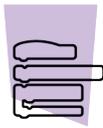
## Habilidades do Século XXI a Serem Desenvolvidas

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



## Lista de Materiais

- 01 Placa Protoboard;
- 01 Placa Arduino Uno R3;
- 01 Cabo USB;
- 01 Sensor Ultrassom HC-SR04;
- 01 Buzzer passivo;
- 08 Jumpers Macho-Macho;
- 01 Notebook;
- Software Arduino IDE ou mBlock.



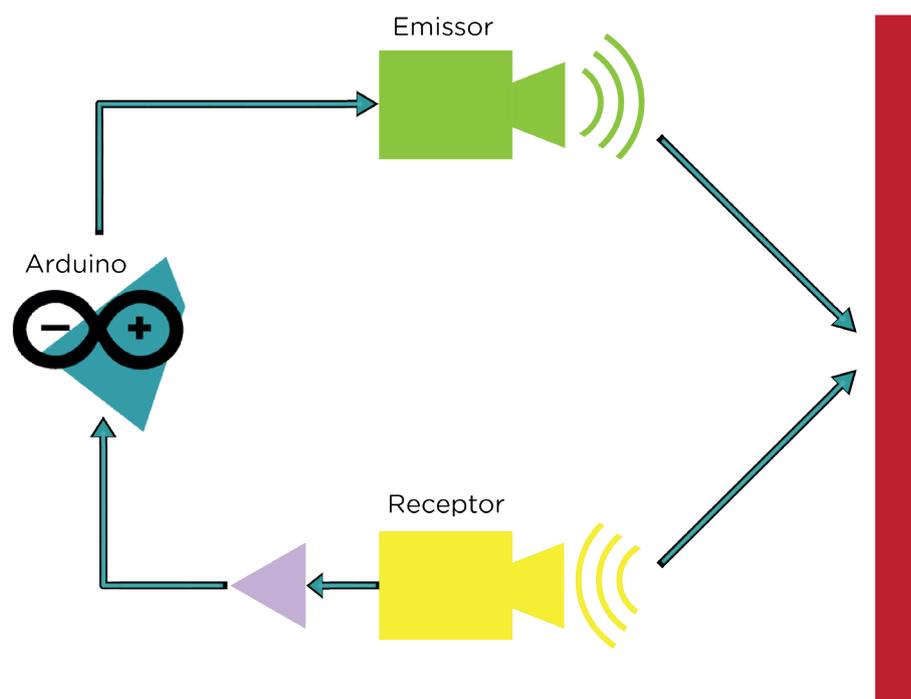
## Roteiro da Aula

### 1. Contextualização (15min):

Conforme estudado na aula anterior (**Aula 34 - Sensor de Distância**), os sensores de distância são encontrados em muitos lugares e podem servir tanto como sensor de proximidade, quanto para ajudar um robô a desviar de determinado objeto. O mesmo princípio utilizado nesses sensores, também é utilizado na medicina diagnóstica, mais precisamente no exame de ultrassonografia e na automação industrial. Neste projeto, utilizaremos um sensor ultrassônico para simular um sensor de estacionamento, encontrado em carros mais novos.

Lembrando que o sensor se baseia no envio e recebimento de ondas ultrassônicas, tomando como referência o tempo que o sinal demora para atingir um objeto e voltar. Definindo assim, qual a distância que se encontra o sensor e o obstáculo detectado, conforme mostra a figura 1.

Figura 1 - Funcionamento do sensor ultrassônico

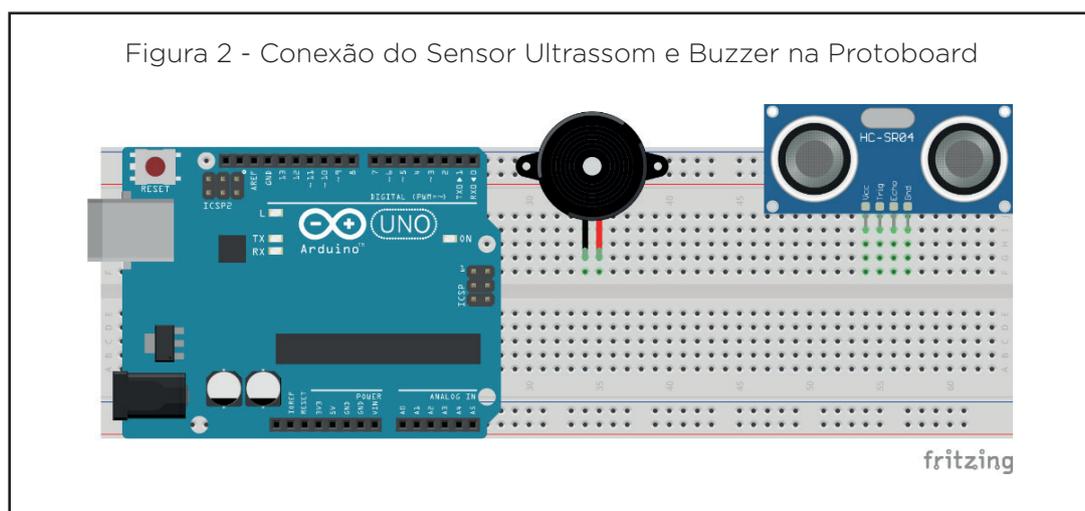


Nesta aula, utilizaremos o **Sensor Ultrassom HC-SR04**, o qual emite um sinal ultrassônico com velocidade correspondente a velocidade do som, que é de aproximadamente 340 m/s, assim, se o sensor estiver a uma distância  $x$  do objeto, o sinal percorrerá uma distância equivalente a  $2x$ , ou seja, a onda é enviada pelo sensor e rebatida no obstáculo, logo, ela percorre 2 vezes a distância procurada.

O alerta sonoro emitido pelo Sensor Ultrassom é resultante do componente eletrônico Buzzer acoplado a ele. Conforme estudamos na **Aula 24 - Buzzer Passivo**, é um componente eletrônico que atua como um mini autofalante e uma das possíveis aplicações dele é funcionar como alarme sonoro para indicar a proximidade com outros automóveis. Isto acontece porque o Buzzer possui um circuito oscilador que produz o som e só necessita ser energizado. Quanto mais perto o sensor se aproximar, maior a frequência do bip.

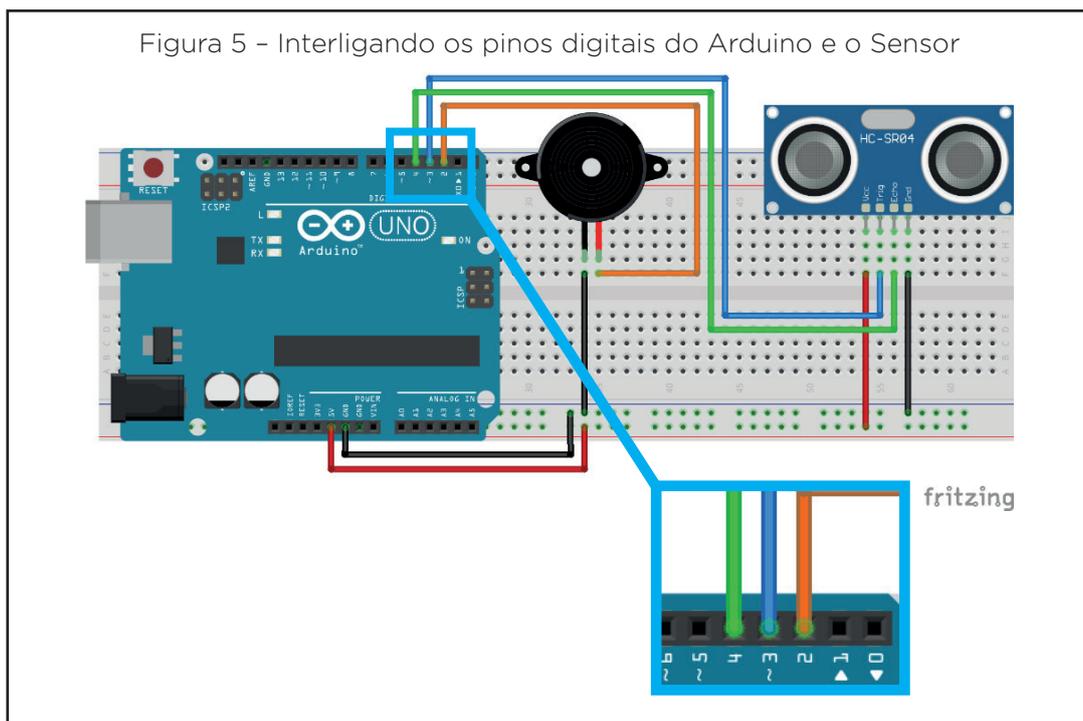
## 2. Montagem e Programação (60min):

Vamos começar com a montagem dos componentes eletrônicos. Encaixe na Protoboard o Buzzer e o Sensor Ultrassom utilizando uma coluna de furos para cada pino do sensor, conforme indicado na figura 2.





Por fim, utilizando mais 2 jumpers, conecte os pinos digitais 3 e 4 da placa de Arduino aos pinos Trig e Echo do Sensor, respectivamente, conforme a figura 5.



Agora, vamos programar!

Com os componentes eletrônicos montados, vamos programar, por código e por blocos, o funcionamento do sensor de estacionamento.

#### i. Linguagem de programação por código

Para esta programação, será necessário a biblioteca Ultrasonic instalada no Software Arduino IDE, (caso não esteja instalada, consulte a **Aula 05 - Softwares Arduino IDE e mBlock**).

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, conforme apresentado no quadro 1.

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

```
/* Programa: Sensor de Estacionamento. */

/* Inclui a biblioteca do sensor. */
#include <Ultrasonic.h>

/* Define os pinos para o buzzer e o sensor. */
int pino_buzzer = 2;
int pino_Trig = 3;
int pino_Echo = 4;

/* Especifica o nome e os pinos para o sensor. */
Ultrasonic Sensor(pino_Trig, pino_Echo);

/* Variável que armazenará as medidas. */
int distancia;
/* Define a frequência do beep em Hertz. */
#define frequencia 3500

void setup() {
  /* Configura o pino do Buzzer como SAÍDA. */
  pinMode(pino_buzzer, OUTPUT);
}

void loop() {
  /* Realiza a medição e armazena na variável "distancia" */
  distancia = Sensor.read();

  /* Condições de distâncias para a intensidades do bip. */
  if (distancia < 80 && distancia > 50) {
    tone(pino_buzzer, frequencia, 100);
    delay(1000);
  }
  if (distancia < 50 && distancia > 30) {
    tone(pino_buzzer, frequencia, 100);
    delay(700);
  }
  if (distancia < 30 && distancia > 20) {
    tone(pino_buzzer, frequencia, 100);
    delay(300);
  }
  if (distancia < 20 && distancia > 10) {
    tone(pino_buzzer, frequencia, 100);
    delay(150);
  }
  /* Distância menor que 10 cm, o bip fica contínuo. */
  if (distancia < 10) {
    tone(pino_buzzer, frequencia);
  }
}
```

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, o projeto estará funcionando, em distâncias menores que 80 cm, o Buzzer emitirá bips e aumentará a frequência quanto menor for a distância.

## ii. Linguagem de programação por blocos

Outra forma de programar o sensor de estacionamento, é por meio da linguagem de programação que utiliza blocos de funções prontas, os quais representam comandos de programação. Vamos utilizar o software mBlock.

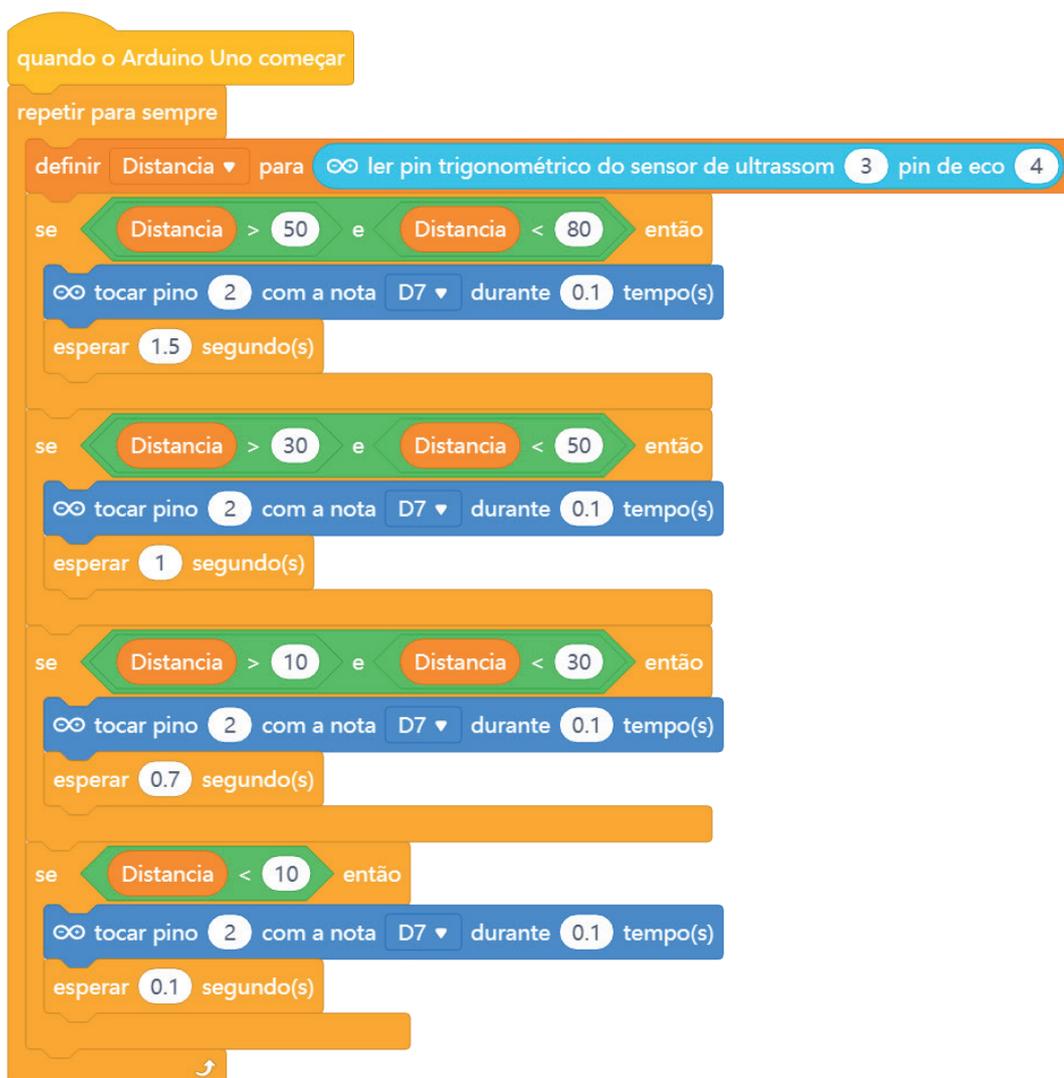
Para conectar o mBlock ao Arduino, você deve clicar no ícone **Adicionar**, localizado no campo **Dispositivos**, e selecionar o Arduino, na biblioteca de dispositivos do mBlock, clicando, na sequência, no botão **OK**.

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

Nesta programação, utilizaremos variáveis que auxiliarão na estrutura do nosso programa (para recordar como criar uma variável, consulte a **Aula 05 - Softwares Arduino IDE e mBlock**).

Monte os blocos, arrastando e soltando, de acordo com a programação de funcionamento do sensor de estacionamento, como mostra a figura 6.

Figura 6 - Programação em blocos do sensor de estacionamento



Assim que os blocos estiverem montados, clique no botão **Conectar** para iniciar a comunicação entre o software mBlock com a placa de 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 sensor de estacionamento.



### Desafios:

- i. Que tal tornar este projeto mais visual? Introduza uma Barra Gráfica de LEDs e programe-o para indicar a proximidade do obstáculo!
- ii. Outra sugestão é inserir um Display 16X2 e apresentar a distância do obstáculo em tempo real.



### E se... ?

- i. O projeto não funcionar, se atente a alguns dos possíveis erros:
  1. Verifique se os jumpers estão na mesma coluna dos terminais dos componentes, fazendo assim a conexão;
  2. Verifique se os jumpers estão ligados aos pinos corretos no Arduino;
  3. Verifique se a programação está adequada a cada porta digital;
  4. Verifique se o Buzzer não está conectado de modo invertido.

### 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 sensor de estacionamento.
- c. Reflita se as seguintes situações ocorreram:
  1. Colaboração e Cooperação: você e os membros de sua

equipe interagiram entre si, compartilhando ideias que promoveram a aprendizagem e o desenvolvimento deste projeto?

**2.** 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/a35robotica>

Acesse, também, pelo QRCode:

