



**Robótica**

**MATERIAL  
DIGITAL**

# **Reage, brinquedo! – Programe a diversão!**

**1º bimestre  
Aulas 5 e 6**

**Ensino Fundamental:  
Anos Finais**

Secretaria da  
Educação



**SÃO PAULO**  
GOVERNO DO ESTADO

## Conteúdos

- Programação condicional com faixas de valores;
- Atuação automática com LEDs, motores ou sirenes;
- Resposta reativa a múltiplas variáveis.

## Objetivos

- Desenvolver sistemas automatizados que respondem a diferentes níveis de sensores com múltiplas reações.

## Para começar



Carros autônomos já são uma realidade!

© Getty Images

## Carros autônomos

Nas aulas passadas, quem “decidia” o que o robô fazia? **Éramos nós**, programando cada passo, cada curva, cada parada!

E se a gente desse um passo além? Vocês já ouviram falar em **carros que tomam decisões por conta própria?**



# Para começar



Carros autônomos já são uma realidade!

© Getty Images



VIREM E CONVERSEM

## Carros autônomos

Como vocês acham que esses carros funcionam? Junte-se com um colega e discutam:

- Como vocês acham que o carro faz para virar ou andar para frente?
- E para desviar de obstáculos, como pessoas e outros carros?

# Qual será o próximo desafio de robótica?

Nossa nova missão será transformar o nosso carrinho em um **brinquedo inteligente**, que reage sozinho ao ambiente. Para isso, vamos:

- aprender a usar **sensores com programação condicional** (se/então/senão);
- testar diferentes **faixas de valores** (ex.: quando o objeto está bem perto ou bem longe);
- criar **efeitos visuais** para deixar o carrinho mais divertido e interativo.



Mas o que significa programação condicional e faixa de valores? Vejamos alguns exemplos reais!

# Exemplos de dispositivos com reação automática e sensores



## Aspiradores robôs

Eles desviam ao encontrar um obstáculo, como um móvel, parede ou animal.



## Portas automáticas

Elas só abrem se uma pessoa ou animal se aproximar.



## Alarmes de presença

Só disparam quando ativados e detectam movimentos.

# Sistema autônomo

**Destaque**



Um sistema autônomo é aquele que consegue perceber o ambiente e reagir sozinho, sem precisar de comandos o tempo todo.



**Comandado:** só faz algo ao **receber uma ordem direta.**

**Ex.:** um carrinho que anda quando um botão da micro:bit é apertado.



**Autônomo:** **reage sozinho** de acordo com o que detecta.

**Ex.:** um carrinho que freia automaticamente quando encontra um obstáculo.





**Pause e responda**

## Sistemas autônomos

Qual dos dispositivos a seguir não é um exemplo de sistema autônomo?

Semáforo que fecha quando o pedestre aperta um botão.

Semáforo que adapta os tempos de parada de acordo com o tráfego.

Aspirador de pó robótico com mapeamento e retorno automático à base.

Drone de entrega que desvia de obstáculos e recalcula a rota em tempo real.





**Pause e responda**

## Sistemas autônomos

Qual dos dispositivos seguir não é um exemplo de sistema autônomo?

**Semáforo que fecha quando o pedestre aperta um botão.**

**Semáforo que adapta os tempos de parada de acordo com o tráfego.**

**Aspirador de pó robótico com mapeamento e retorno automático à base.**

**Drone de entrega que desvia de obstáculos e recalcula a rota em tempo real.**

## Relembre

### O que são sensores?

Nós entendemos o mundo pelos nossos sentidos, como a visão, a audição e o tato. De forma parecida, os robôs também precisam “sentir” o ambiente. Para isso, eles usam sensores, que funcionam como os olhos, os ouvidos e a pele das máquinas. Assim, o robô consegue perceber o que está acontecendo ao seu redor e reagir de acordo com a situação.

#### Para refletir

Vocês se recordam dos sensores que usamos no ano passado? Quais eram suas funções?

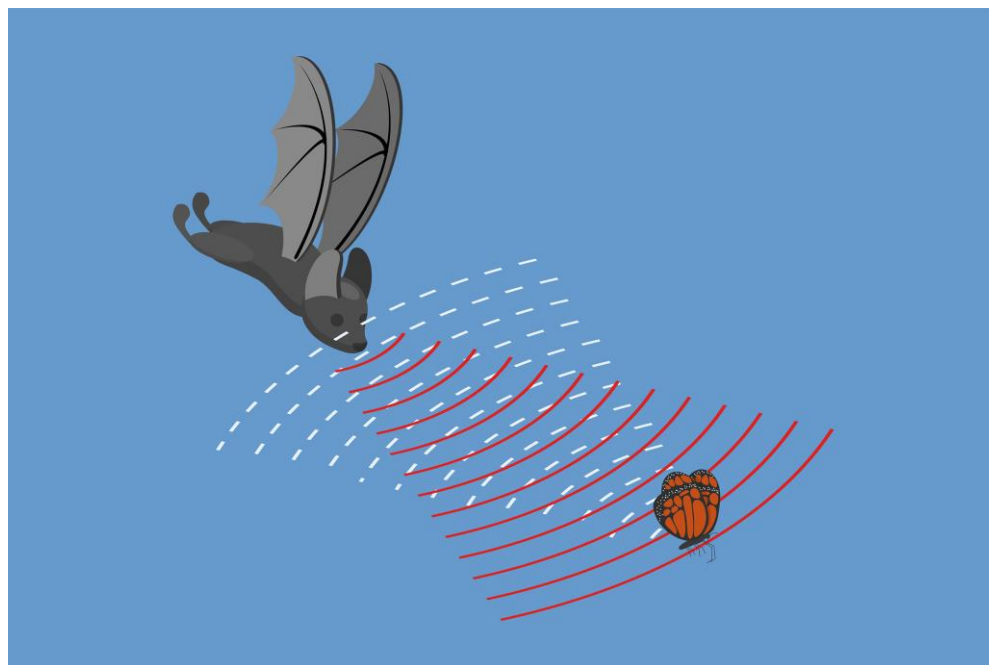


Produzido pela SEDUC-SP com apoio da ferramenta Google Gemini.

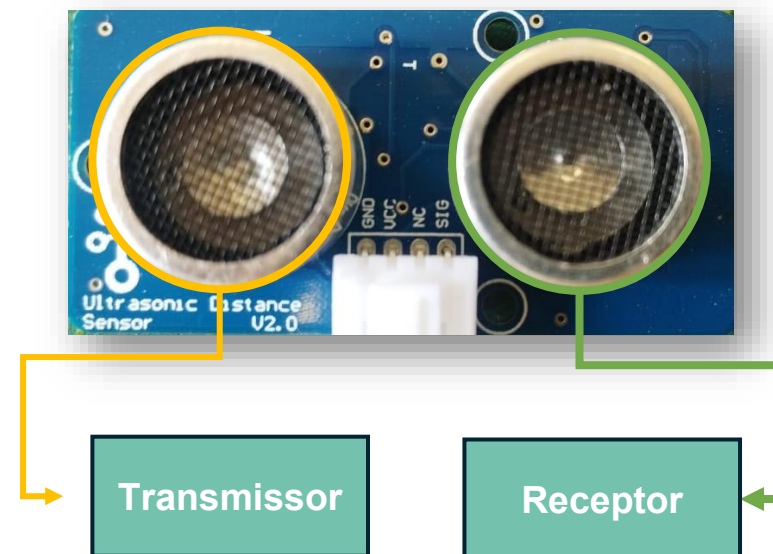
# Sensor ultrassônico

O sensor ultrassônico funciona como o sonar de um morcego: ele emite ondas sonoras e capta o eco de retorno.

Medindo esse tempo de ida e volta, o robô descobre a distância.



© Getty Images



Produzido pela SEDUC-SP com apoio da ferramenta Google Gemini.

### Condicionais (se/então/senão)

São usadas para decidir o que fazer com os dados captados. Por exemplo:

**Se** a distância for menor que 10 cm → **parar**. **Senão** → **continuar andando**.



Produzido pela SEDUC-SP com apoio da ferramenta Google Gemini.

**Destaque**



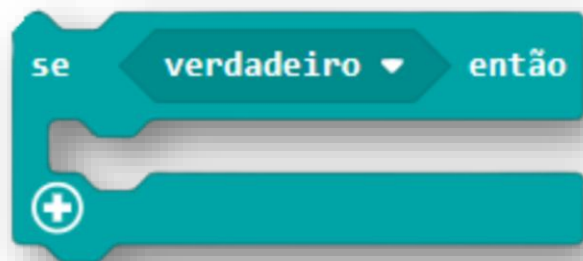
**Nem sempre é só “sim” ou “não”.**

Podemos programar diferentes **faixas de valores**, com cada uma gerando uma reação.

Isso permite criar robôs **mais inteligentes e interativos**, capazes de reagir de várias formas ao ambiente.



## Condicionais



Esse bloco permite executar ações **somente quando uma condição for verdadeira**.



Para situações com **duas ou mais opções**, você pode usar “**se ... então ... senão ...**”.

Produzido pela SEDUC-SP com a ferramenta Microsoft MakeCode.



Você encontra esses blocos na categoria “**Lógica**”.

## Blocos comparativos

Igual a (=)

Maior que (>)

Menor que (<)

É possível combinar várias condições usando operadores lógicos como “e” (“and”), “ou” (“or”) e “não” (“not”).



Produzido pela SEDUC-SP com a ferramenta Microsoft MakeCode.

Para refletir

O robô pode comparar leituras de sensores (como luz, temperatura ou distância) ou até mesmo valores guardados em variáveis. Assim, o robô reage ao que está acontecendo no ambiente em tempo real.

**Em quais projetos do ano passado utilizamos esse recurso?**

## Na prática

### Nosso foco:

- Blocos de condição;
- Blocos de comparação.

Usaremos esses blocos para encarar o desafio de dar ao nosso carro de brinquedo autonomia de movimento.



Produzido pela SEDUC-SP com a ferramenta Microsoft MakeCode.

## Na prática

# Ferramentas para programar nosso carro

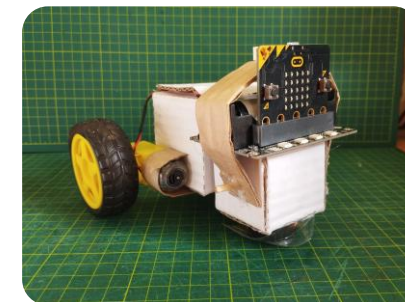
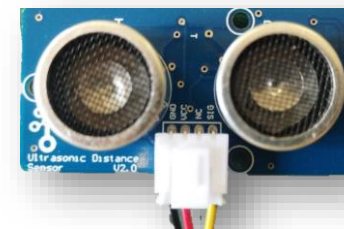
- *Notebook*;
- Sensor ultrassônico;
- Micro:bit;
- O protótipo do carro;
- *Shield*;
- Cabo USB.



### FICA A DICA

#### Cuidados com os *kits*:

- Sugerimos que cada grupo use **sempre o mesmo kit** → numerar para facilitar.
- **Mãos limpas e secas** ao manusear.
- **Não arremessar** os componentes.
- Faltando peças? O docente irá organizar os grupos para que **compartilhem shields e micro:bits**.



Produzido pela SEDUC-SP com a ferramenta Canva.



## Plataforma MakeCode

<https://makecode.microbit.org>

Clique em “novo projeto” e nomeie-o com o nome de sua preferência.

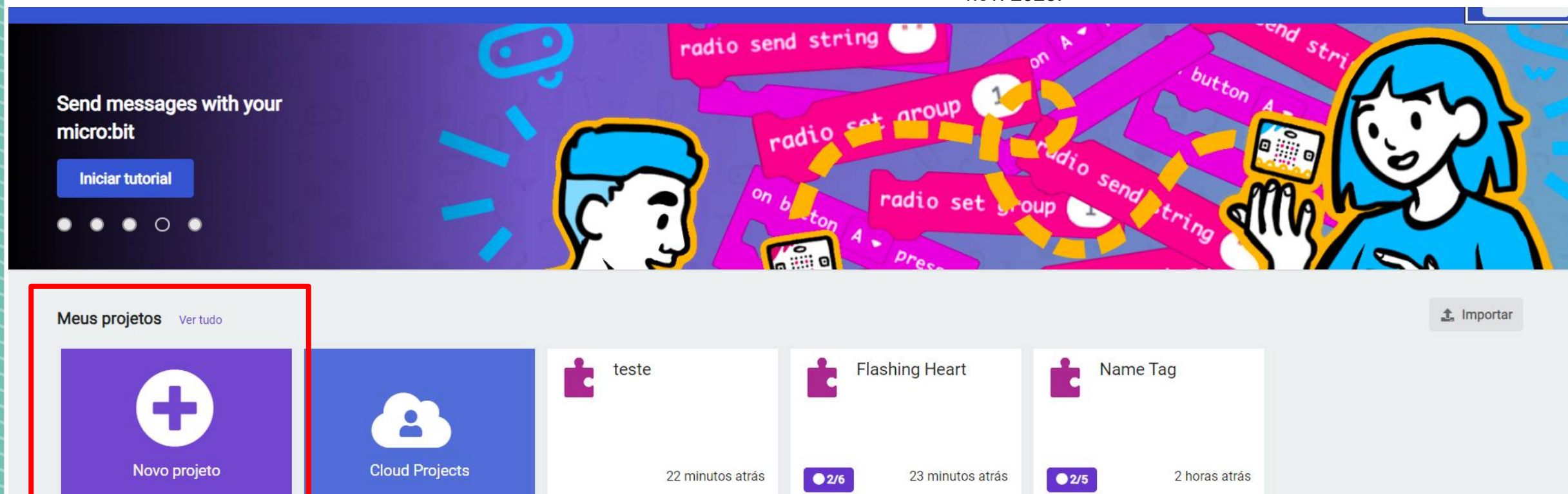


FICA A DICA

**Esqueceu como entrar no MakeCode? Ele também é acessível pela Sala do Futuro!**

Página inicial da plataforma Microsoft MakeCode

Disponível em: <https://makecode.microbit.org>. Acesso em: 12 nov. 2025.





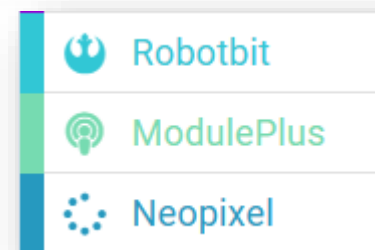
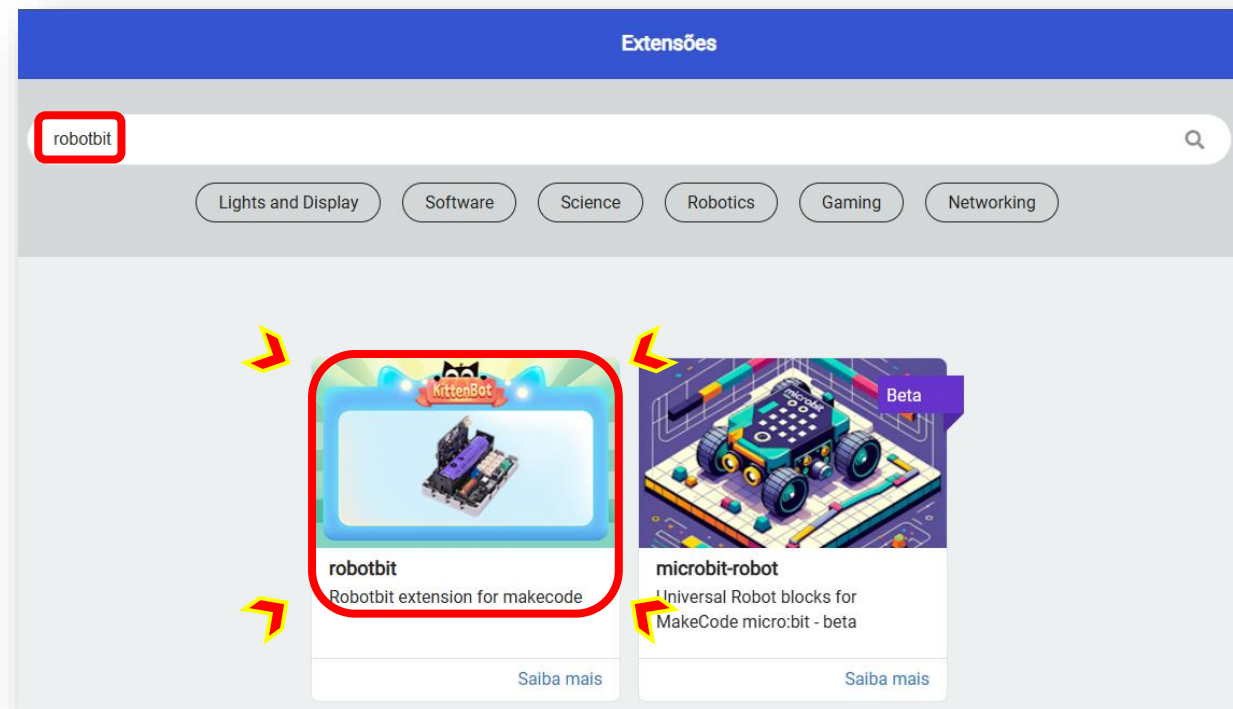
## Micro:bit: sensor e motor

Antes de começar, você precisa adicionar a extensão **robotbit** do MakeCode em seu projeto.

A. Clique em:  Extensões

B. Digite **robotbit** e adicione a extensão ao seu projeto.

Isso habilitará os blocos de controle de motores, servos e outros componentes da *shield*.

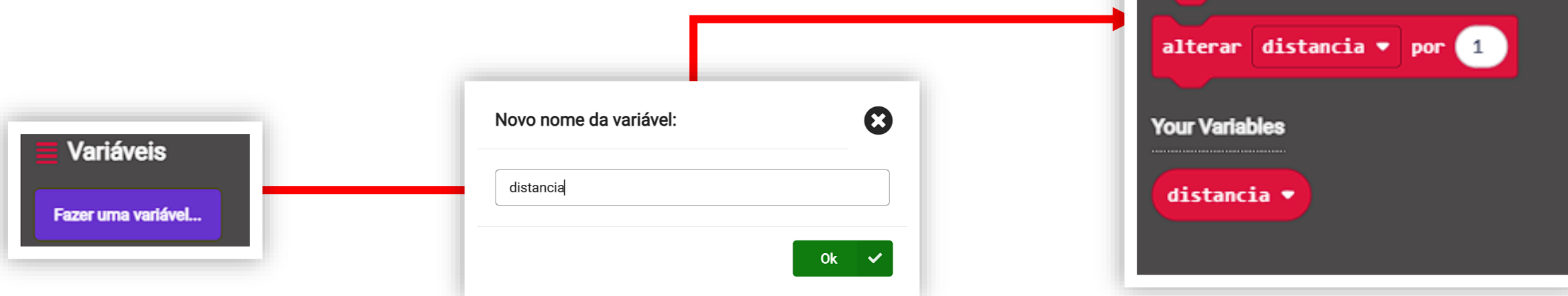




## Micro:bit: variável “distancia”

Vá até a categoria “**Variáveis**” e clique em “**Fazer uma nova variável...**”.

Perceba que serão criados 3 blocos para sua variável.





## Micro:bit: definir a variável “distancia” para a leitura do sensor

- Vá até a categoria “**Variáveis**”, clique em **definir distancia** para **0** e encaixe-o dentro do bloco “**Sempre**”.
- No lugar do “**0**”, você deve colocar **Ultrasonic(with RGB) pin P1**. Clique na extensão **robotbit** e procure por esse bloco em “Modules”. Encaixe-o dentro do espaço que contém o número “0”. Altere o pino “P1” para “P2”.







### Micro:bit: definir a variável “distancia” para a leitura do sensor



Produzido pela SEDUC-SP com a ferramenta Microsoft MakeCode.

**Destaque**



Até agora, o código instrui o robô a ler continuamente a distância de um objeto usando um **sensor ultrassônico** conectado à **porta P2**.

A medição, feita em centímetros, é armazenada em uma variável chamada de “**distancia**”, que é atualizada constantemente para refletir a distância em tempo real.



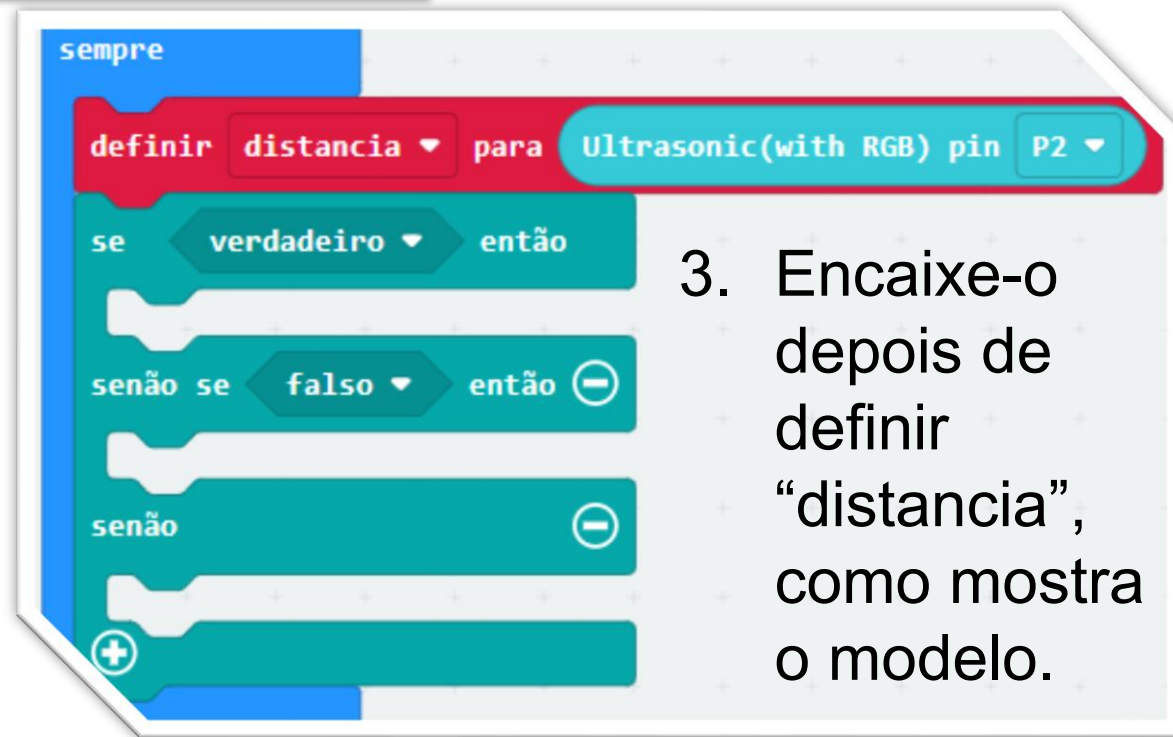
## Micro:bit: inserindo a condicional

1. Em “Lógica”, encontre o bloco:



Perceba que só há espaço para 2 condições. No entanto, queremos 3!

2. Clique no sinal de “mais” para adicionar mais uma condição. Ele ficará assim:




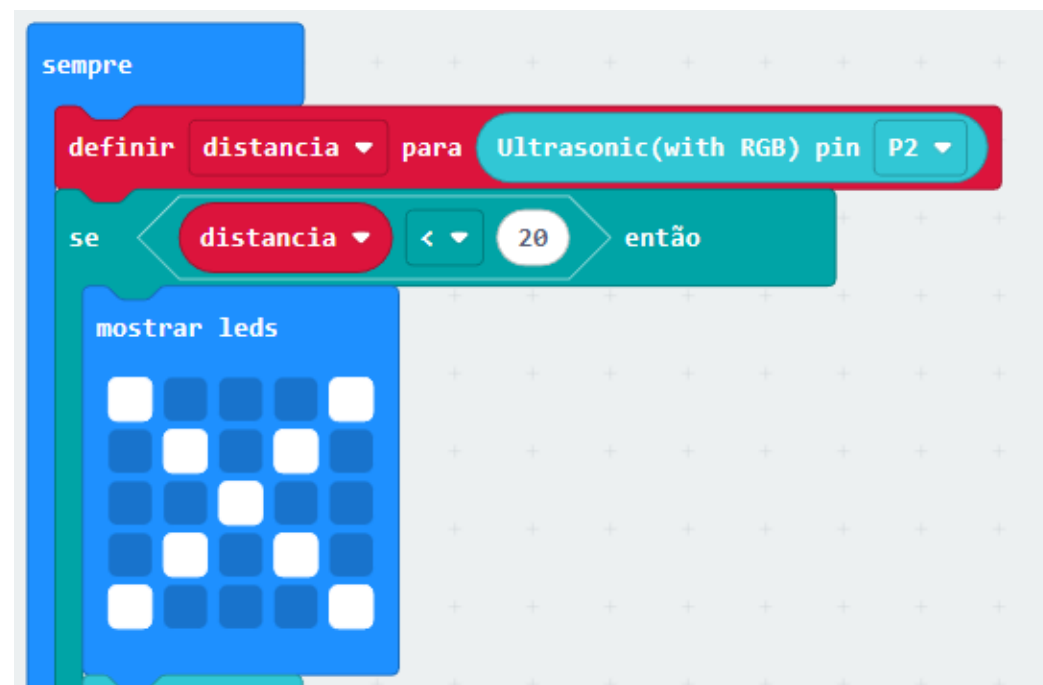
3. Encaixe-o depois de definir “distancia”, como mostra o modelo.



### Micro:bit: 1ª condição e ação

A primeira condição avisará se o obstáculo está perto. Nesse caso, o carro deverá se deslocar para trás.

- Vamos, então, até a lógica para buscar um bloco de comparação: 
- Encaixe o bloco em cima de “verdadeiro” e compare 2 elementos: se a variável “distancia” for menor que 20 cm, mostre um X nos LEDs.

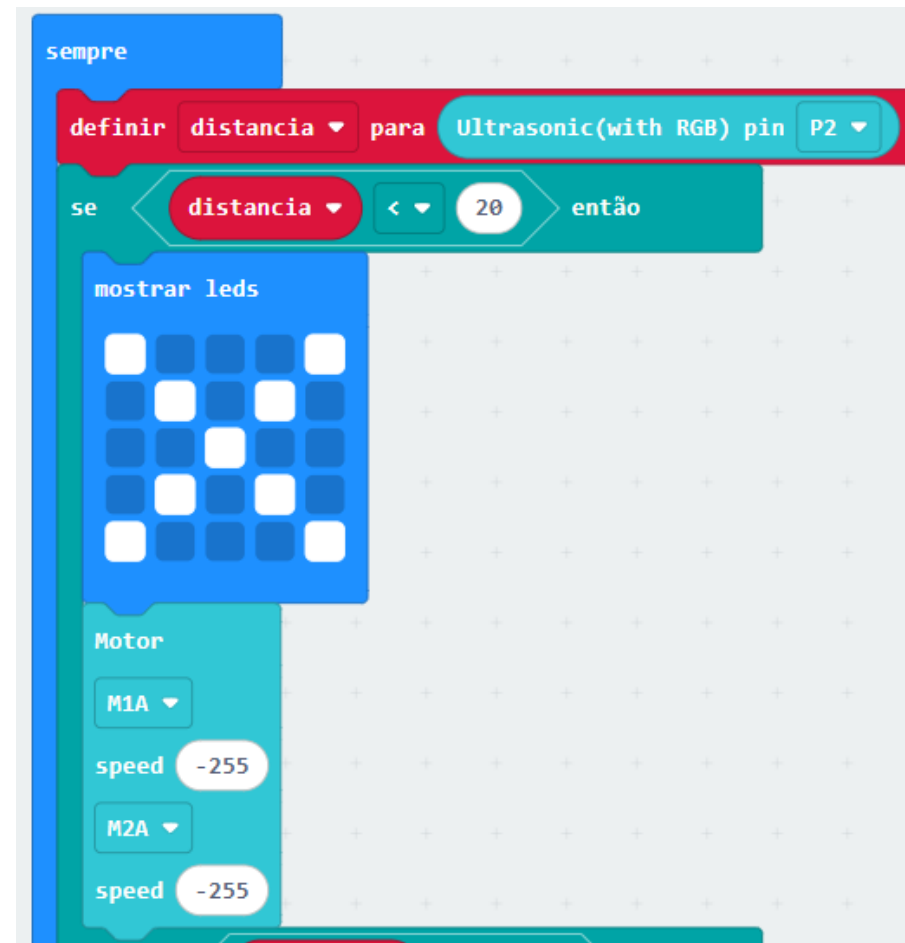


Imagens: Produzido pela SEDUC-SP com a ferramenta Microsoft MakeCode.



## Micro:bit: 1ª condição e ação

- Ache nossa variável **distancia** e encaixe-a dentro do primeiro “0”; mantenha o sinal “menor que”. Enfim, coloque “20” no lugar de “0”.
- Para concluir, vá até a categoria “Robotbit” e selecione o bloco que controla os dois motores simultaneamente.
- Arraste-o para logo abaixo do bloco “mostrar LEDs”. Em seguida, configure o segundo motor para M2A e ajuste a velocidade de ambos para o valor máximo, - 255.



Imagens: Produzido pela SEDUC-SP com a ferramenta Microsoft MakeCode.

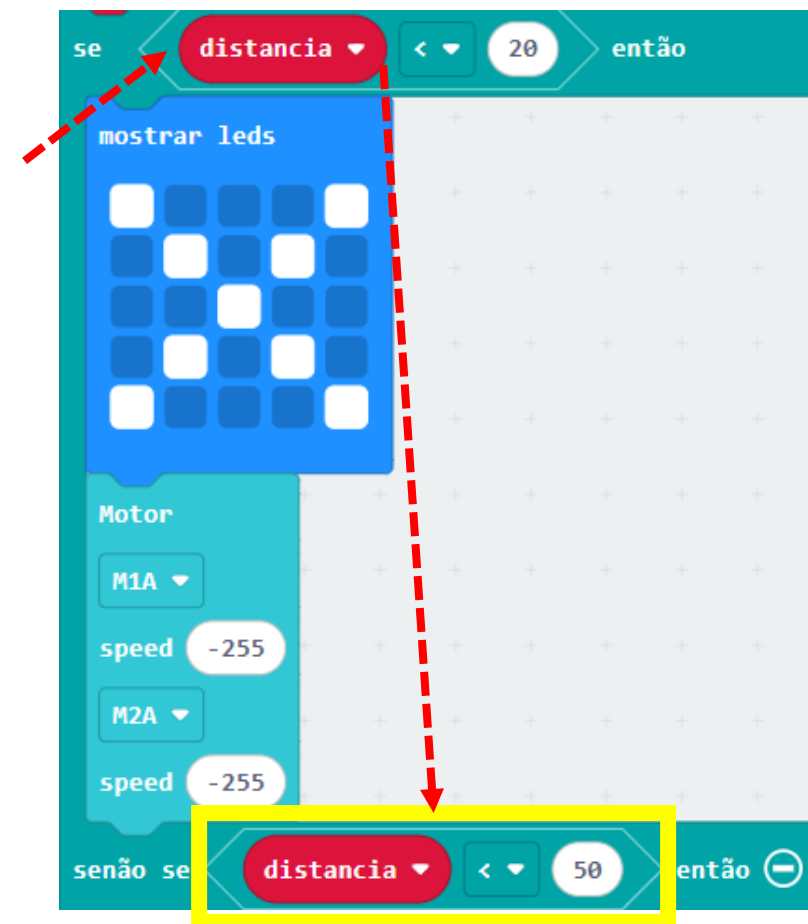




## Micro:bit: 2ª condição e ação

A segunda condição servirá para indicar quando o obstáculo estiver a uma distância intermediária, fazendo o carro girar uma das rodas e, assim, mudar de direção.

1. Duplica o primeiro bloco de comparação:
  - Clique com o botão direito sobre o bloco.
  - Selecione “**Duplicar**”.
  - Encaixe o novo bloco no espaço de “**falso**”.
2. Configure o bloco duplicado:
  - Altere o operador para “menor que”.
  - Substitua o valor “0” por “50”.





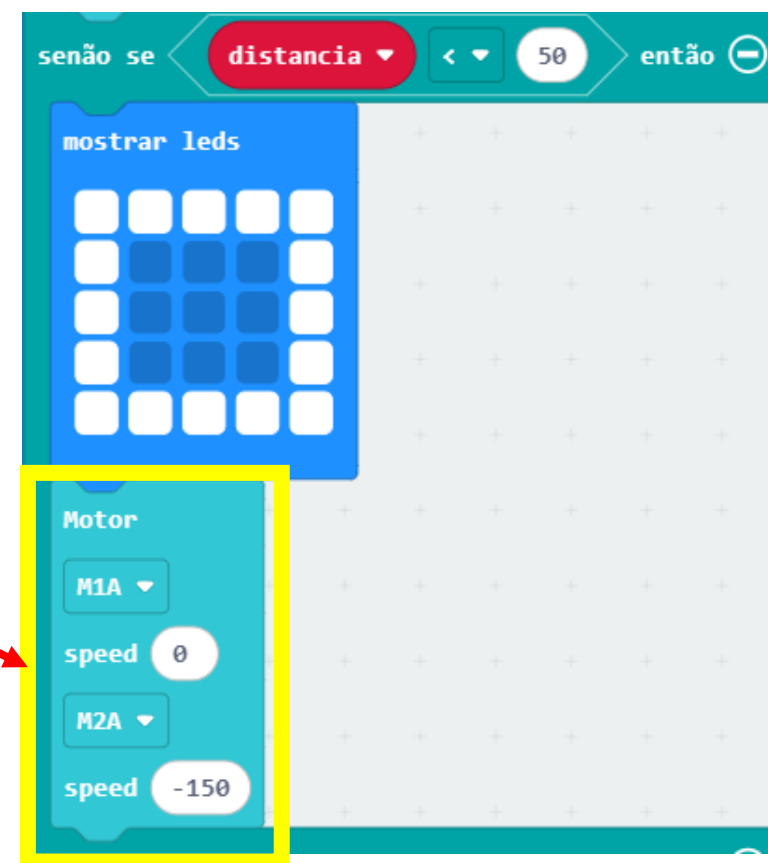
### Micro:bit: 2ª condição e ação

#### 3. Defina a ação:

- Mostre um quadrado nos LEDs.
- Faça com que os motores funcionem em velocidade diferente. Para isso, encontre os blocos que fazem 2 motores funcionarem juntos, na categoria “Robotbit”. Altere a velocidade (*speed*) do “M1A” para “0” e a do “M2A” para “-150”.

#### Destaque

Se a distância for menor ou igual a 50 cm, o carro exibirá a seta e rotacionará para mudar de direção.



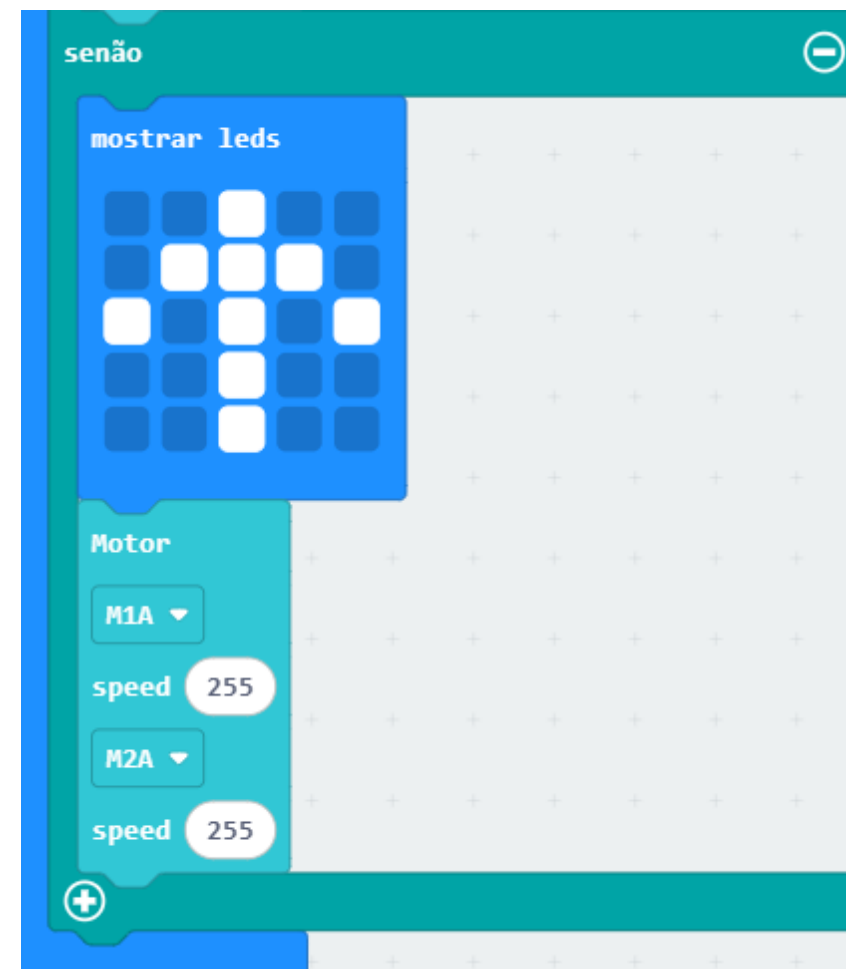


### Micro:bit: 3ª condição e ação

Por fim, a última condição, a condição “senão”, medirá se a distância é maior que 50 cm, sinalizando com uma seta para cima e avisando que é seguro se deslocar para frente.

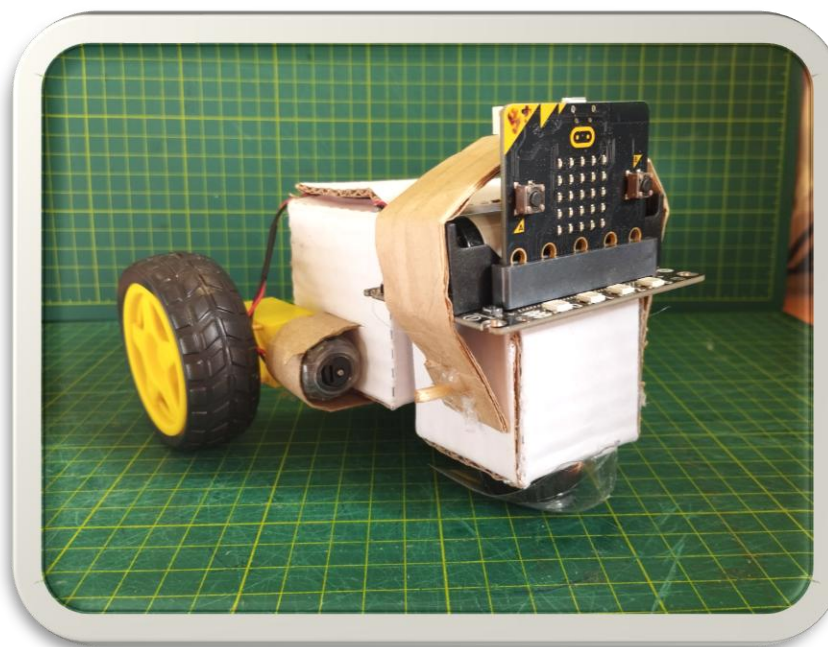
**4.** Na categoria “Básico”, puxe e encaixe o bloco “mostrar leds” embaixo do “senão” e desenhe uma seta para cima.

**5.** Depois, duplique o bloco do motor da condição anterior e encaixe-o embaixo de “mostrar leds”. Não esqueça de alterar ambas as velocidades para “255”.



### Micro:bit: código completo

Agora, temos o código completo! Já podemos baixá-lo para a placa, conectar o sensor na *shield* e testar. Durante o teste, observem se a matriz de LEDs está exibindo os ícones corretos, de acordo com cada situação.



Produzido pela SEDUC-SP

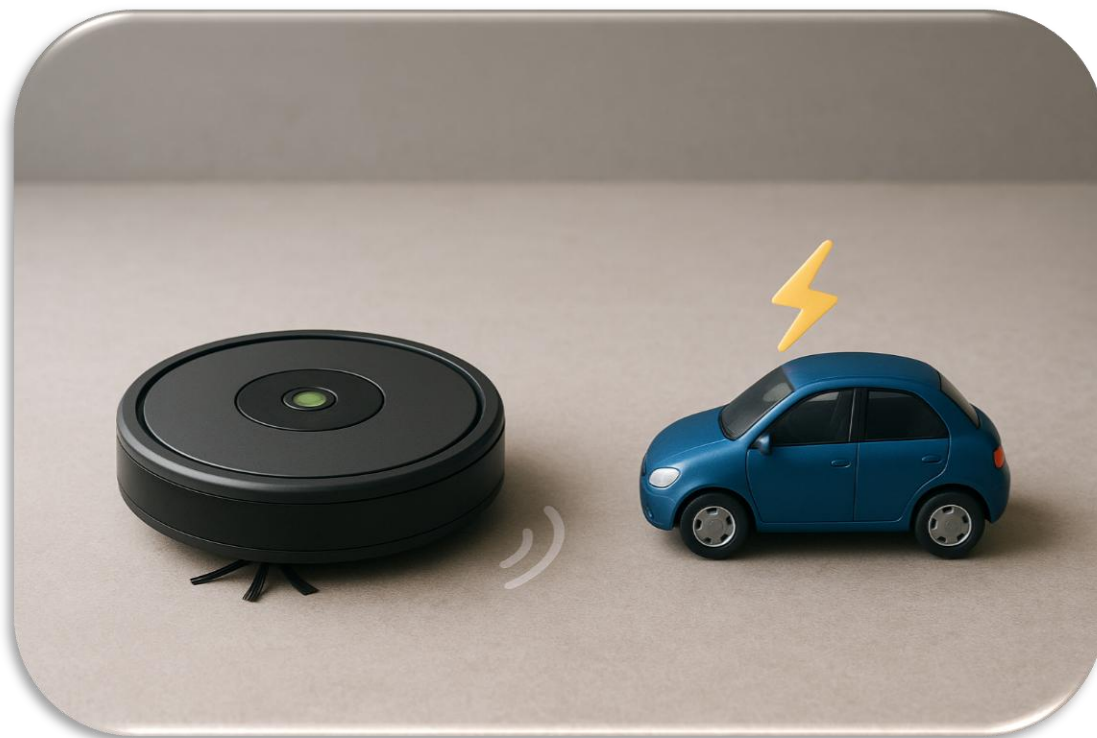


Produzido pela SEDUC-SP com a ferramenta Microsoft MakeCode.

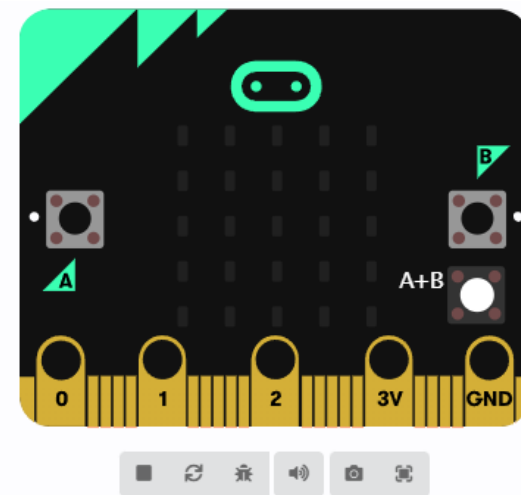




Essa programação é um ótimo exemplo do funcionamento de dispositivos **robóticos autônomos**. Pode ser usada como base em projetos de robôs aspiradores, carrinhos de bate-bate, brinquedos que interagem com o ambiente etc.



Produzido pela SEDUC-SP com apoio da ferramenta ChatGPT.



Produzido pela SEDUC-SP com a ferramenta Microsoft MakeCode.

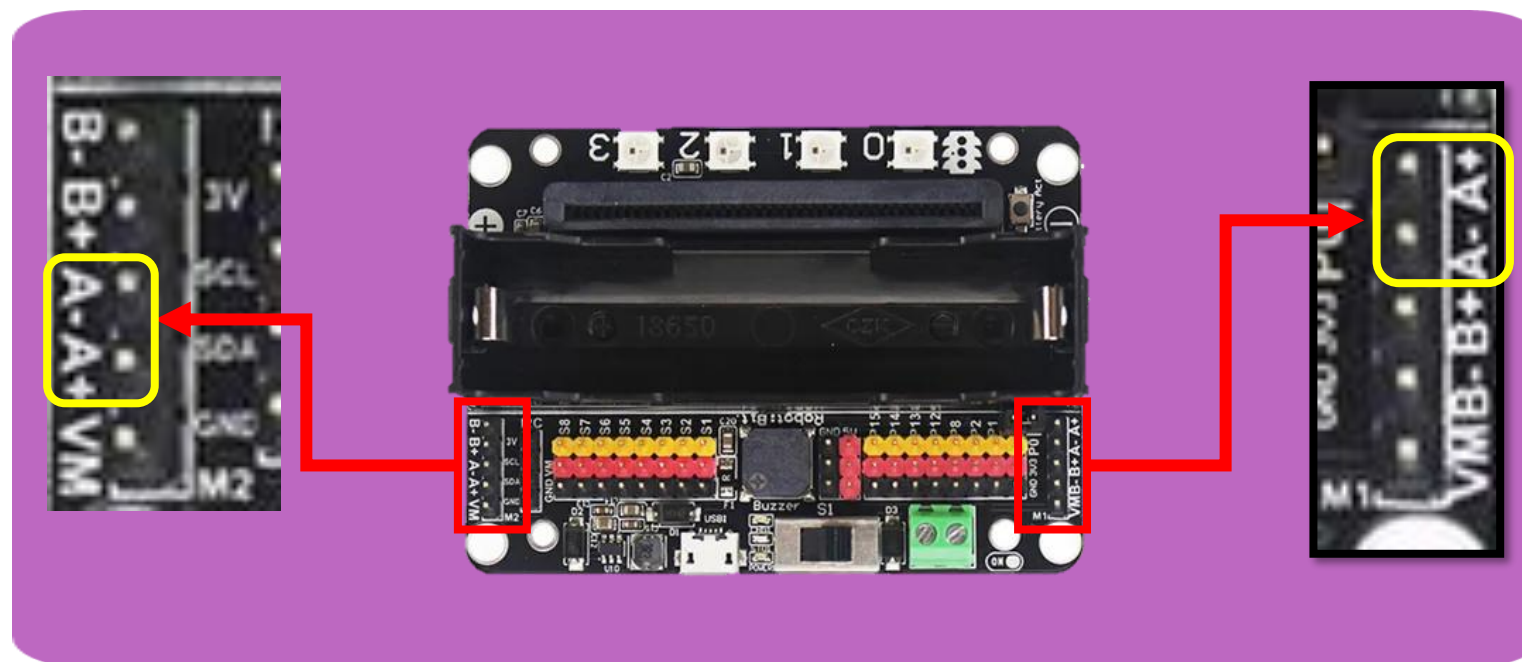
# Ligando os motores e o sensor ultrassônico na *shield*

Conecte os cabos de um os motores nos pinos **M1A** (positivo e negativo) e os cabos do outro motor nos pinos **M2A** (positivo e negativo).



### FICA A DICA

Caso os motores não girem no sentido desejado, inverta a conexão dos cabos (polaridade) nos pinos.



Produzido pela SEDUC-SP com a ferramenta Canva.





## Micro:bit: encaixando o sensor e o motor na *shield* e no carro



### FICA A DICA

É importante que cada grupo divida as tarefas para facilitar a prototipagem e tornar o trabalho mais eficiente.

### Automatizando nosso carrinho

**Nesta aula**, demos vida ao nosso carrinho: sensores, condicionais e decisões automáticas! Vocês provaram que conseguem transformar código em **inteligência no movimento**.

- Se o carro precisasse frear suavemente em vez de parar de repente, como você adaptaria os valores e velocidades dos motores na programação?
- Imagine que você queira que o carro também dê um aviso sonoro (*buzzer* ou música) quando o obstáculo estiver muito perto. Onde e como você incluiria essa condição no código?



## Referências

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Brasília (DF), 2018. Disponível em: [https://www.gov.br/mec/pt-br/escola-em-tempo-integral/BNCC\\_EI\\_EF\\_110518\\_versaofinal.pdf](https://www.gov.br/mec/pt-br/escola-em-tempo-integral/BNCC_EI_EF_110518_versaofinal.pdf). Acesso em: 12 nov. 2025.

BRASIL. Ministério da Educação. **Computação**: complemento à BNCC. Brasília (DF), 2022. Disponível em: <https://www.gov.br/mec/pt-br/escolas-conectadas/BNCCComputaoCompletoDiagramado.pdf>. Acesso em: 12 nov. 2025.

CENTRO DE MÍDIAS SP. **Página inicial**, [s.d.]. Disponível em: <https://cmsp.ip.tv/>. Acesso em: 12 nov. 2025.

LEMOV, D. **Aula nota 10 3.0**: 63 técnicas para melhorar a gestão da sala de aula / Doug Lemov; tradução: Daniel Vieira, Sandra Maria Mallmann da Rosa; revisão técnica: Fausta Camargo, Thuinie Daros. 3. ed. Porto Alegre: Penso, 2023.

MICROSOFT MAKECODE. **Página inicial**, [s.d.]. Disponível em: <https://makecode.microbit.org>. Acesso em: 07 jan. 2026.

## Referências

OPENCADD. **Entenda como a robótica e os sistemas autônomos podem ser usados para melhorar a sustentabilidade**, 19 maio 2023. Disponível em: <https://www.opencadd.com.br/blog/robotica-e-os-sistemas-autonomos-na-sustentabilidade>. Acesso em: 12 nov. 2025.

ROSENSHINE, B. Principles of instruction: research-based strategies that all teachers should know. **American Educator**, v. 36, n. 1, Washington, 2012. p. 12-19. Disponível em: <https://www.aft.org/ae/spring2012>. Acesso em: 12 nov. 2025.

SÃO PAULO (Estado). Secretaria da Educação. **Currículo Paulista**, 2019. Disponível em: [https://efape.educacao.sp.gov.br/curriculopaulista/wp-content/uploads/2023/02/Curriculo\\_Paulista-etapas-Educa%C3%A7%C3%A3o-Infantil-e-Ensino-Fundamental-ISBN.pdf](https://efape.educacao.sp.gov.br/curriculopaulista/wp-content/uploads/2023/02/Curriculo_Paulista-etapas-Educa%C3%A7%C3%A3o-Infantil-e-Ensino-Fundamental-ISBN.pdf). Acesso em: 12 nov. 2025.

TIPO TEMPORÁRIO. **O que é um sensor ultrassônico?** Disponível em: <https://tipotemporario.com.br/elektra/blog/voce-sabe-o-que-e-e-como-funciona-o-sensor-ultrassonico/>. Acesso em: 9 set. 2025.

UNIVERSAL ROBOTS. **Sensores na robótica**: quais usar em suas aplicações, 19 abr. 2023. Disponível em: <https://www.universal-robots.com/br/blog/sensores-na-robotica-quais-usar-em-suas-aplicacoes/>. Acesso em: 12 nov. 2025.

## Referências

UNIVERSAL ROBOTS. **Robôs autônomos**: o que são e seu uso na Indústria 4.0, 24 jul. 2023. Disponível em: <https://www.universal-robots.com/br/blog/robos-autonomos-o-que-sao-e-seu-uso-na-industria-40/>. Acesso em: 12 nov. 2025.

WALKER, J. **What are autonomous robots?** 8 Applications for today's AMRs. Locus Robotics, 8 jul. 2022. Disponível em: <https://locusrobotics.com/blog/what-are-autonomous-robots>. Acesso em: 12 nov. 2025.

Identidade visual: Imagens © Getty Images



# Para professores



# Para professores

**Olá, docente!** 🙌 Este material contém algumas ferramentas e recursos que visam tornar a aula mais interativa, acessível e interessante.

**Recomendamos que utilize sempre o modo apresentação do Power Point.**

Este material foi organizado para que você consiga desenvolver a aula apoiado no PDF, contudo, a experiência será mais rica e mais profunda com os recursos que o Power Point apresenta.

**Outro recurso importante é o Complemento à BNCC de Computação. Recomendamos a leitura!**


Além do Material Digital, disponibilizamos materiais com um passo a passo de **como fazer a codificação, o download da programação na placa e/ou montar o protótipo** para apoiar a condução e o planejamento da aula.


Os links para os vídeos estão disponíveis no repositório (CMSP) e no YouTube.


## Destaque




**Apoie-se em nossos recursos!** 😊

 [Tutoriais 6º Ano](#)


 [Tutoriais 7º Ano](#)


 [Tutoriais 8º Ano](#)

 [Tutoriais 9º Ano](#)

 [Tutoriais 1ª Série do Ensino Médio](#)

 [Tutoriais 2ª e 3ª Séries do Ensino Médio](#)

 [Lista de Reprodução: Kit de Robótica](#)

 [Lista de Reprodução: Orientações adicionais](#)

 [Manual: Kit de Robótica](#)

Caso não consiga acessar algum dos links acima, eles também estão listados na seguinte planilha online:

## Slide 2



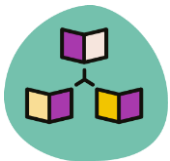
### Habilidade:

**(EF69CO03)** Descrever com precisão a solução de um problema, construindo o programa que implementa a solução descrita.

## Slides 3 e 4



**Tempo:** 10 minutos.



**Dinâmica de condução:** carros autônomos estão na mídia como exemplos de automação e uso de sensores. Pode ser que alguns estudantes já tenham visto vídeos ou reportagens sobre o tema. Logo, essa dinâmica serve para mobilizar seus conhecimentos prévios sobre o tema. De qualquer maneira, caso não tenham visto, podem imaginar seu funcionamento a partir de conhecimentos prévios de robótica. Assim, estimulamos a criatividade e o pensamento crítico sobre a automação e o uso de recursos como a inteligência artificial.

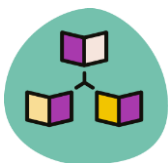
Incentive-os a compartilharem suas ideias. Caso haja disponibilidade de tempo, você pode anotar as respostas na lousa e retomá-las depois da programação e teste do protótipo da aula. Alguns temas que podem surgir e que são importantes para a robótica são: a segurança e a confiabilidade da automação.



### **Aprofundamento:**

Alguns vídeos sobre o tema são:

- NERDOLOGIA. Carros autônomos | Nerdologia Tech. **YouTube**, 30 maio 2019. Disponível em: <https://www.youtube.com/watch?v=kWf4ZFO78qE>. Acesso em: 12 nov. 2025.
- TECMUNDO. A história dos carros autônomos – História da Tecnologia. **YouTube**, 30 abr. 2025. Disponível em: <https://www.youtube.com/watch?v=lekHCjIwDg0>. Acesso em: 12 nov. 2025.



**Dinâmica de condução:** o foco é ajudar os estudantes a compreenderem o que é um sistema autônomo e como ele se diferencia de um sistema comandado. O ideal é que o professor comece ativando os conhecimentos prévios da turma, retomando a pergunta feita nas aulas anteriores: “Quem decidia o que o robô fazia até agora?”



**Expectativas de respostas:** durante a explanação do conteúdo, é esperado que os estudantes:

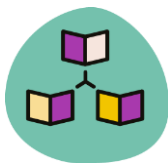
- relacionem os **carros autônomos** a tecnologias que “dirigem sozinhas” ou “sabem o caminho sem o motorista”;
- citem exemplos de **equipamentos automáticos do cotidiano**, como **portas automáticas, aspiradores robôs, elevadores, semáforos inteligentes, sensores de estacionamento, drones e alarmes**;
- percebam que esses dispositivos “**reagem**” a **estímulos do ambiente** – luz, movimento, som ou distância –, sem intervenção humana constante;
- comecem a compreender a diferença entre um **sistema comandado** (“só age quando alguém manda”) e um **sistema autônomo** (“decide sozinho com base no que detecta”).



**Aprofundamento:** para enriquecer a explicação, você pode recorrer a artigos contextualizados. Esses materiais ajudam você e os estudantes a visualizar a aplicação dos conceitos em situações reais: OPENCADD. **Entenda como a robótica e os sistemas autônomos podem ser usados para melhorar a sustentabilidade**, 19 maio 2023. Disponível em: <https://www.opencadd.com.br/blog/robotica-e-os-sistemas-autonomos-na-sustentabilidade>. Acesso em: 12 nov. 2025.

UNIVERSAL ROBOTS. **Robôs autônomos:** o que são e seu uso na Indústria 4.0, 24 jul. 2023. Disponível em: <https://www.universal-robots.com/br/blog/robos-autonomos-o-que-sao-e-seu-uso-na-industria-40/>. Acesso em: 12 nov. 2025.





**Dinâmica de condução:** comece com uma analogia simples: **pergunte à turma se eles sabem como um morcego “enxerga” no escuro**. Depois, revele que o sensor ultrassônico faz algo muito parecido – emite sons que nós não ouvimos e mede quanto tempo leva para o som “voltar”. Mostre a imagem do sensor (*slide 11*) e destaque as duas “lentes”: uma é o transmissor e a outra, o receptor.

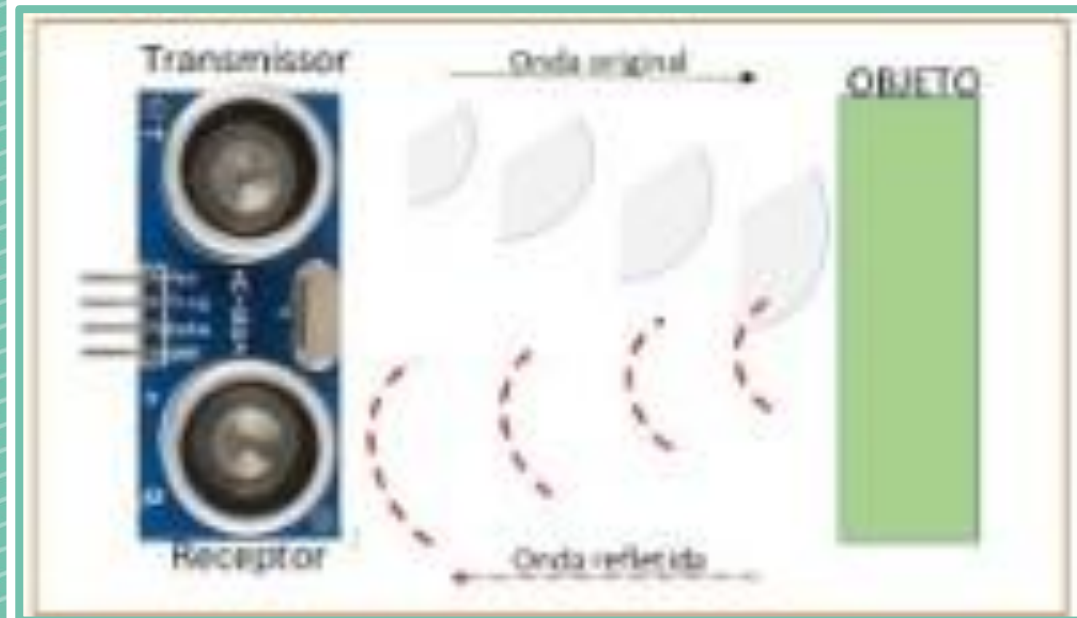


**Expectativas de respostas:** durante essa etapa, os estudantes provavelmente irão:

- reconhecer que o sensor ultrassônico mede a distância entre o robô e um objeto;
- associar o funcionamento a exemplos do dia a dia (carros, portões, torneiras, aspiradores, drones);
- compreender que o sensor não vê imagens, mas calcula tempo e distância por som.



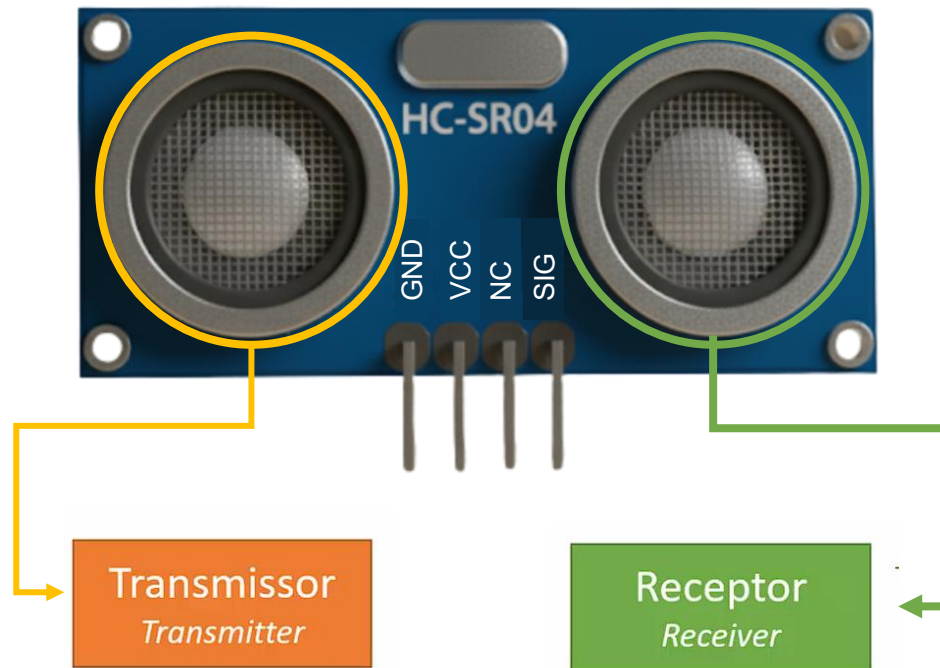
## Aprofundamento



### Especificações técnicas

- Alimentação → 5V DC
- Corrente de Operação → 15 mA
- Ângulo de efeito → 15°
- Alcance → 2cm ~ 4m
- Precisão máxima → 3 mm/cm

# Sensor ultrassônico

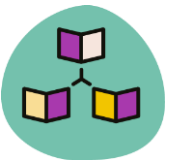


- Transmissor: envia sinais de ultrassom na direção em que está apontando. Atua como um alto-falante;
- Receptor: parte do sinal enviado é refletido pelo objeto que recebeu o sinal, sendo o receptor o responsável por detectar esses sinais de retorno e enviar para a saída do módulo. Atua com um microfone.

## Slides 12 a 14



**Tempo:** 15 minutos.



**Dinâmica de condução:** inicie com o momento “Relembre”, retomando brevemente os projetos anteriores do 8º ano que utilizaram estruturas condicionais no MakeCode.

Peça aos estudantes que citem exemplos de situações em que o código tomava decisões automáticas, dependendo de uma condição.

Mostre imagens ou trechos curtos de código para ajudar na recordação.



**Expectativas de respostas:**

“No alarme climático, um sinal sonoro e os LEDs eram acionados quando a temperatura estava alta.”

“No regador automático, o servo motor era ativado quando o solo estava seco.”

“Na casa sustentável, o sistema ligava os LEDs conforme a intensidade da luminosidade.”



**Aprofundamento:** conduza a turma a perceber que todos esses projetos usavam blocos “**se... então...**” (condicionais) para **automatizar ações**.

Mostre que compreender como e quando usar condicionais é essencial para criar sistemas inteligentes e sustentáveis.



**Conceito-base:** Condicional (se/então/senão) – estrutura lógica usada para que o programa tome decisões automáticas com base em uma condição detectada por sensores.

## Slides 13 a 14

### Conecte com o cotidiano:

inicie com o exemplo de um **semáforo**:

- “SE o sinal estiver verde, ENTÃO os carros andam.”
- “SE o sinal estiver vermelho, ENTÃO os carros param.”

Relacione com **ações do dia a dia**:

- “SE o celular apita, ENTÃO você o pega; SENÃO, continua o que estava fazendo.”

Mostre que as **condicionais** na programação funcionam da mesma forma: o sistema reage conforme uma condição.

## Slides 13 a 14

### Explore a imagem do robô:

Crie uma **história** com a cena:

“O robô está se aproximando de um cubo misterioso. Como ele sabe que há algo à frente?”

Explique que ele precisa de um **sensor ultrassônico** para “enxergar” o obstáculo.

Pergunte: “O que o robô deveria fazer se a distância fosse perigosa?”.

Destaque o uso do **SE... ENTÃO**:

“SE a distância for menor que 10 cm, ENTÃO o robô para para não bater.”

Mostre que esse tipo de raciocínio é o mesmo que usamos na **programação com condicionais**.



## Slides 15 a 32



**Recomendações:** ao longo da atividade, observe os estudantes. Perceba como eles interagem entre si, quem apresenta maior dificuldade de compreensão, seja no uso da ferramenta, seja na lógica básica de programação. Converse com os estudantes que apresentarem maior facilidade na atividade para que eles possam atuar como multiplicadores e auxiliares nas tarefas.

Boa aula!

## Slide 32

**Desafio bônus:** se houver tempo disponível ao final da aula, proponha que eles modifiquem a programação, enfeitem ou aprimorem seu protótipo.

**Dica extra:** apresente o desafio como uma missão especial ou um nível extra para aumentar o engajamento. Dessa forma, os estudantes que terminarem antes se sentirão motivados a continuar explorando sem pressão.



**FICA A DICA**

**Flexibilidade e adaptação:** esteja aberto a modificar as programações e o protótipo conforme necessário, com base no *feedback* dos estudantes e nas suas observações durante a aula.

# Para professores

## Tarefas de Robótica:

**Caro professor**, seguem instruções para postagem da **atividade de aula** para seus estudantes (se houver). Caso tenha dúvidas, disponibilizaremos um vídeo tutorial na *playlist* de orientações adicionais. Orientamos que a postagem seja feita **antes ou durante a aula** para que o estudante possa **registrar** a entrega da atividade **durante a aula**.

O objetivo desse envio é que o estudante **registre**, na Sala do Futuro, a atividade realizada em sala de aula, para acompanharmos o **engajamento** com as aulas de Robótica e possibilitar a você, docente, avaliar a **aprendizagem e a evolução do estudante**.

Orientamos também que a atividade seja postada sem prazo de término especificado. Assim, caso esteja com dificuldades em acessar a Sala do Futuro ou a internet no dia, o estudante poderá finalizar a tarefa posteriormente.

**Destaque**



**Importante: nem todas as aulas do bimestre possuem tarefas!** Para saber para quais aulas estão previstas tarefas, consulte o **escopo-sequência** do componente.

# Para professores

## Tarefas de Robótica:

**Localizador:** **efrob09** (Ensino Fundamental, Robótica, 9º ano)

1. Acesse o link: <http://tarefas.cmsp.educacao.sp.gov.br>.
2. Clique em “**Atividades**” e, em seguida, em “**Modelos**”.
3. Na sequência, clique em “**Buscar por**”, selecione a opção “**Localizador**”.
4. Copie o localizador acima e cole-o no campo de busca.
5. Clique em “**Procurar**”. Uma lista de tarefas do componente aparecerá. Elas estarão organizadas pelo título da aula.
6. Selecione a tarefa que **corresponde à aula do dia** (busque pelo título da aula) para envio à turma, clicando na seta verde que aparece na frente da atividade.
7. Defina qual ou quais turmas receberão a atividade. Selecione a data de envio, mantenha sem prazo de resposta e clique em “**Publicar**”
8. Informe à turma a data de agendamento e, caso deseje, combine o prazo da atividade.

**Pronto! A atividade foi enviada com sucesso!**



