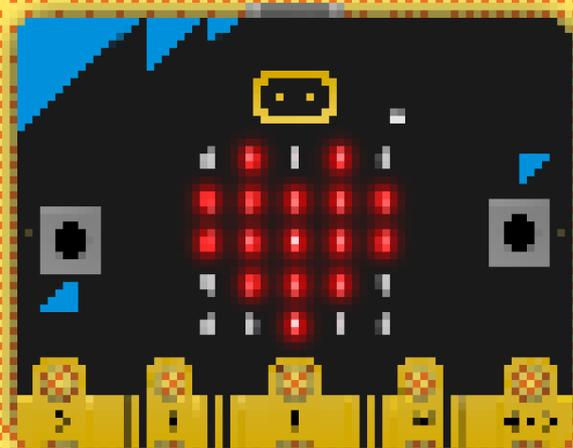


# Aprendendo com MICRO:BIT BBC



## Coordenadores:

Breno Ferraz de Oliveira  
Eduardo de Amorim Neves

## Autores:

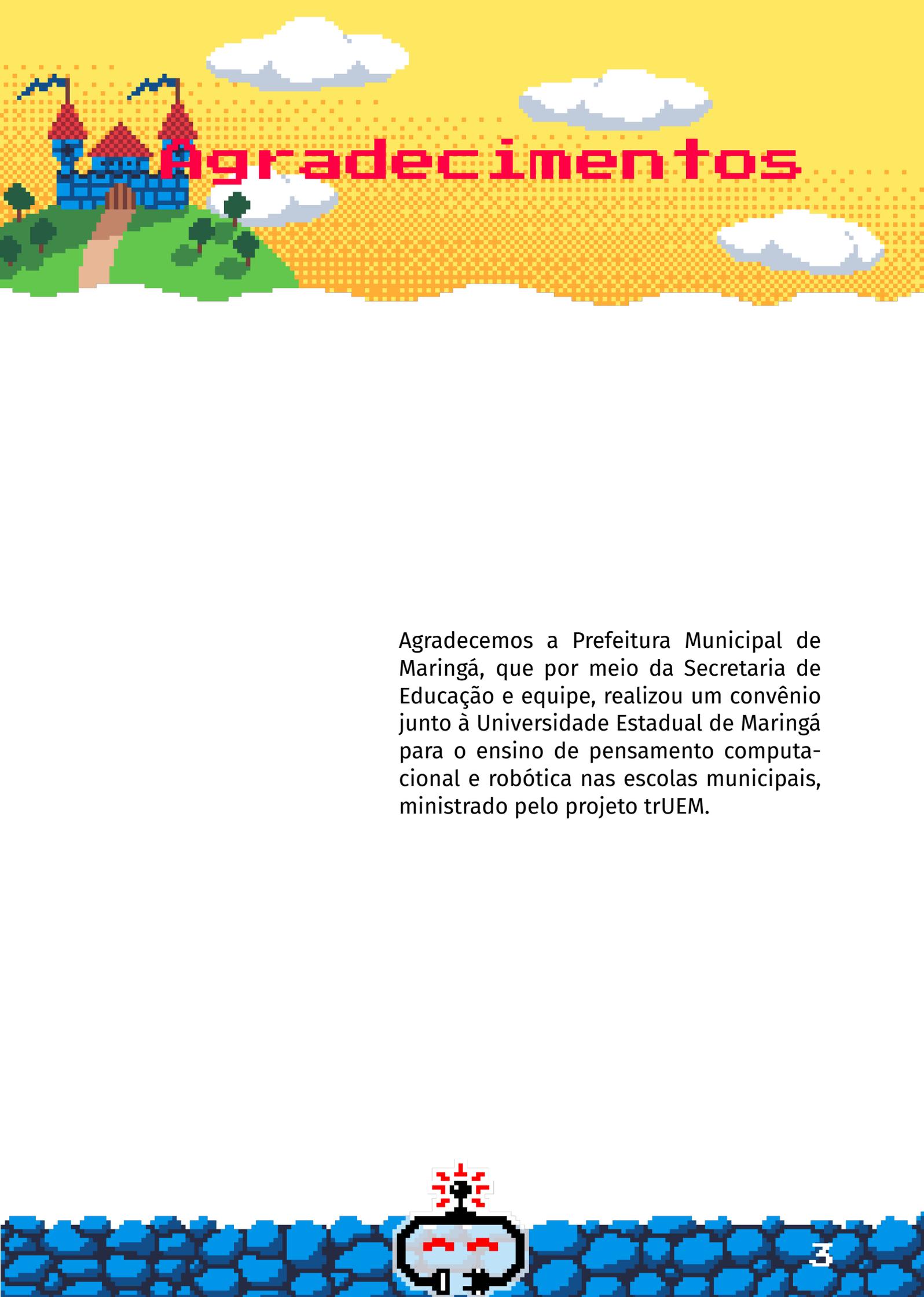
Andrey Karvat  
Arão Victor Rocha Dos Santos  
João Pedro Podolan  
José Leonardo Oler  
Luciane Guarnieri Brodbeck  
Rafaela Lavagnoli  
Thainá Gomes P. Zampieri

trUEM

2023 - v1







# Agradecimentos

Agradecemos a Prefeitura Municipal de Maringá, que por meio da Secretaria de Educação e equipe, realizou um convênio junto à Universidade Estadual de Maringá para o ensino de pensamento computacional e robótica nas escolas municipais, ministrado pelo projeto trUEM.





# Sumário

<b>0</b>	<b>Conhecendo o micro:bit</b>	<b>9</b>
0.1	O que é o micro:bit . . . . .	9
0.2	Na plataforma micro:bit . . . . .	12
<b>1</b>	<b>Letra inicial</b>	<b>19</b>
1.1	Novos blocos: . . . . .	19
1.1.1	mostrar leds . . . . .	19
<b>2</b>	<b>Coração pulsante</b>	<b>23</b>
<b>3</b>	<b>Coração lento</b>	<b>27</b>
3.1	Novo blocos: . . . . .	27
3.1.1	pausa . . . . .	27
<b>4</b>	<b>Crachá</b>	<b>35</b>
4.1	Novos blocos: . . . . .	35
4.1.1	mostrar string "Hello!" . . . . .	35
<b>5</b>	<b>no iniciar</b>	<b>43</b>
5.1	Novos blocos: . . . . .	43
5.1.1	no iniciar . . . . .	43
<b>6</b>	<b>Cumprimento do micro:bit</b>	<b>47</b>
<b>7</b>	<b>Fogos de artifício</b>	<b>53</b>
<b>8</b>	<b>Usando os botões</b>	<b>63</b>
8.1	Input . . . . .	63
8.1.1	no botão A pressionado . . . . .	63
<b>9</b>	<b>Quadro mágico</b>	<b>69</b>
9.1	Limpendo a tela . . . . .	69
9.1.1	limpar tela . . . . .	69
<b>10</b>	<b>Os sons das emoções</b>	<b>77</b>
10.1	play sound risadinha until done . . . . .	77
<b>11</b>	<b>Dado Digital</b>	<b>89</b>
11.1	Montando um dado . . . . .	89
11.1.1	mostrar número . . . . .	89



# SUMARIO

11.1.2	escolher aleatório 0 até 10 . . . . .	89
<b>12</b>	<b>Montando um termômetro</b>	<b>97</b>
12.1	temperatura . . . . .	97
<b>13</b>	<b>Conhecendo a lógica</b>	<b>103</b>
13.1	Novos Blocos . . . . .	104
13.1.1	escolher de forma aleatória verdadeiro ou falso . . . . .	104
13.1.2	se verdadeiro então . . . . .	104
13.1.3	se verdadeiro então, senão . . . . .	104
<b>14</b>	<b>Pedra, papel e tesoura</b>	<b>111</b>
14.1	em agitar: . . . . .	111
14.2	Variáveis . . . . .	112
<b>15</b>	<b>Bola mágica</b>	<b>133</b>
<b>16</b>	<b>Batata Quente</b>	<b>149</b>
<b>17</b>	<b>Fantasma</b>	<b>163</b>
<b>18</b>	<b>Extensões</b>	<b>173</b>
18.1	Maqueen . . . . .	173
18.2	Régua . . . . .	173
<b>19</b>	<b>Carro controlado por luz</b>	<b>179</b>
<b>20</b>	<b>Carro que desvia</b>	<b>191</b>
<b>21</b>	<b>Carro segue faixa</b>	<b>223</b>
<b>22</b>	<b>Carrinho de controle remoto</b>	<b>263</b>
<b>23</b>	<b>Semáforo</b>	<b>305</b>
<b>24</b>	<b>Carro colorido</b>	<b>319</b>
<b>25</b>	<b>Montando uma calculadora</b>	<b>349</b>



Caro leitor,

Este livro foi escrito com o intuito de auxiliar na aprendizagem do **pensamento computacional**, que é uma abordagem usada para solução de problemas que utiliza conceitos básicos da computação, e da **robótica**, a ciência e técnica da concepção, construção e utilização de robôs. As atividades desenvolvidas neste livro, sob as perspectivas do pensamento computacional, têm como finalidade contribuir na construção do pensamento lógico, na habilidade de reconhecimento de padrões e no desenvolvimento do raciocínio. Os quatro pilares do pensamento computacional são a decomposição, o reconhecimento de padrões, a abstração de um problema e os algoritmos.

Neste livro você encontrará atividades na forma de jogos e brincadeiras que foram desenvolvidos para serem o ponto de partida para a compreensão e a criação de projetos maiores, que facilitam nossas rotinas e otimizam nosso dia-a-dia.

Cada capítulo deste livro possui um QR Code e hiperlink pelo(s) qual(is) é possível acessar atividades completas, para que você possa testá-las e alterá-las, e ainda criar outras com base nas atividades desenvolvidas neste material. Para acessar uma atividade pelo QR Code, basta apontar a câmera de seu celular para a imagem que aparece no canto superior direito de cada capítulo. O código mostrará um hiperlink e, ao clicar nele, a atividade será aberta online.

Este livro foi elaborado pelo **trUEM**, que é um projeto de extensão, desenvolvido na Universidade Estadual de Maringá (UEM), desde setembro de 2019, fruto de uma parceria de Guilherme Rocha, Denise Adorno Lopes e Breno Ferraz de Oliveira, sendo este o atual coordenador do projeto. O nome trUEM, faz referência ao TRUE da linguagem computacional e a UEM. O projeto visa atender pessoas de todas as idades, em especial crianças, que têm interesse em robótica, tendo elas conhecimento prévio sobre pensamento computacional, ou não. O livro conta com muitas ilustrações, para que o aprendizado seja simples, acessível, tangível e inclusivo.

Esperamos que seja uma experiência proveitosa e capaz de, não só instigar o conhecimento pelo pensamento computacional e robótica, mas também o aperfeiçoamento de ambos.

Com carinho,

Equipe trUEM.

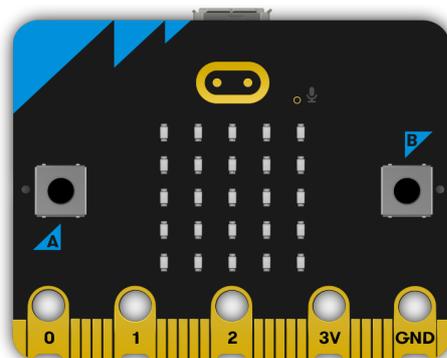




# Capítulo 0: Conhecendo o micro:bit

## O que é o micro:bit

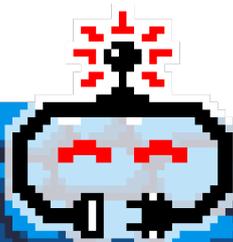
O micro:bit é um microprocessador que atua como o cérebro do computador. Foi desenvolvido pela BBC (British Broadcasting Corporation) em 2015, em parceria com diversas organizações. Ele foi criado com intuito principal de auxiliar no ensino de programação e tecnologia para jovens e crianças, ajudando, também, no desenvolvimento do raciocínio lógico, criatividade e resolução de problemas.



Ele possui diversos recursos, como ilustrado na Figura 1. Dentre eles, um arranjo de 25 LEDs (componentes que convertem eletricidade em luz) que pode ser utilizado para mostrar letras, números e figuras. Os LEDs do micro:bit podem ter sua função invertida e funcionarem como sensores de luz, permitindo detectar o nível de luminosidade de um ambiente. No canto superior direito aos LEDs há um microfone e os sons captados por ele podem ser usados em comandos para acender ou apagar os LEDs.

No micro:bit existem dois botões, A e B, que podem ser programados para acionarem um trecho do código elaborado. O logotipo também funciona como um botão de acionamento, pois é sensível ao toque.

O micro:bit possui também 25 pinos, posicionados na parte inferior da placa, que permitem a conexão de dispositivos externos como motores, sensores extras, atuadores e outros componentes elétricos.



# Conhecendo o micro:bit

Há também no micro:bit uma antena de rádio que permite a comunicação através de dispositivos sem fio. A função rádio pode ser utilizada para enviar mensagens para outros micro:bits, criar jogos, controle remoto e muito mais. O micro:bit ainda possui uma antena BLE (*Bluetooth Low Energy*) que permite a comunicação sem fio com computadores, smartphones e tablets que possuam esse protocolo de comunicação.

No micro:bit há ainda um acelerômetro, que detecta quando a placa é movida, e uma bússola, com a função de identificar o campo magnético da Terra.

O micro:bit possui também um alto-falante que emite sons pré-programados, como efeitos sonoros e melodias, e sons que podem ser programados utilizando as notas musicais.

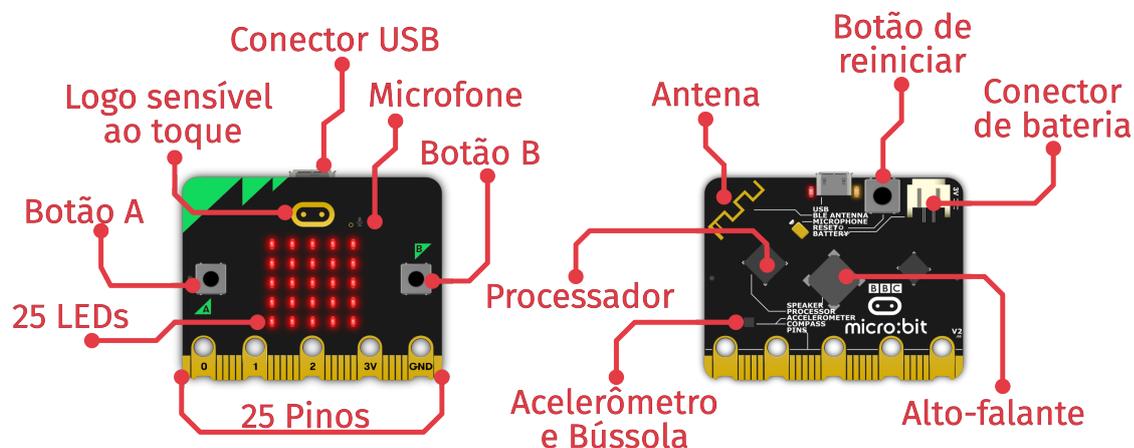


Figura 1: Localização dos componentes presentes na placa.

# Conhecendo o micro:bit

Com o micro:bit é possível criar inúmeros projetos de programação, dos mais simples até os mais complexos, usando algumas linguagens de programação. Essas linguagens são a forma de tradução dos comandos que queremos que o processador realize para uma linguagem que o computador consiga compreender e, dessa maneira, executar o que foi pedido. São elas: Python e JavaScript.

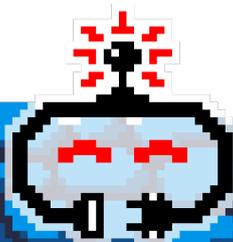
Para trabalharmos com o micro:bit podemos utilizar, também, a linguagem de Blocos, uma interface didática, com uma programação atrativa e intuitiva, facilitando assim, a realização de atividades, jogos e suas outras aplicações.

Ao usarmos uma linguagem de programação, precisamos ter bem claro que a programação é o comando passo a passo das atividades que pedimos para o micro:bit fazer, serve como um guia das tarefas que o processador precisa executar. Realizaremos estes comandos através da linguagem de Blocos usada na plataforma de interface de desenvolvimento, o MakeCode, que lida com a programação do micro:bit.

Os programas são organizados em forma de algoritmos. Os algoritmos são os passos que devem ser seguidos pelo micro:bit em cada programação, como se fosse uma receita a ser seguida.

## MICRO:DICA

Para resumir as informações contidas nesta introdução podemos pensar no micro:bit como um mini computador que faz o que pedimos quando falamos a língua dele.



# Conhecendo o micro:bit

## Na plataforma micro:bit

1. Para iniciar a parte da programação do micro:bit, devemos entrar na plataforma MakeCode. Para isso, podemos pesquisar em um navegador da internet (de preferência o Google Chrome) pelo site: <https://makecode.microbit.org>
2. Se a página não estiver em português, clique na engrenagem, localizada no canto superior direito da página, e vá em *Language* como na Figura 2;

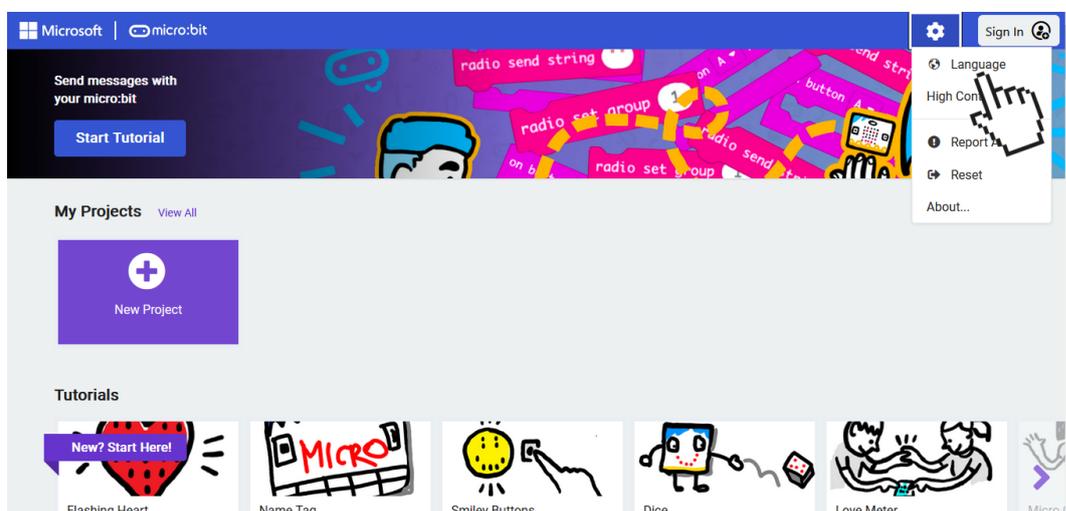


Figura 2: Página inicial da plataforma MakeCode.

# Conhecendo o micro:bit

3. Mude o idioma para Português (Brasil), como apresentado na Figura 3;

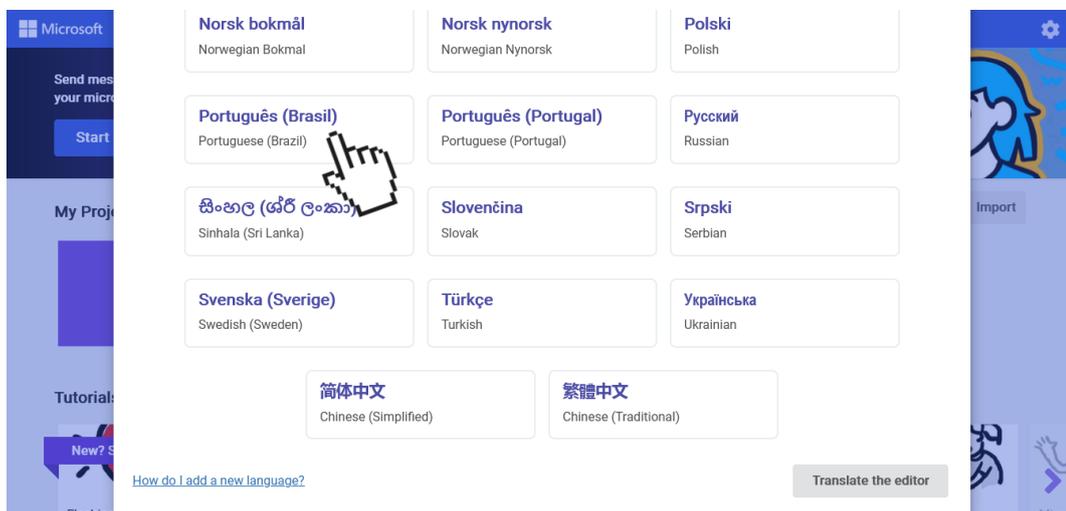


Figura 3: Configuração de idioma.

4. Após este procedimento, a página estará traduzida para o português e pronta para a criação de novos projetos. Então, é só começar. Selecione o ícone **Novo projeto**. Uma janela aparecerá na tela, como representado na Figura 4. Escolha um nome para a atividade. Depois, clique em **Criar** e comece a programar.

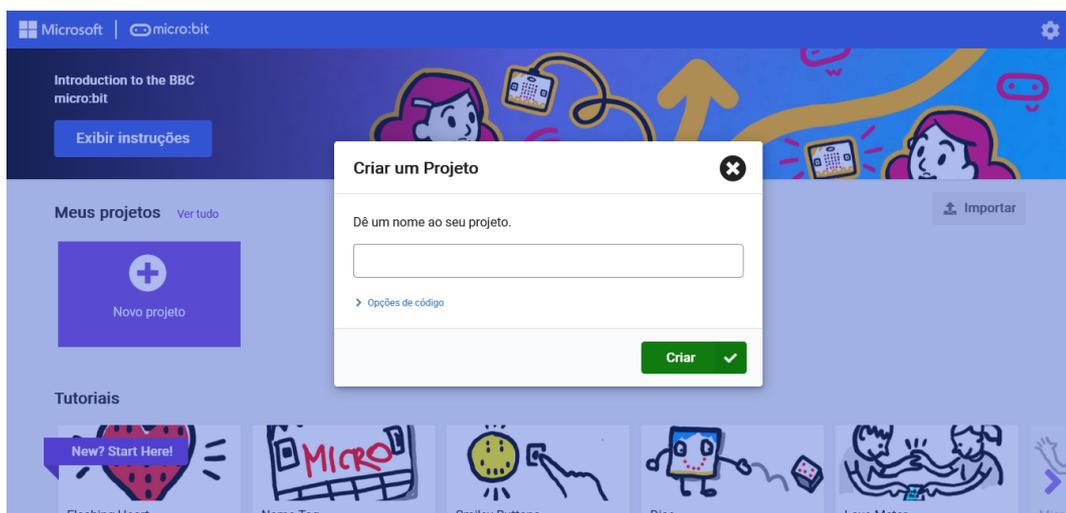
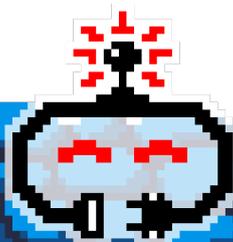


Figura 4: Criando um projeto.



# Conhecendo o micro:bit

5. Na nova interface, representada pela Figura 5, podemos observar três partes: a **simulação da placa física (1)** à esquerda, a **biblioteca de funções (2)** no centro e o **espaço de edição (3)** para a construção do programa à direita.

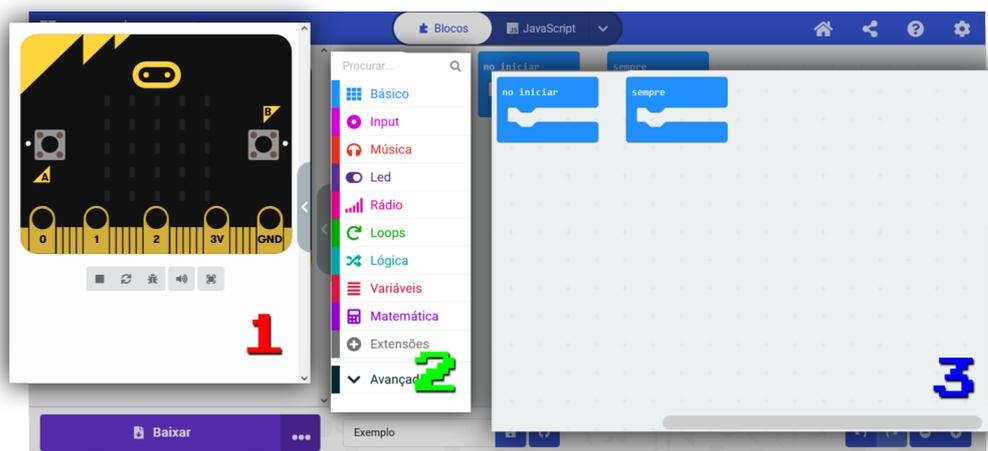


Figura 5: Tela inicial do projeto.

A **simulação da placa física** é a visualização do programa escrito, isto é, o que acontecerá no micro:bit quando o código for baixado. Assim, é possível verificar no simulador se o código foi escrito corretamente, antes de baixá-lo.

A **biblioteca de funções** é uma coleção de subprogramas utilizados no desenvolvimento de softwares, ou seja, quando se está usando uma linguagem de programação e uma biblioteca é aplicada, um conjunto de funções que já foram escritas por outros desenvolvedores será acessado. No micro:bit, são os blocos prontos que usaremos para montar o código.

O **espaço de edição** é onde o programa será escrito. Neste espaço, serão utilizadas as funções da biblioteca, na forma de blocos, para montar os comandos que serão executados pelo micro:bit.

**Atenção:** Certifique-se que na parte superior central a opção **Blocos** esteja selecionada, para que a linguagem de programação se mantenha.

# Conhecendo o micro:bit

**Aprenderemos agora como baixar os códigos produzidos para o micro:bit**

- 1.** Primeiramente, conecte o micro:bit na entrada USB do seu computador, como representado na Figura 6;

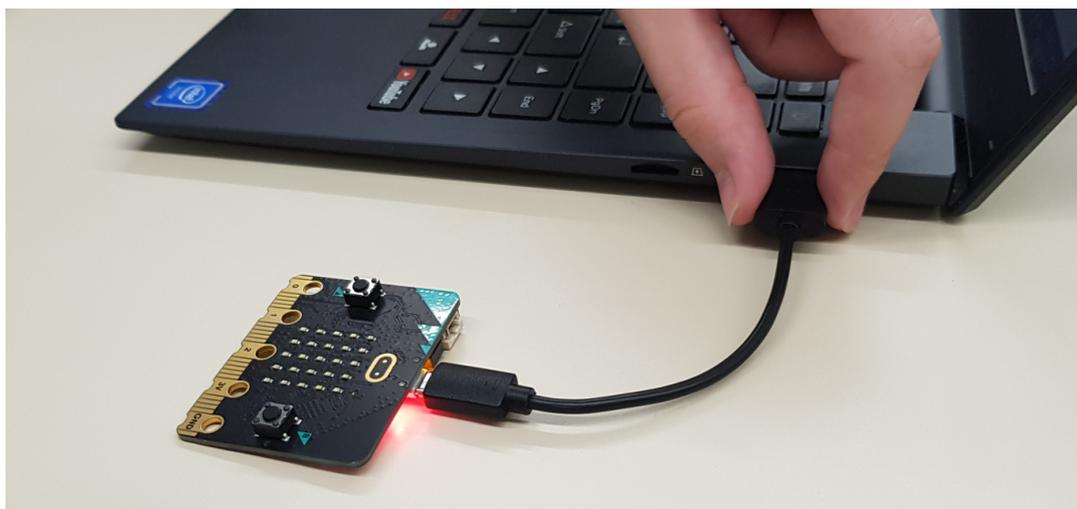


Figura 6: Conectando o micro:bit ao computador.

- 2.** Com o micro:bit conectado, selecione os três pontinhos localizados no canto inferior esquerdo da plataforma, como na Figura 7;

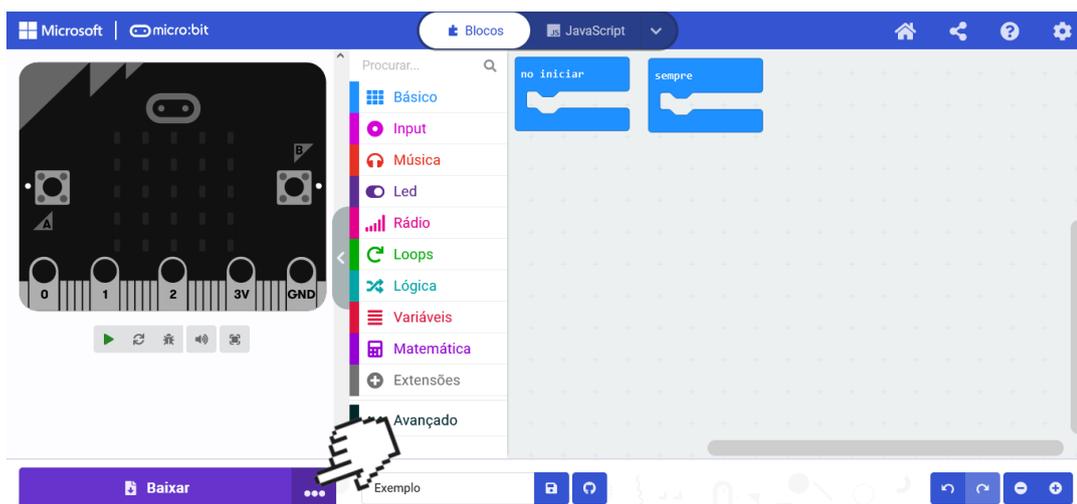
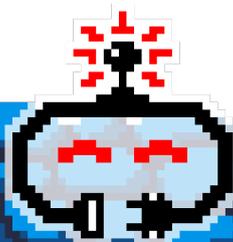


Figura 7: Conectando um dispositivo.



# Conhecendo o micro:bit

3. Em seguida, selecione a opção **Connect device**, como mostrado na Figura 8;

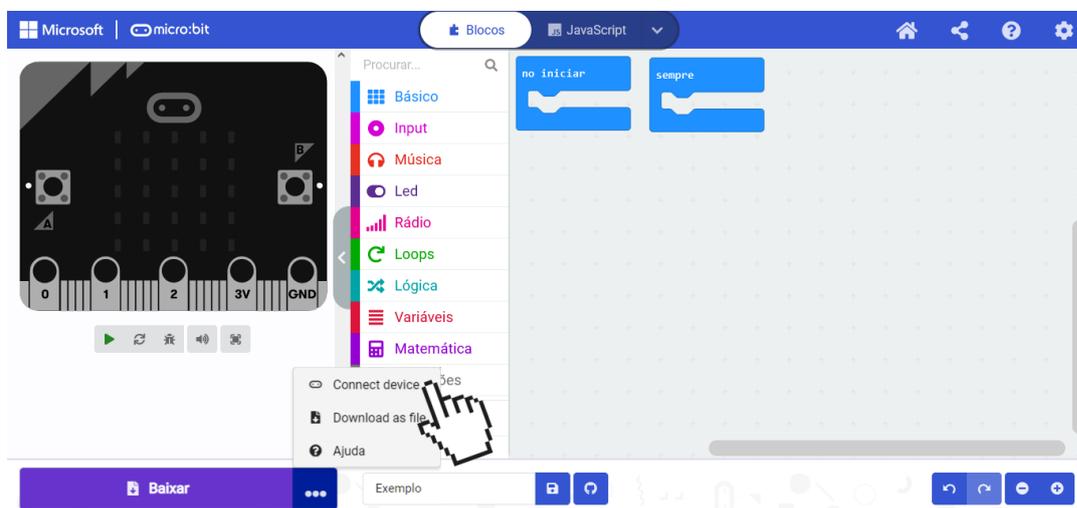


Figura 8: Conectando o micro:bit.

4. Selecione **Próximo** nas duas janelas seguintes que aparecerem, como nas Figuras 9 e 10;

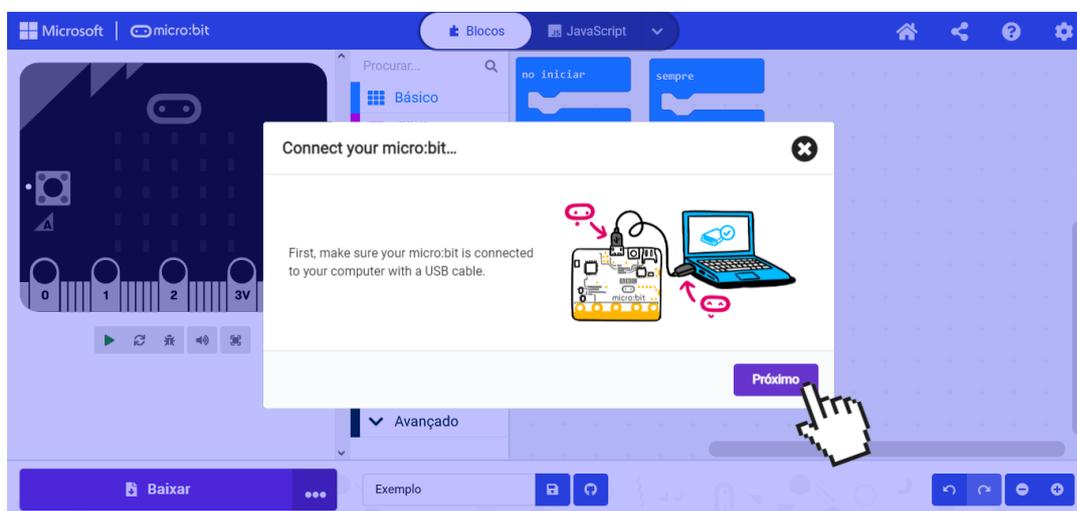


Figura 9: Conectando o micro:bit.

# Conhecendo o micro:bit

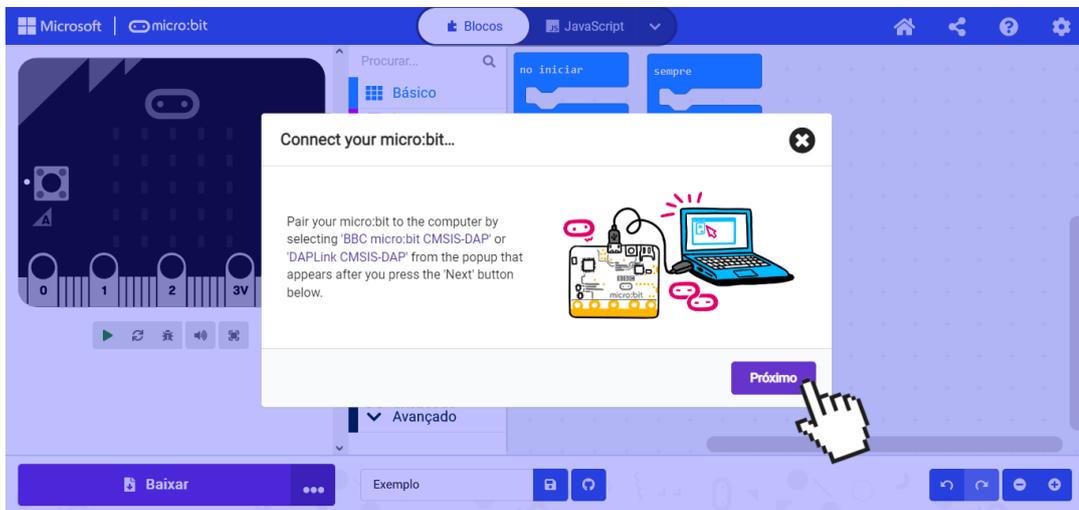


Figura 10: Conectando o micro:bit.

5. Selecione a placa do micro:bit e clique em conectar, como na Figura 11;

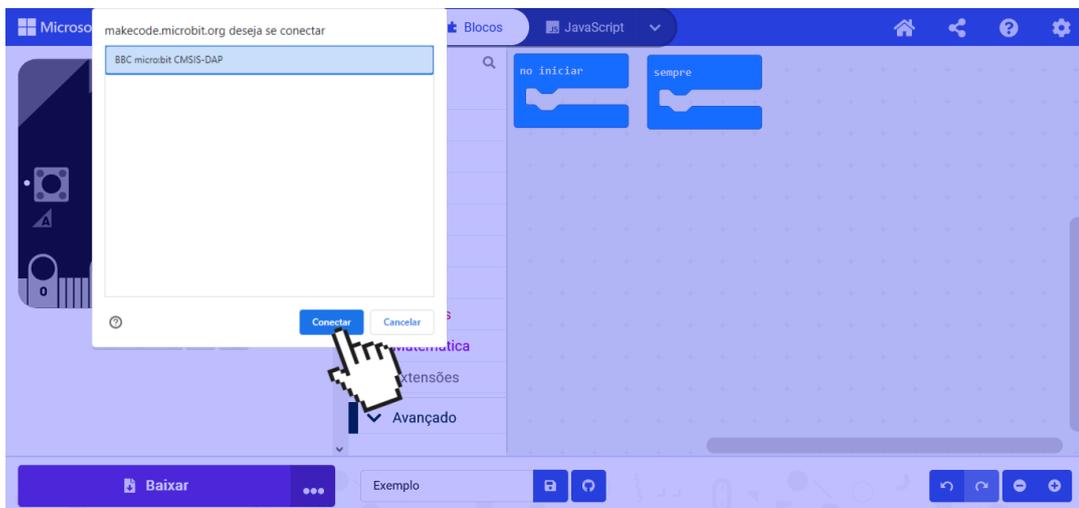


Figura 11: Conectando o micro:bit.



# Conhecendo o micro:bit

6. Selecione **Feito**, como na Figura 12. Agora o micro:bit está conectado ao computador;

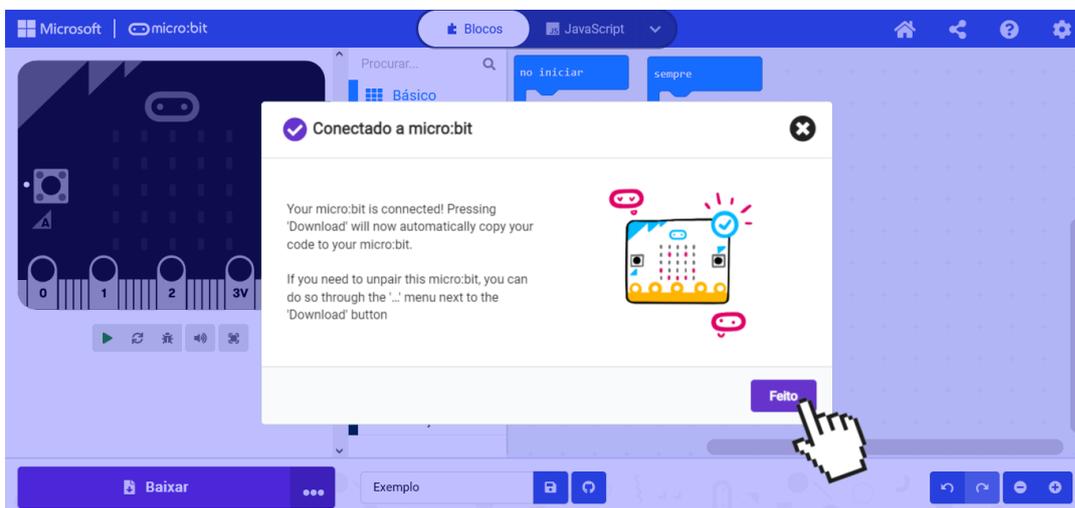


Figura 12: Conectado ao micro:bit.

7. Enquanto o micro:bit estiver conectado ao computador via cabo USB, sempre que você montar uma programação, basta ir no botão **Baixar**, como na Figura 13, que o programa será direcionado diretamente para a placa conectada.

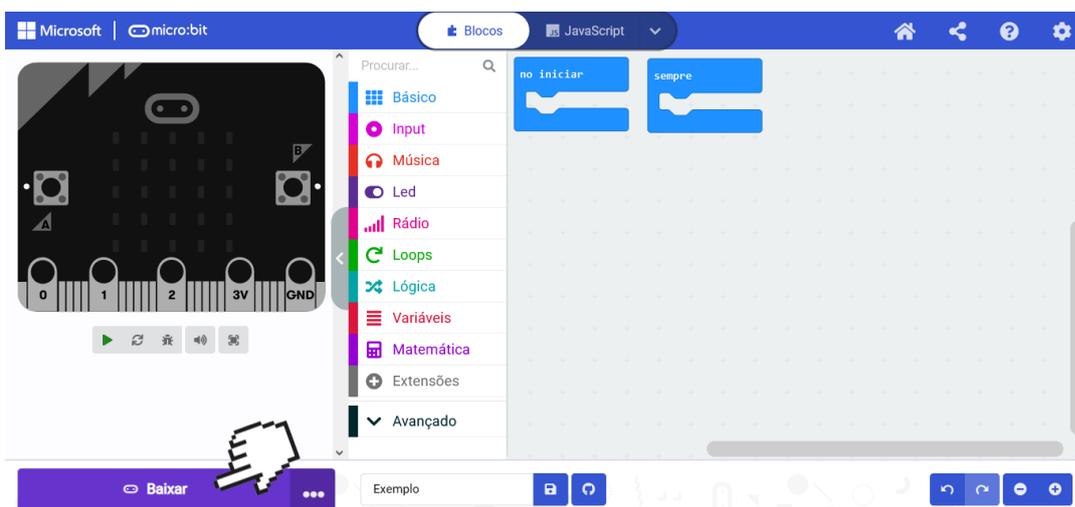


Figura 13: Baixando um programa para o micro:bit.



# Capítulo 1: Letra inicial

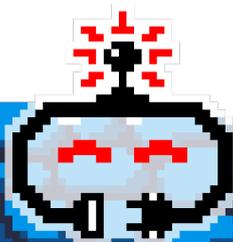
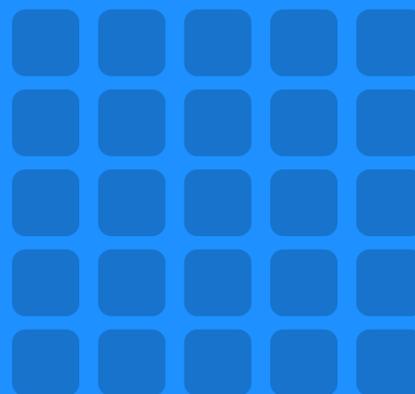
## Novos blocos:

Para o segundo código, continuaremos trabalhando na aba **Básico**, mas agora vamos utilizar o comando **mostrar leds** e faremos com que o micro:bit mostre a letra inicial do nosso nome.

### mostrar leds

Quando este bloco é usado, podemos acender apenas os LEDs selecionados. Assim, é possível fazer nosso próprio desenho. Para selecionar quais serão acesos, basta clicar nos quadrados.

mostrar leds



# Letra inicial

## Passo a passo:

1. Selecione o comando **mostrar leds** da aba **Básico** e leve-o para o bloco de acionamento **sempre**, Figuras 1.1 e 1.2;

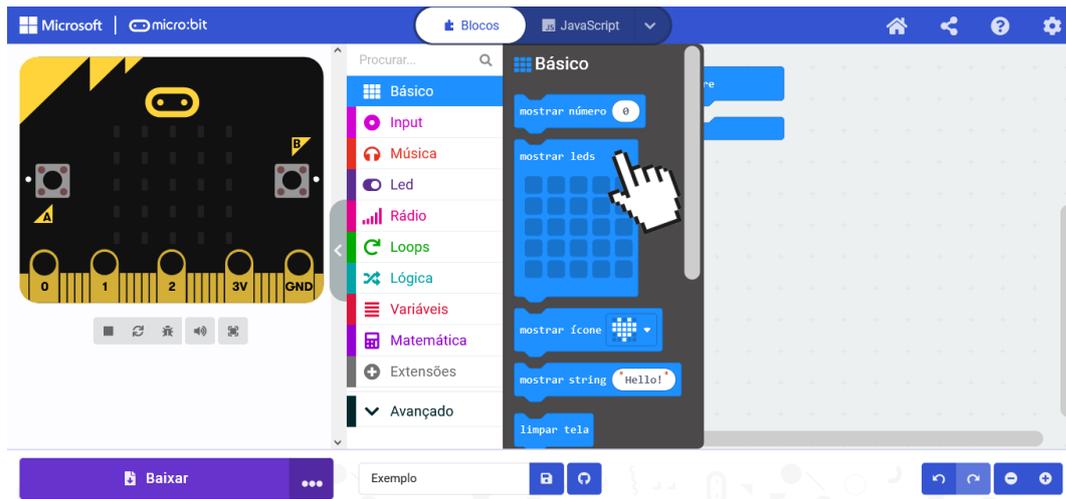


Figura 1.1: Selecionando o bloco mostrar leds.

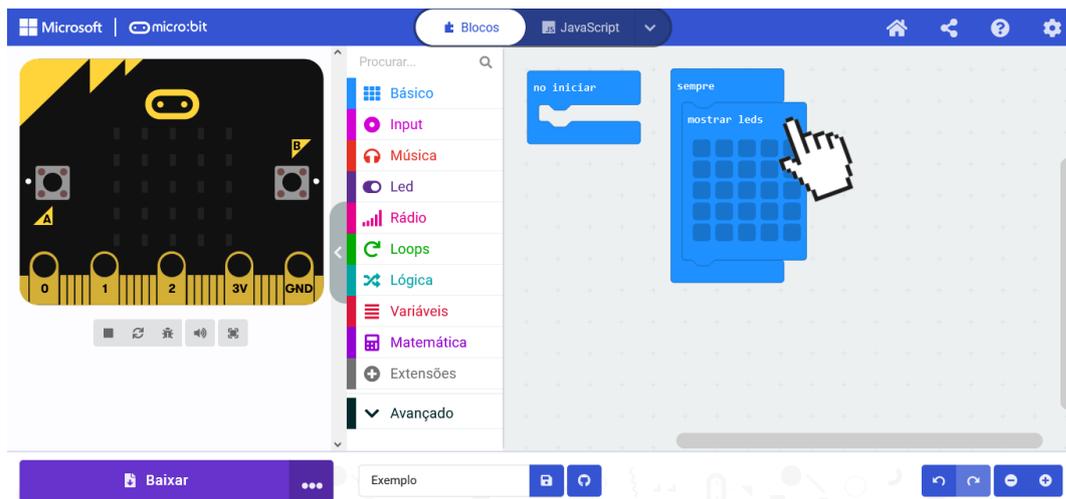


Figura 1.2: Montando o bloco mostrar leds.

# Letra inicial

2. Em **mostrar leds** desenhe a letra inicial do seu nome, como no exemplo da Figura 1.3;



Figura 1.3: Desenhando uma letra.

**Agora, é só baixar o programa para o micro:bit.**

[Clique aqui para acessar o código.](#)

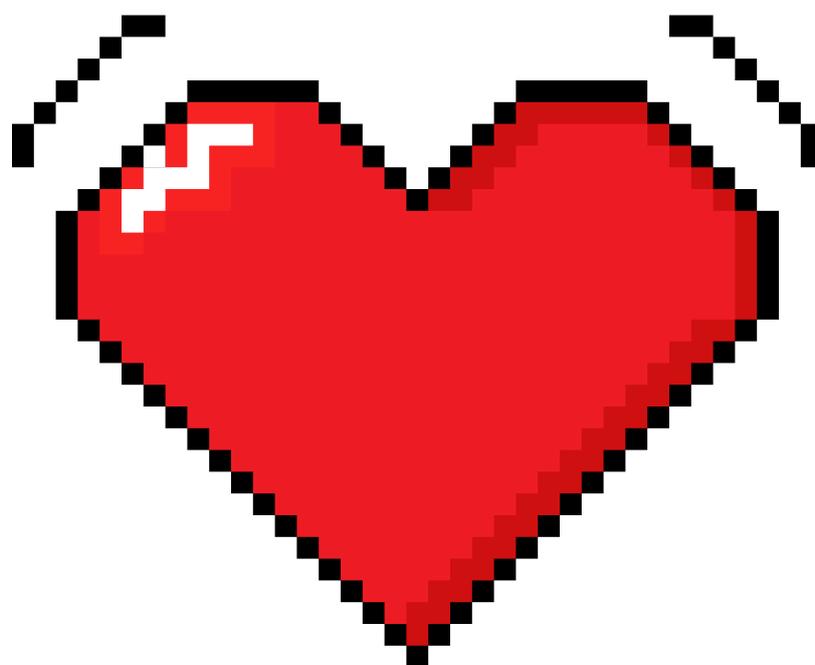




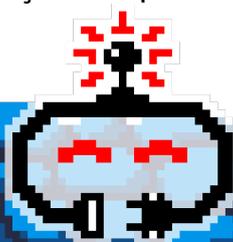
## Letra inicial



# Capítulo 2: Coração pulsante



A programação agora fará o coração pulsar nos LEDs do micro:bit. Para isso, use os comandos já conhecidos e faça com que a figura ganhe movimento.



# Coração pulsante

## Passo a passo:

1. Selecione na aba **Básico** um **mostrar ícone** e leve-o até o bloco **sempre**, como mostram as Figuras 2.1 e 2.2;



Figura 2.1: Selecionando o bloco mostrar ícone.

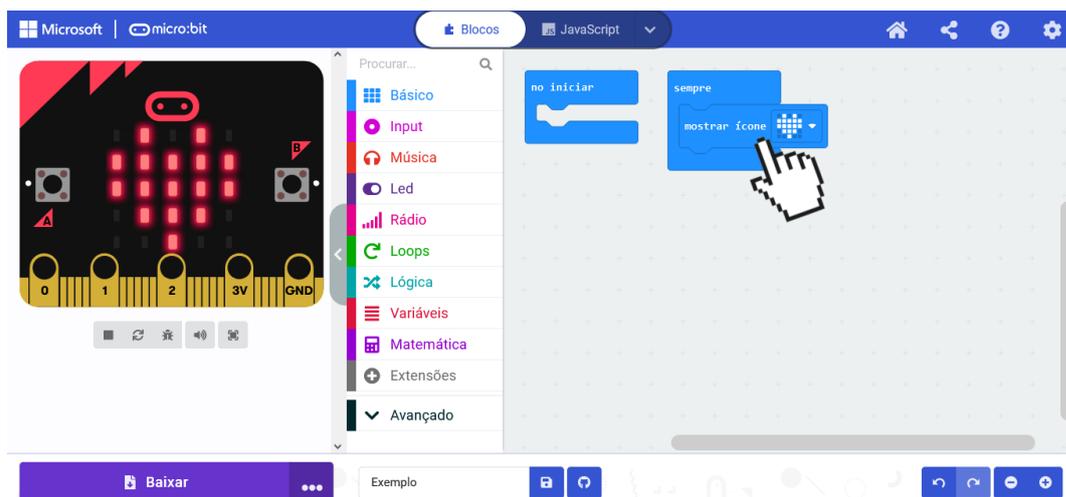


Figura 2.2: Montando um comando.

# Coração pulsante

2. Pegue mais um comando **mostrar ícone** e coloque-o logo abaixo do outro, como mostram as Figuras 2.3 e 2.4;

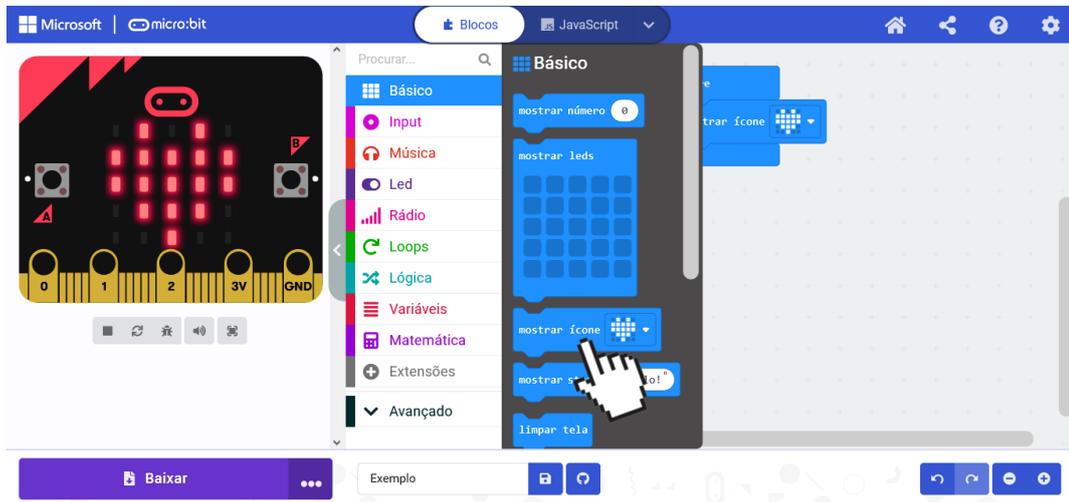


Figura 2.3: Selecionando o bloco mostrar ícone.

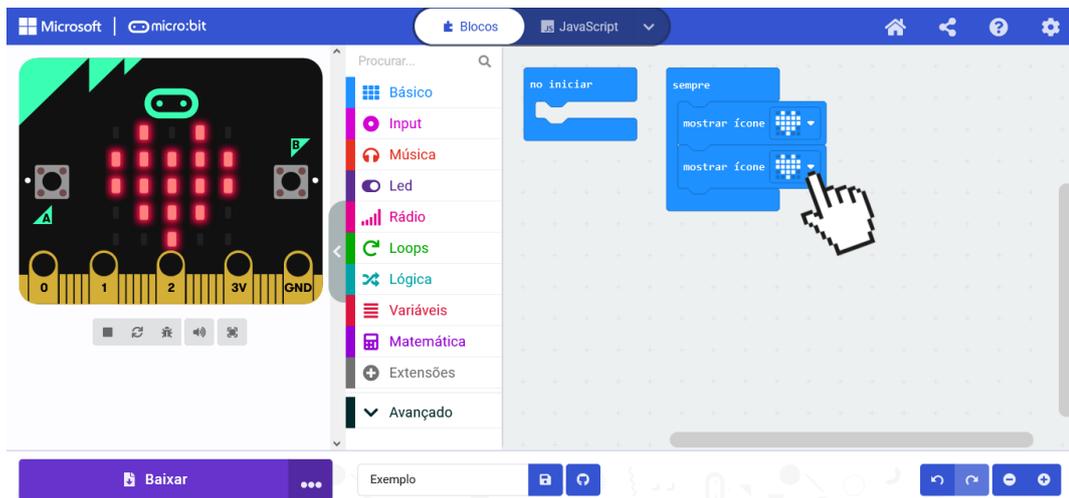


Figura 2.4: Montando comandos.



# Coração pulsante

3. No segundo **mostrar ícone**, escolha o ícone **coração pequeno**, como nas Figuras 2.5 e 2.6;

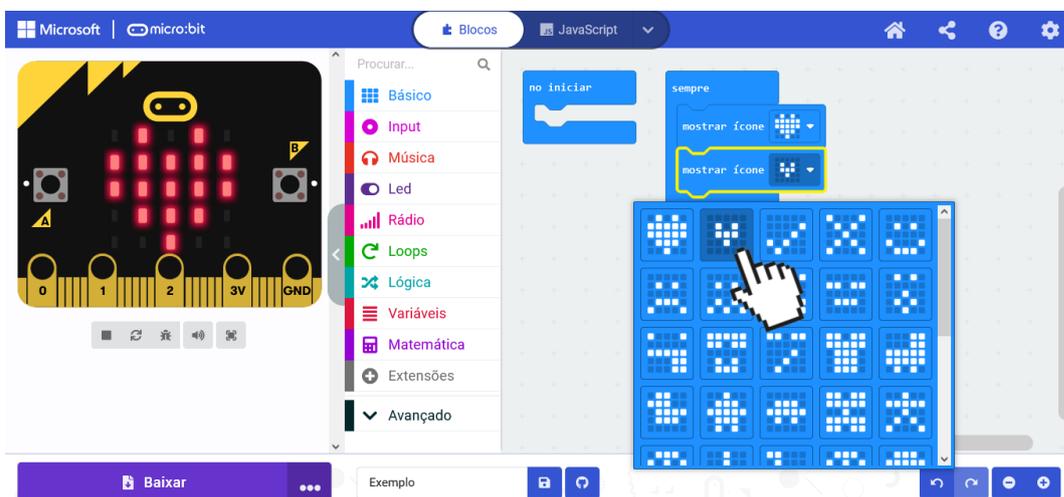


Figura 2.5: Selecionando um ícone.



Figura 2.6: Ícones do coração pulsante.

**Baixe o programa para o micro:bit e veja o coração pulsar.**

[Clique aqui para acessar o código.](#)



# Capítulo 3: Coração lento



## Novo blocos:

O coração pulsou no micro:bit, mas será que é possível fazê-lo pulsar em uma velocidade diferente? Existe um comando no micro:bit, da aba **Básico** que torna isso possível, que é a **pausa**.



### pausa

Este bloco faz com que exista um tempo entre os comandos. É o intervalo entre as atividades realizadas pelo micro:bit.



# Coração lento

## Passo a passo:

1. Seleccione o **mostrar ícone** e leve-o para o bloco de acionamento **sempre**, como na Figura 3.1 e na Figura 3.2;



Figura 3.1: Selecionando o bloco mostrar ícone.

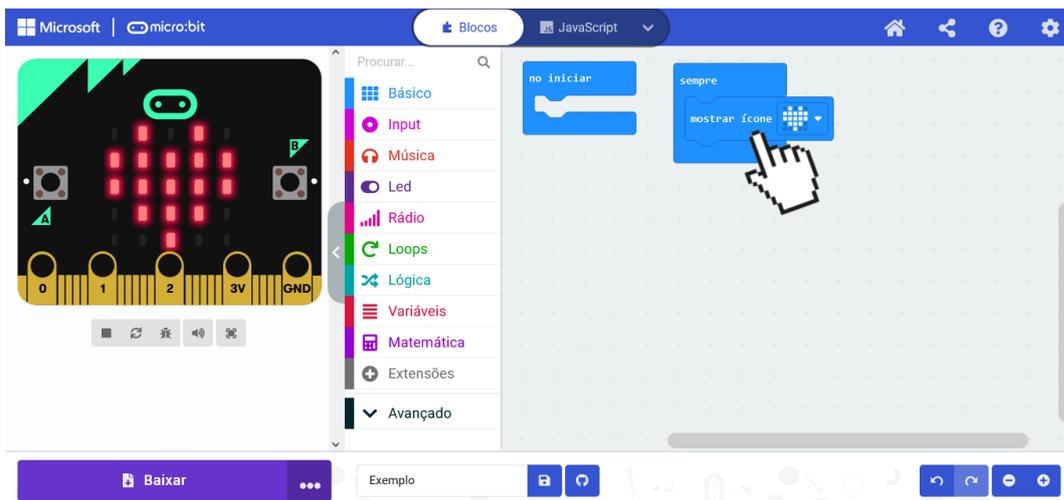


Figura 3.2: Montando comandos.

# Coração lento

2. Em seguida selecione uma **pausa** e coloque-a logo abaixo, como na Figura 3.3;

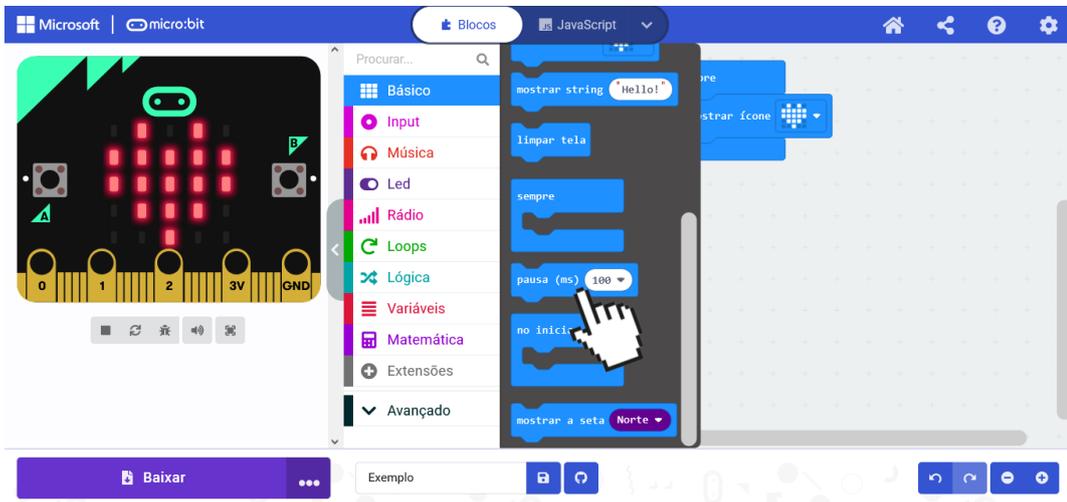


Figura 3.3: Selecionando uma pausa.

3. Na sequência, selecione outro **mostrar ícone** e mais uma **pausa**, como nas Figuras 3.4 e 3.5;

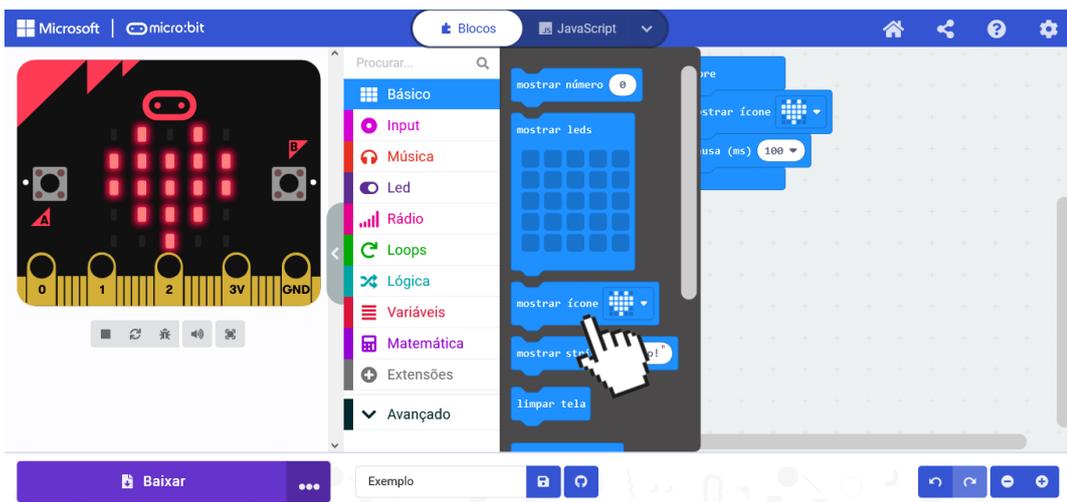


Figura 3.4: Selecionando o bloco mostrar ícone.



# Coração lento

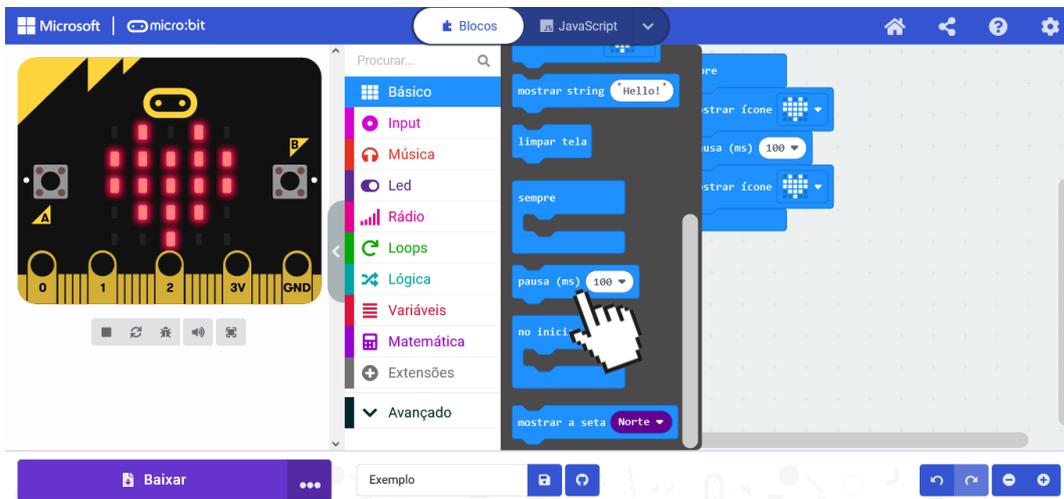


Figura 3.5: Selecionando mais uma pausa.

4. No segundo **mostrar ícone** coloque o **coração pequeno**, como nas Figuras 3.6 e 3.7;

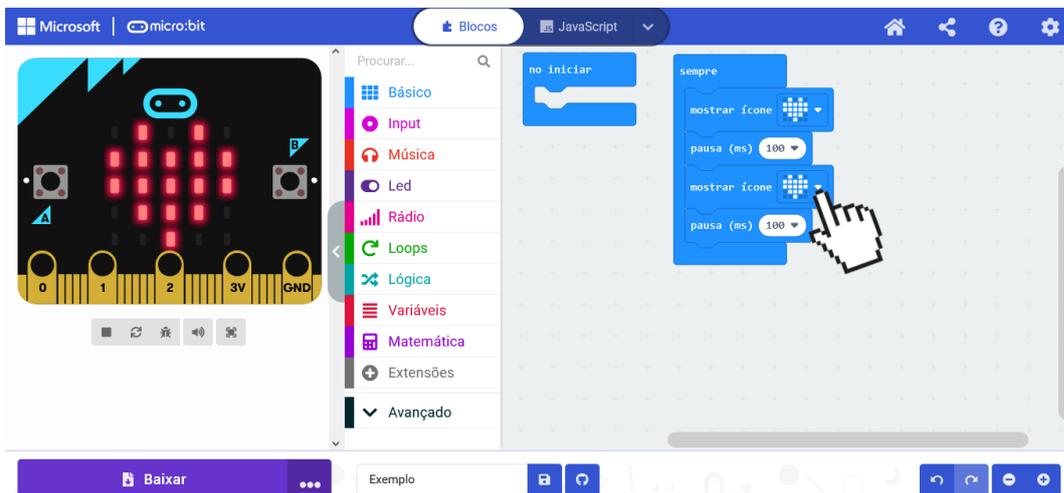


Figura 3.6: Selecionando um ícone.

# Coração lento

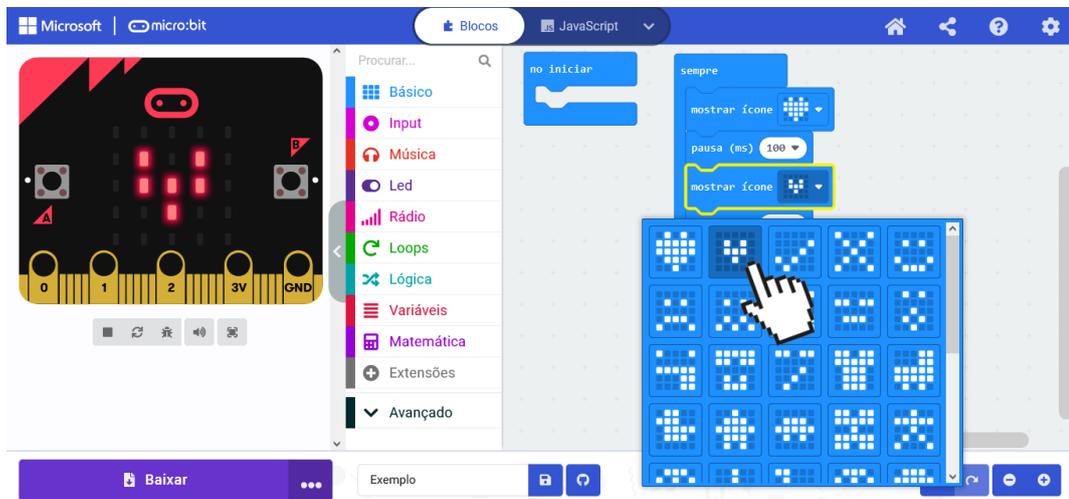


Figura 3.7: Alterando um ícone.

5. Em **pausa**, selecione a opção 500 ms, como nas Figuras 3.8 e 3.9;



Figura 3.8: Alterando o tempo da pausa.



# Coração lento

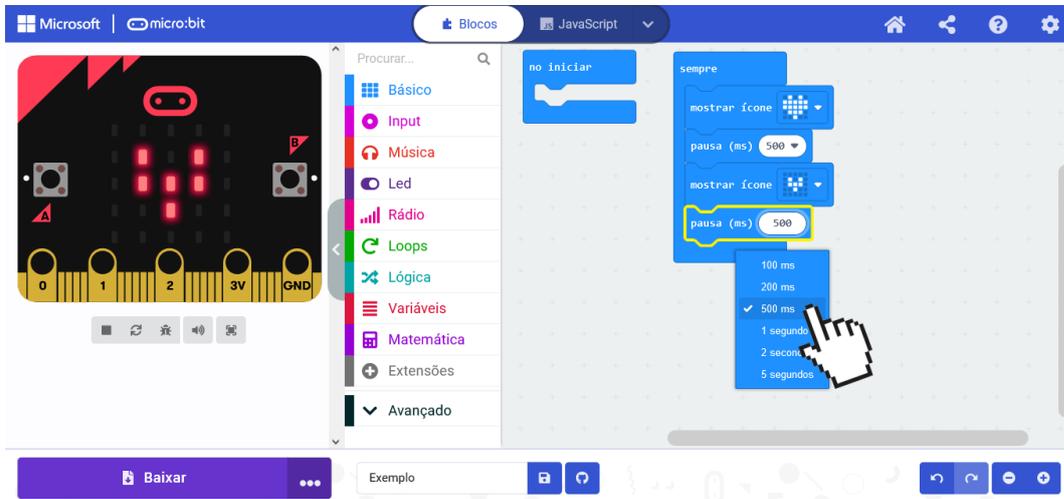


Figura 3.9: Alterando o tempo da pausa.

**Baixe o programa para o micro:bit, como na Figura 3.10.**

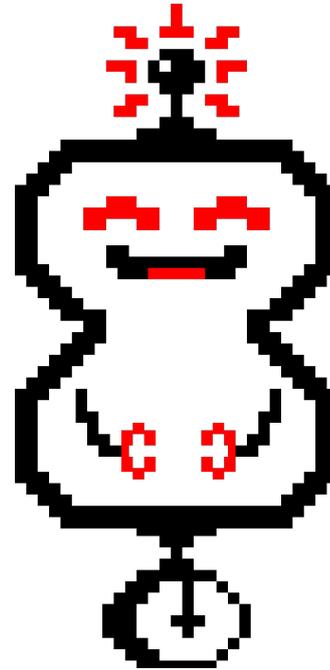


Figura 3.10: Baixando o código para o micro:bit.

## Coração lento

### **MICRO:DICA**

Mude os tempos das pausas e veja em quais velocidades, máxima e mínima, o coração pode pulsar.



[Clique aqui para acessar o código.](#)

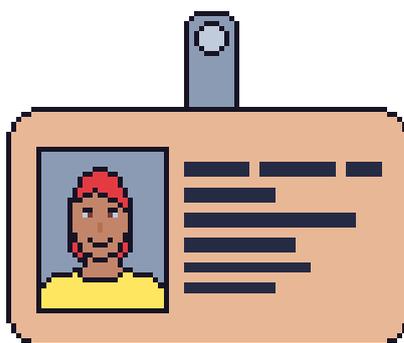




Coração lento



# Capítulo 4: Crachá



## Novos blocos:

Agora que já foi aprendido como programar imagens utilizando os LEDs do micro:bit, que tal se apresentar utilizando a plaquinha? Para isso utilize um novo bloco da aba **Básico**.

### mostrar string "Hello!"

Este bloco é usado para mostrar nos LEDs do micro:bit a mensagem que desejar. **Importante:** Não utilize ç e acentos, porque o micro:bit não reconhece esses caracteres.

mostrar string "Hello!"



# Crachá

## Passo a passo:

1. Adicione a função **mostrar string** “Hello!” no bloco **sempre**, como mostram as Figuras 4.1 e 4.2;

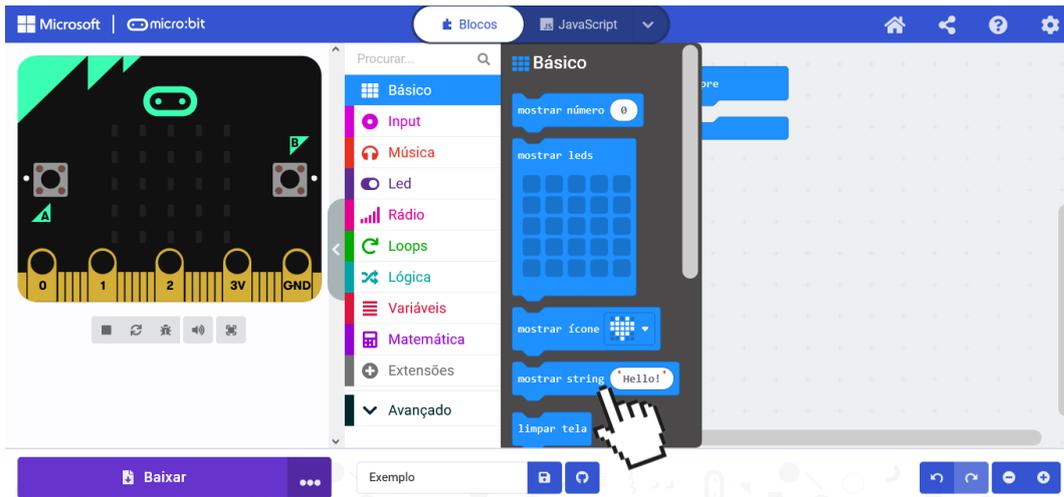


Figura 4.1: Selecionando o bloco mostrar string.

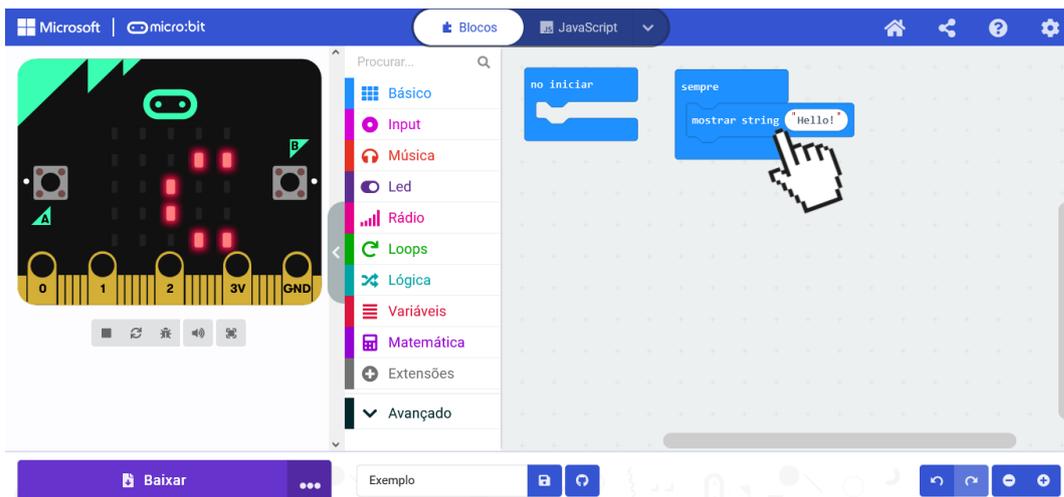


Figura 4.2: Montando o comando mostrar string.

# Crachá

2. Em seguida, substitua a palavra **Hello!** pelo seu nome, como na Figura 4.3. Para isso, é só clicar sobre a palavra **Hello!** para poder editar o texto;

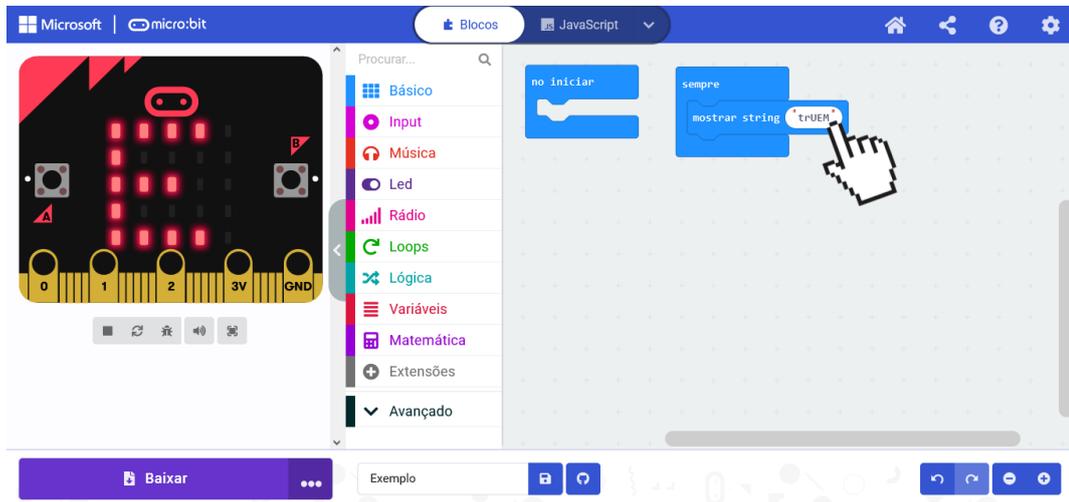


Figura 4.3: Escrevendo uma palavra no micro:bit.

3. Logo abaixo do comando **mostrar string**, coloque uma **pausa (ms)**, da aba **Básico**, como nas Figuras 4.4 e 4.5;

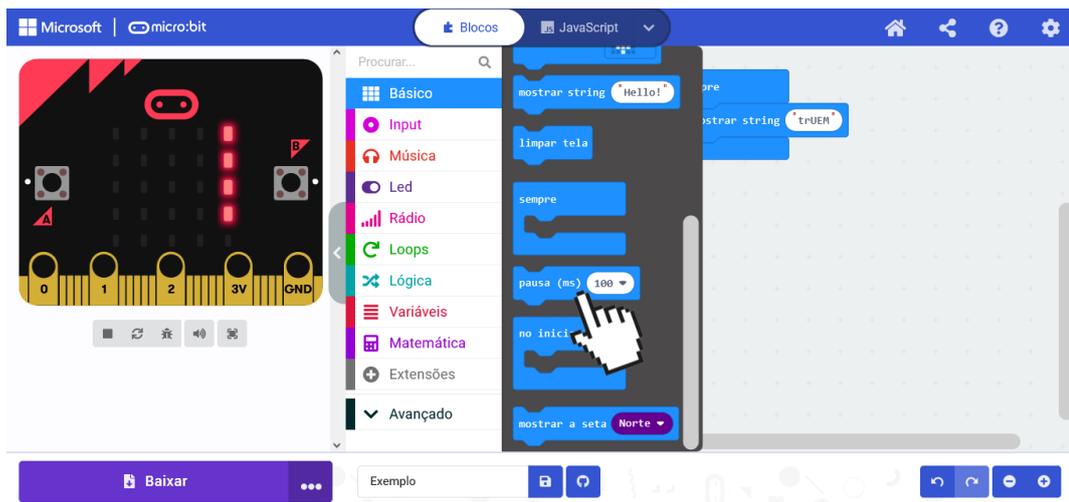
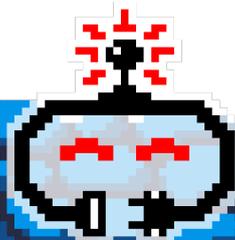


Figura 4.4: Selecionando uma pausa.



# Crachá

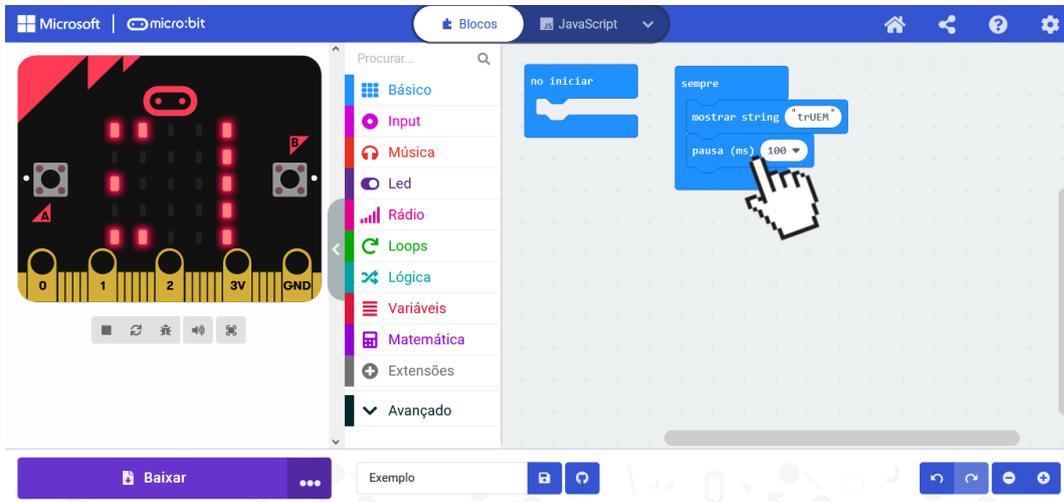


Figura 4.5: Montando o comando de pausa.

4. Na **pausa (ms)**, substitua **100** por **500**, para que o intervalo entre os comandos seja de meio segundo, como na Figura 4.6;

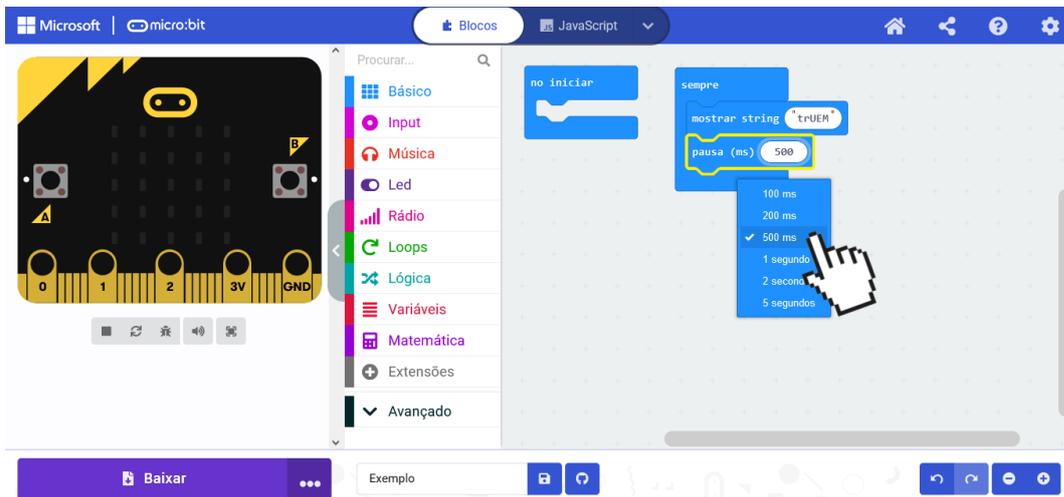


Figura 4.6: Alterando o tempo da pausa.

# Crachá

5. Selecione novamente na aba **Básico** o comando **mostrar string** "Hello!" e coloque-o logo abaixo da **pausa**, como mostram as Figuras 4.7 e 4.8. Troque o texto **Hello!** pela sua idade;

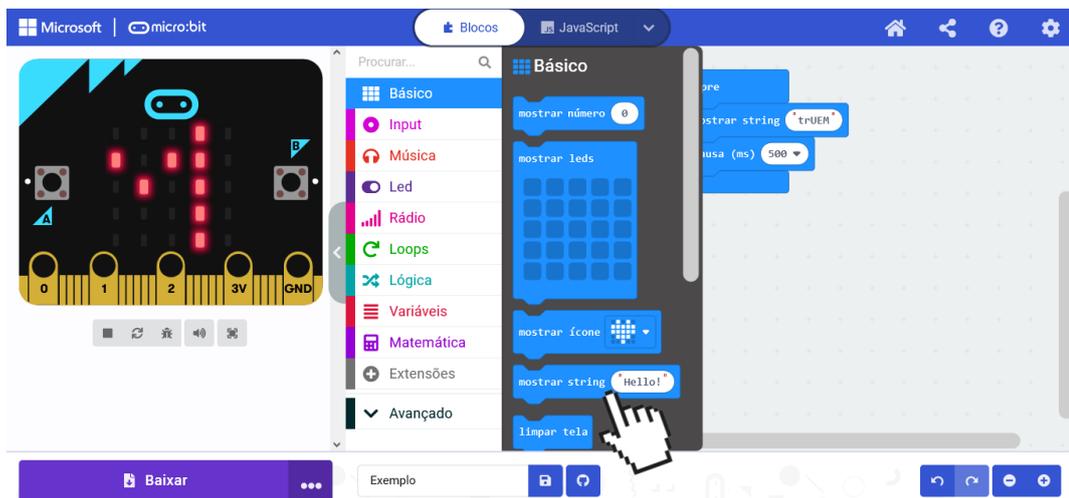


Figura 4.7: Selecionando o comando mostrar string.

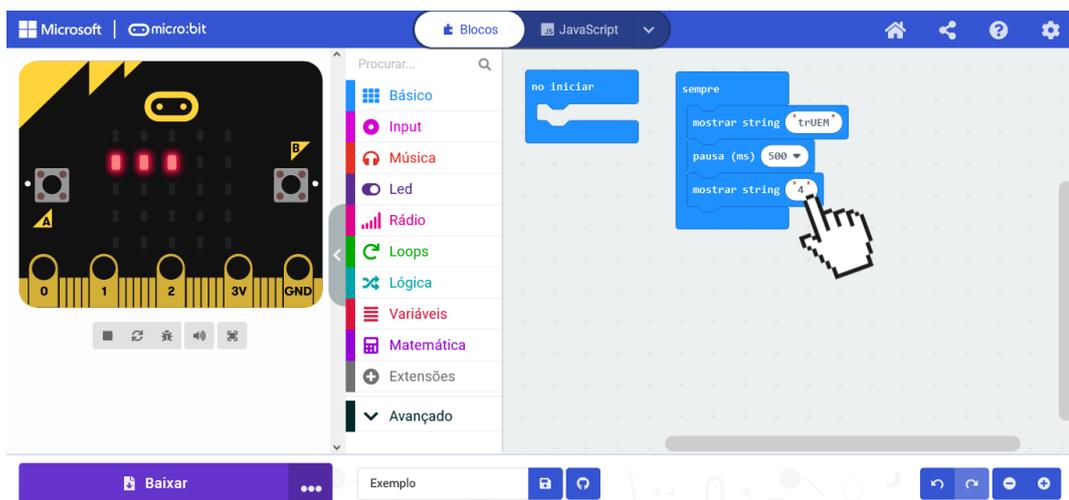


Figura 4.8: Escrevendo um número no micro:bit.



# Crachá

6. Pegue mais uma **pausa (ms)** e, novamente, substitua 100 por 500. Coloque-a abaixo do comando **mostrar string** com a sua idade, como nas Figuras 4.9 e 4.10;

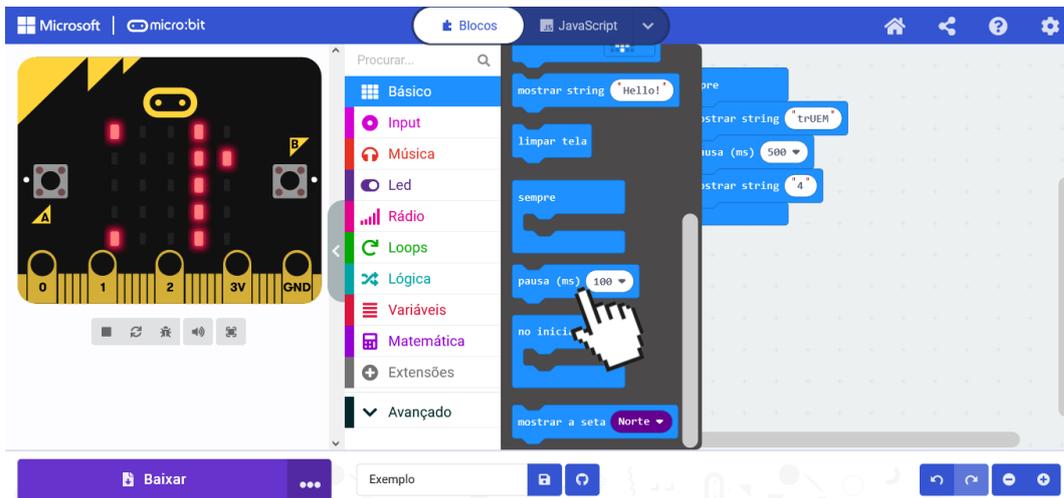


Figura 4.9: Selecionando uma pausa.

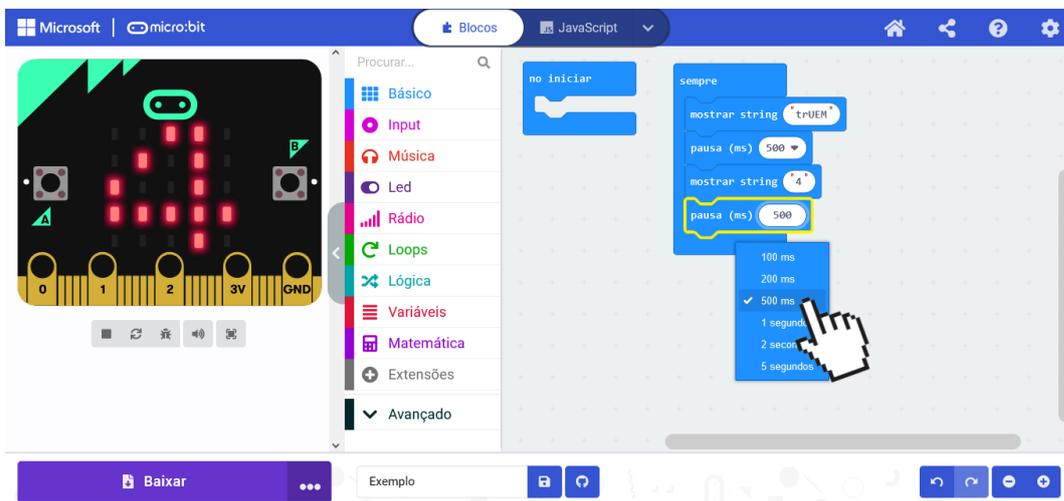


Figura 4.10: Alterando o tempo da pausa.

Baixe o programa para o micro:bit.

## Crachá

Agora com o crachá pronto, vá até a frente da sala e apresente-se para a turma!

[Clique aqui para acessar o código.](#)





Crachá



# Capítulo 5: no iniciar



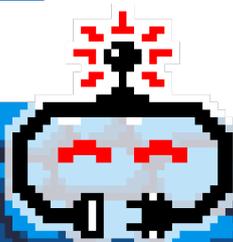
## Novos blocos:

Como visto na primeira atividade, existem alguns **blocos de acionamento** e, dentre eles, há o **no iniciar**. Faremos agora uma atividade com esse novo bloco, para entendermos como ele funciona.



**no iniciar**

O **no iniciar** é um bloco de comando que é executado toda vez que a placa é ativada.



## Passo a passo:

1. Selecione, na aba **Básico**, o comando **mostrar string** e coloque-o no bloco **no iniciar** como na Figura 5.1;



Figura 5.1: Selecionando o comando mostrar string.

2. Escreva, entre as aspas de **mostrar string**, a palavra **LIGADO**, como na Figura 5.2.

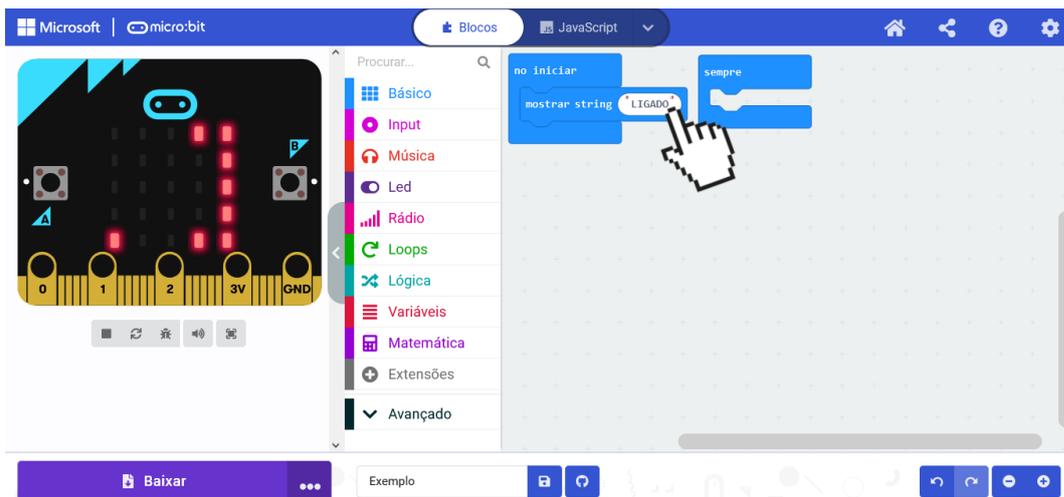


Figura 5.2: Escrevendo uma palavra no micro:bit.

no iniciar

Baixe o programa para o micro:bit, como na Figura 5.3.

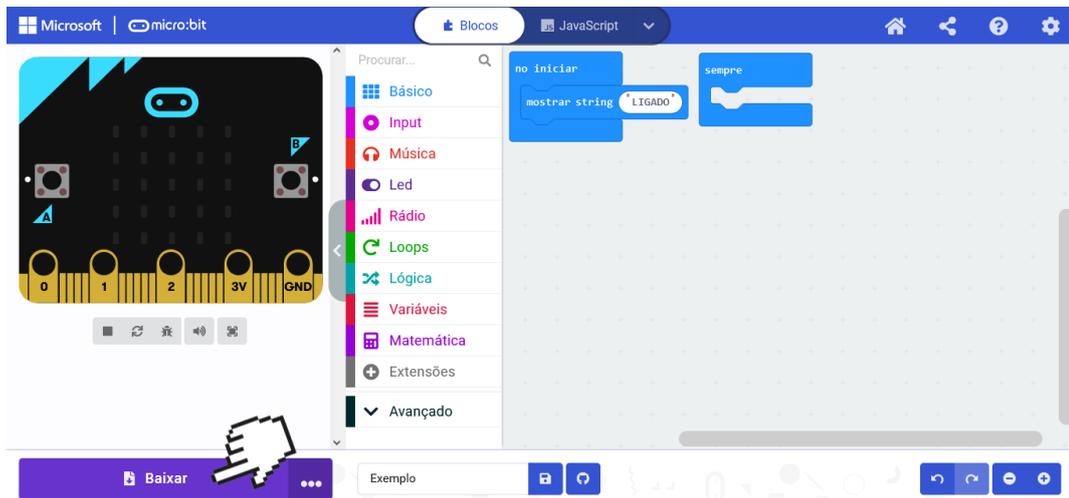


Figura 5.3: Baixando o programa para o micro:bit.

Percebeu que o LIGADO do início apareceu apenas uma vez? Quando colocávamos no bloco **sempre**, o que acontecia com as palavras? É assim que o **no iniciar** e o **sempre** irão funcionar nas próximas práticas.

[Clique aqui para acessar o código.](#)

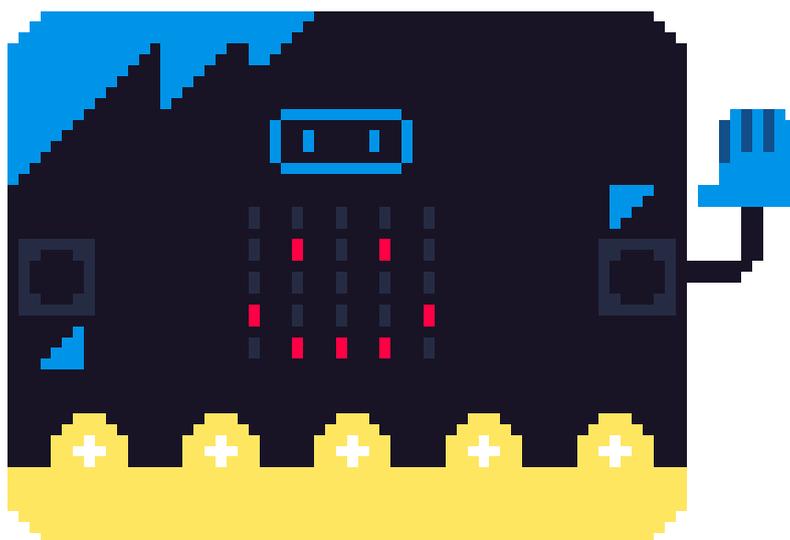




no iniciar



# Capítulo 6: Cumprimento do micro:bit



Sabe-se que ao colocar um comando no bloco **no iniciar** ele será executado apenas uma vez quando o micro:bit for iniciado e que, colocando no bloco **sempre**, os comandos se repetem. Que tal unir esses dois blocos de acionamento para ver o que acontece?

Nesta atividade faremos com que o micro:bit cumprimente dizendo um “**OLÁ**” e dê um sorriso simpático quando acionado.



# Cumprimento do micro:bit

## Passo a passo:

1. Selecione na aba **Básico** o comando **mostrar string**, como na Figura 6.1;

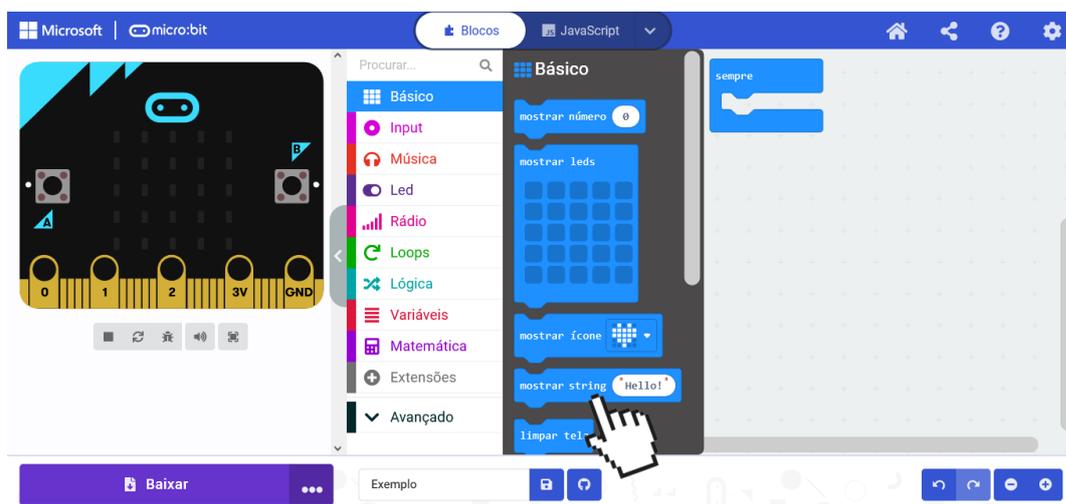


Figura 6.1: Selecionando o comando mostrar string.

2. Coloque-o no bloco de acionamento **no iniciar**, como mostra a Figura 6.2;

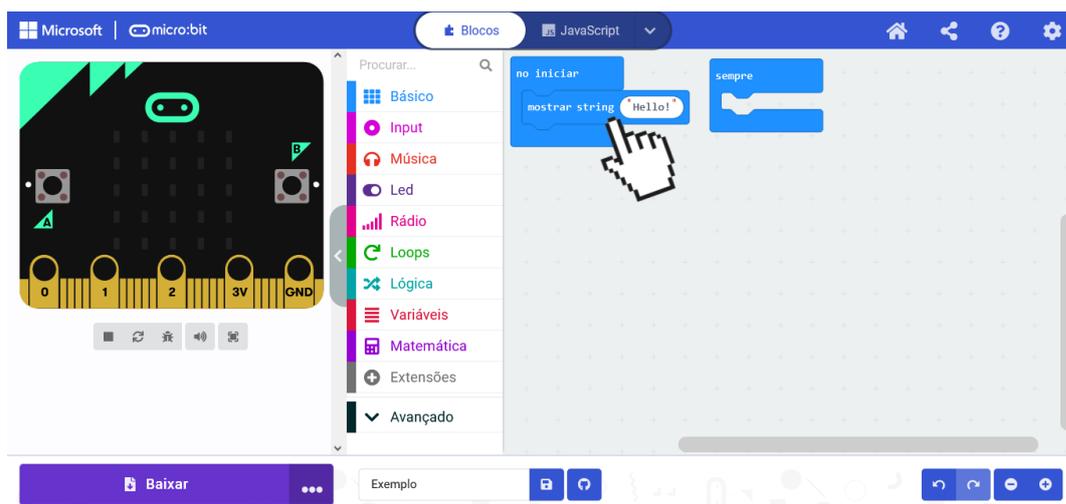


Figura 6.2: Montando um comando.

# Cumprimento do micro:bit

3. Escreva entre as aspas de **mostrar string** a palavra **OLA**, com letra maiúscula e sem acento, tal como na Figura 6.3;

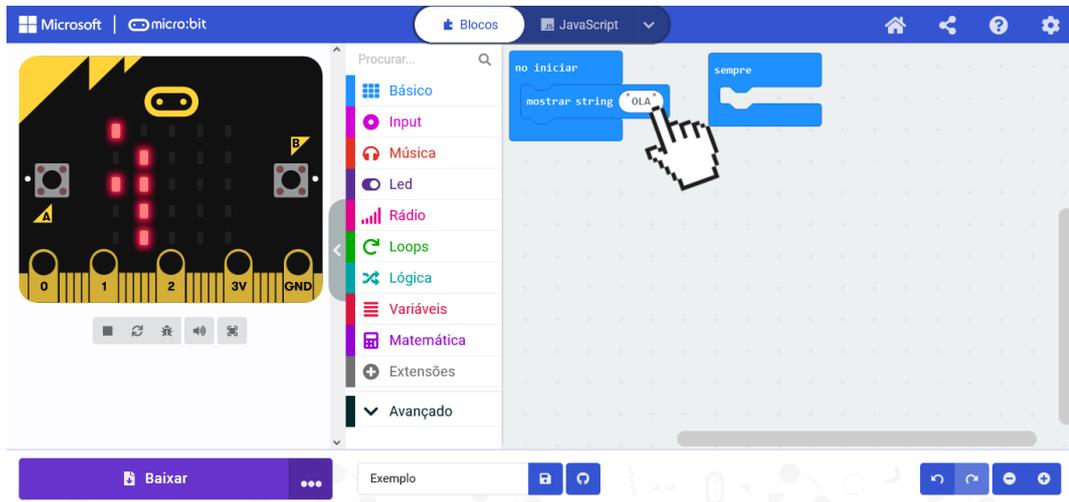


Figura 6.3: Escrevendo no micro:bit.

4. No **sempre** leve o comando **mostrar ícone**, da aba **Básico**, veja na Figura 6.4;



Figura 6.4: Montando um comando.



# Cumprimento do micro:bit

- Escolha, no comando **mostrar ícone**, a **carinha feliz**, como mostra a Figura 6.6;

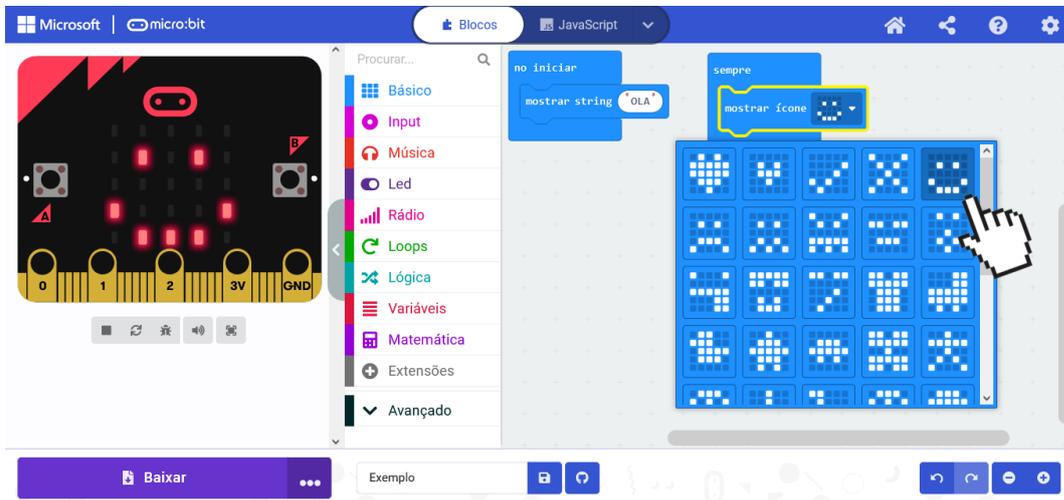


Figura 6.5: Selecionando um ícone.

**Baixe o programa para o micro:bit, como indicado na Figura 6.6.**

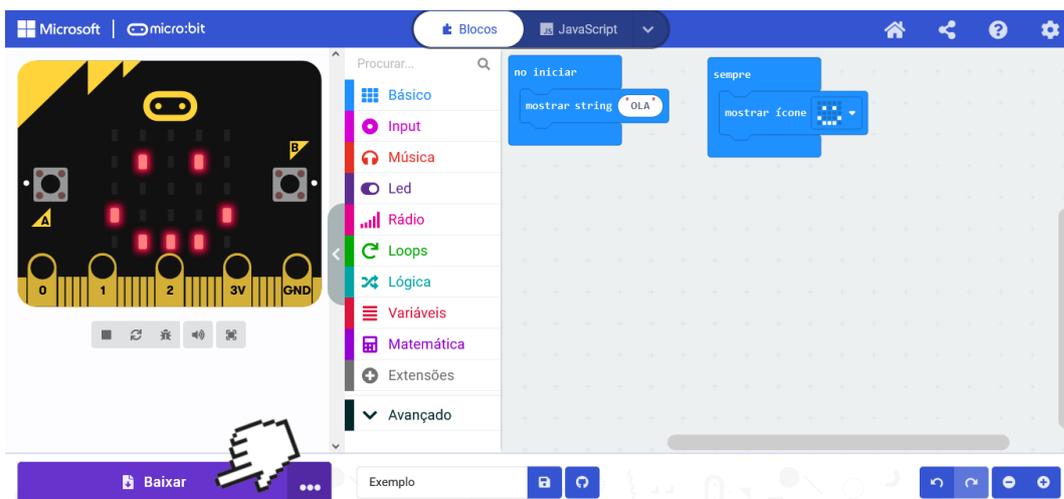


Figura 6.6: Baixando o código para o micro:bit.

## Cumprimento do micro:bit

Você percebeu que o OLA do início foi mostrado apenas uma vez e a carinha do bloco **sempre** ficou aparecendo nos LEDs? É assim que o **no iniciar** e o **sempre** irão funcionar nas próximas práticas.

[Clique aqui para acessar o código.](#)





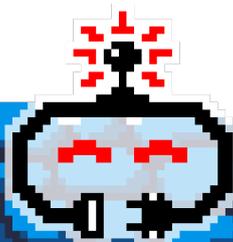
Cumprimento do  
micro:bit



# Capítulo 7: Fogos de artifício



Agora que já se sabe algumas funções, como por exemplo, colocar tempo entre os comandos de desenhos que aparecem nos LEDs do micro:bit, que tal criar um efeito de fogos de artifício?



# Fogos de artifício

## Passo a passo:

1. Selecione o bloco **no iniciar** e coloque o comando **mostrar leds**, como nas Figuras 7.1, 7.2;

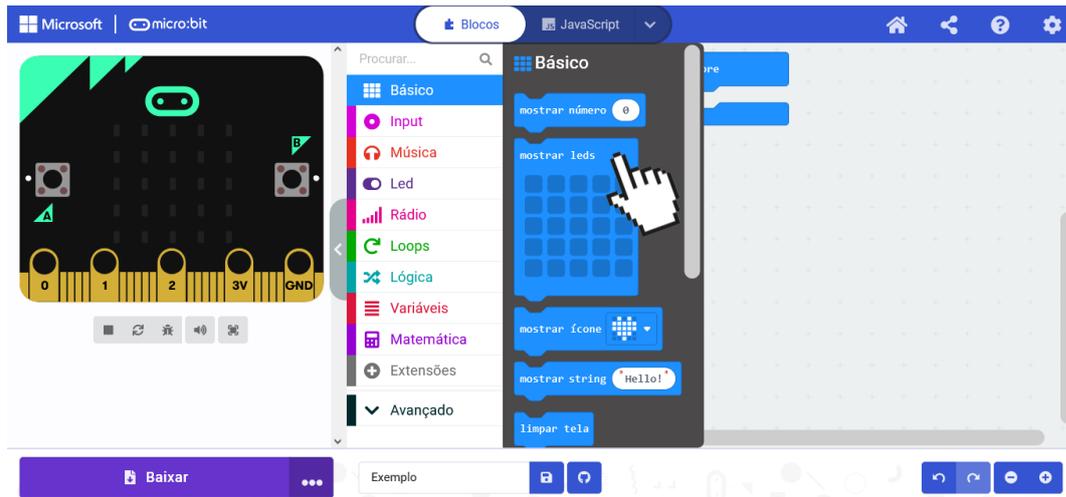


Figura 7.1: Selecionando um comando.



Figura 7.2: Montando um comando.

# Fogos de artifício

2. Em sequência, desenhe um **sim** como mostrado na Figura 7.3, para indicar que o pavio foi aceso e, assim, reforçar o funcionamento do bloco de acionamento **no iniciar**;

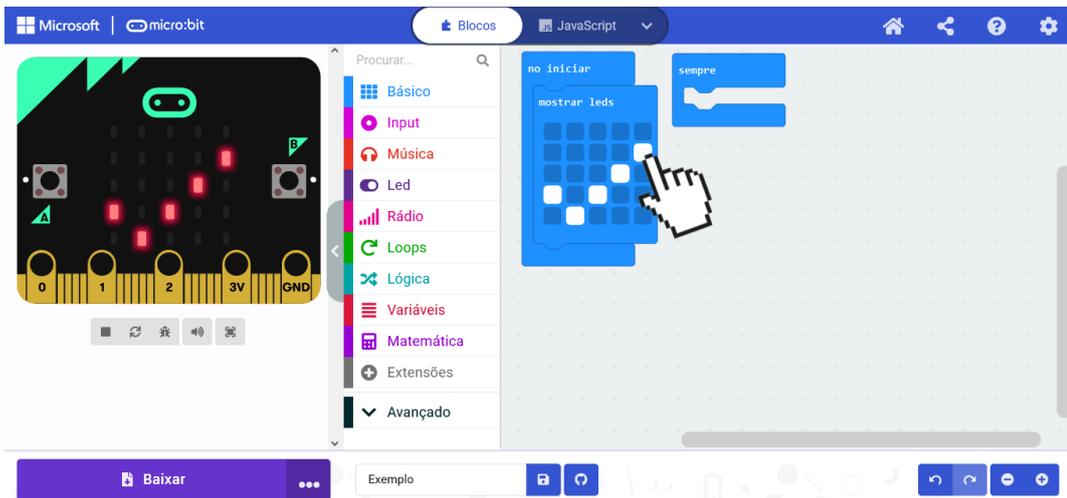


Figura 7.3: Desenhando um ícone.

3. Em seguida, adicione uma **pausa** de **1 segundo** como demonstrado nas Figuras 7.4, 7.5 e 7.6

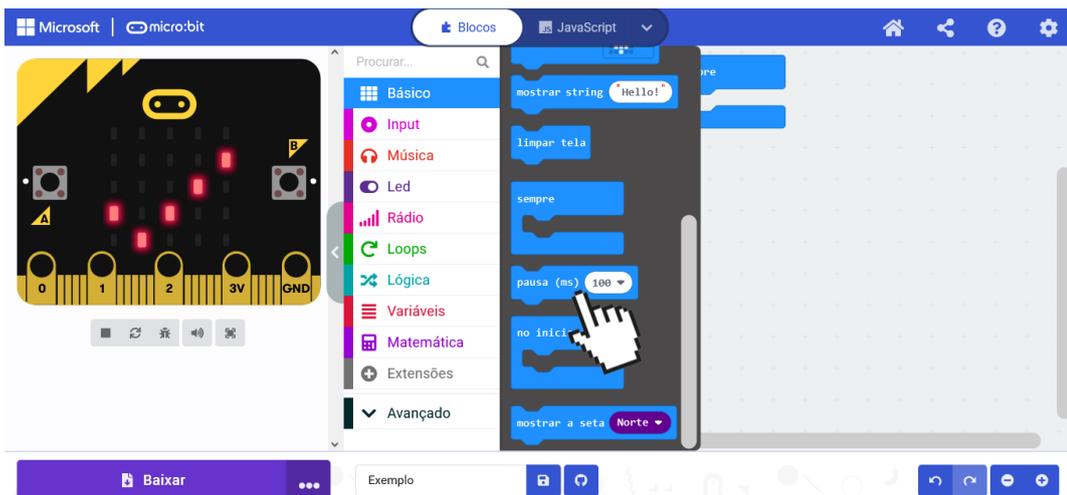


Figura 7.4: Selecionando uma pausa.



# Fogos de artifício

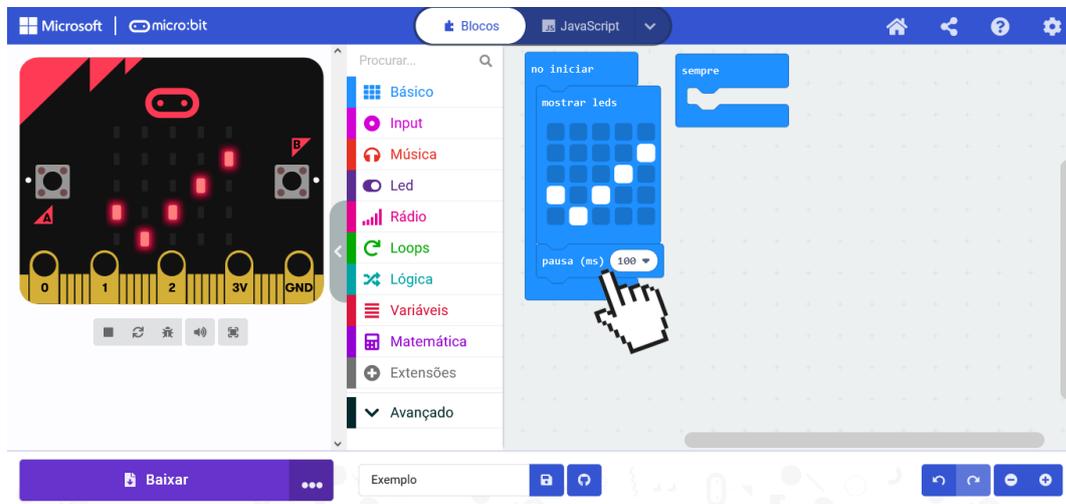


Figura 7.5: Montando uma pausa.



Figura 7.6: Alterando o tempo de uma pausa.

# Fogos de artifício

4. Em **sempre**, coloque o **mostrar leds** três vezes. Entre cada mostrador de LEDs, coloque uma **pausa** de **500 ms**, como nas Figuras 7.7, 7.8, 7.9, 7.10, 7.11 e 7.12;

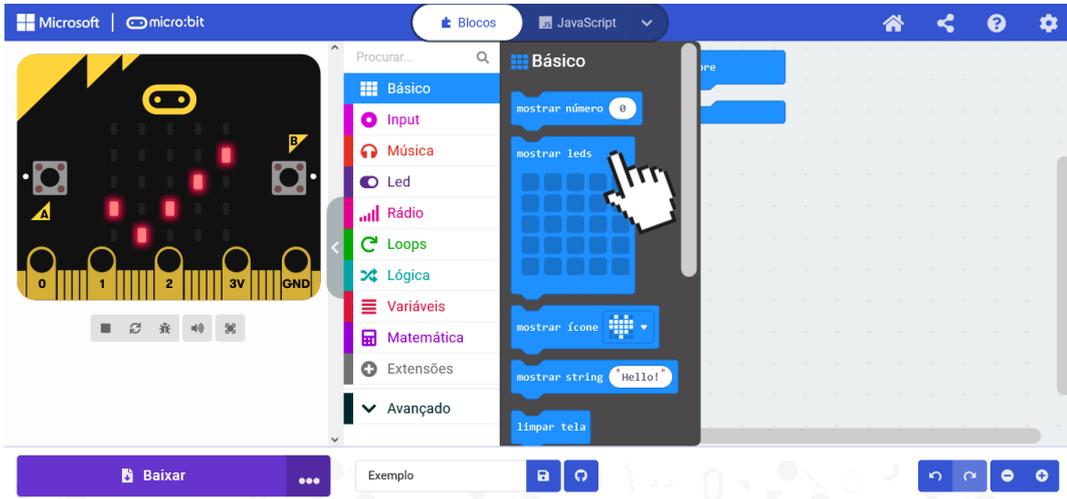


Figura 7.7: Selecionando o mostrar leds.



Figura 7.8: Montando o comando mostrar leds.



# Fogos de artifício

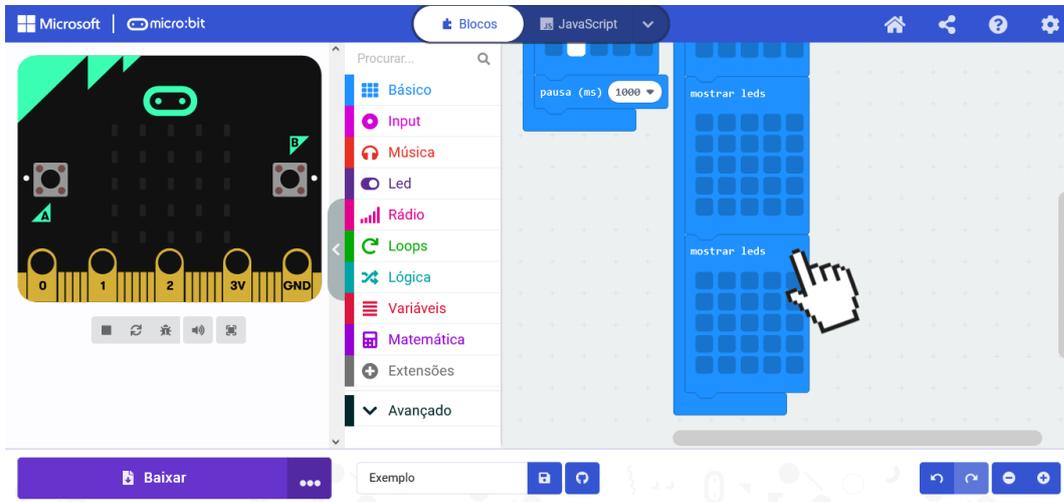


Figura 7.9: Montando comandos mostrar leds.

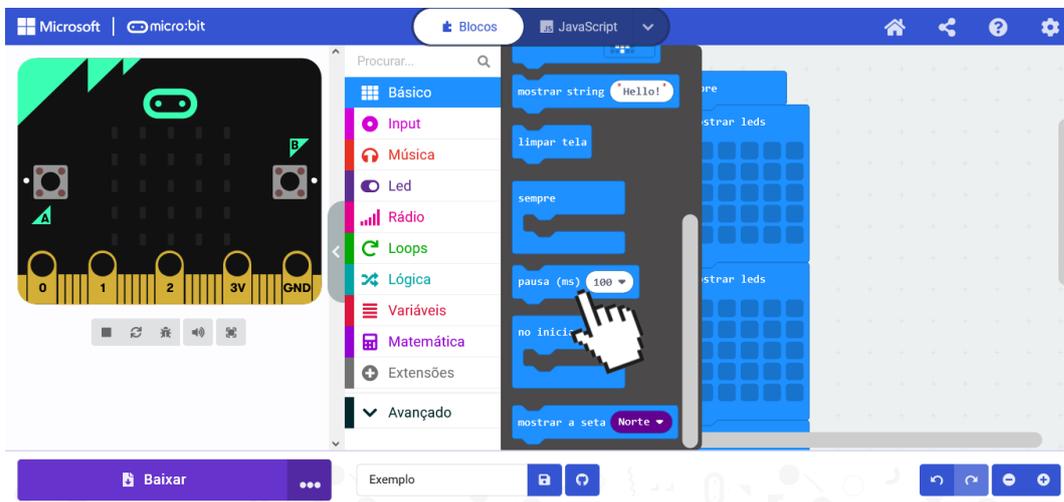


Figura 7.10: Selecionando uma pausa.

# Fogos de artifício

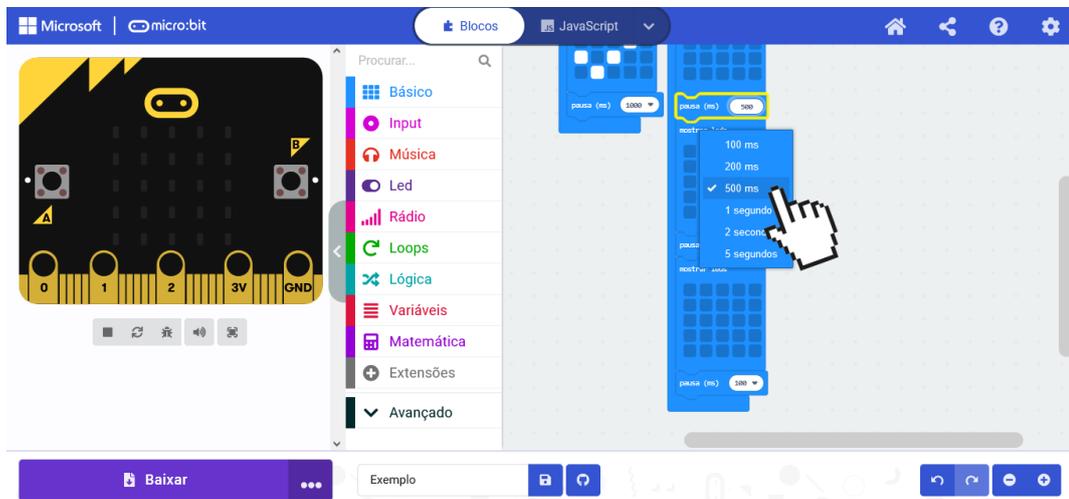


Figura 7.11: Alterando o tempo de uma pausa.



Figura 7.12: Sequência dos blocos para o efeito de fogos de artifício.



# Fogos de artifício

5. Acenda apenas o LED do meio no primeiro **mostrar leds**, como na Figura 7.13;



Figura 7.13: Acendendo um LED.

6. No segundo, contorne o LED acesso anteriormente. Veja como fazer na Figura 7.14;

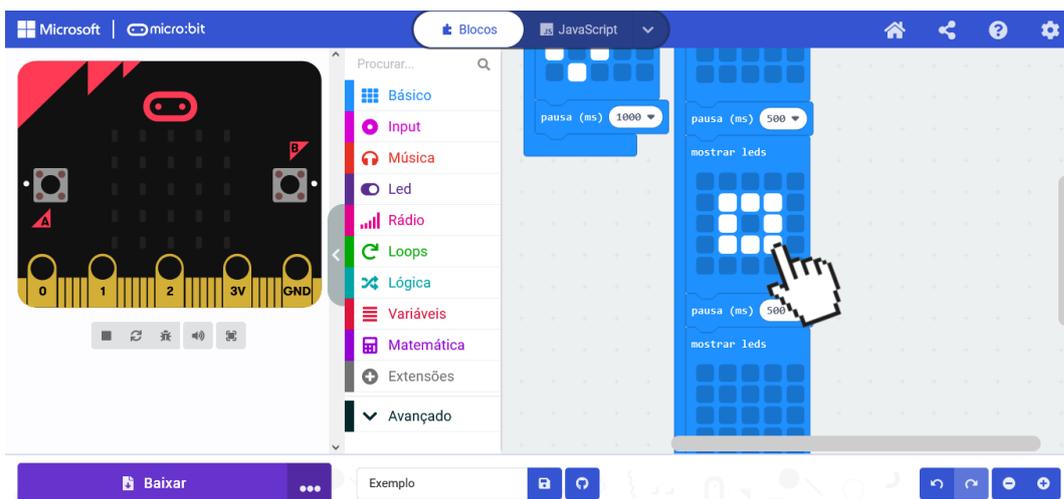


Figura 7.14: Acendendo LEDs.

# Fogos de artifício

7. Por último, contorne toda a borda de LEDs, como representado na Figura 7.15;



Figura 7.15: Acendendo LEDs.

**Baixe o programa para o micro:bit, Figura 7.16, e veja o efeito dos fogos de artifício!**

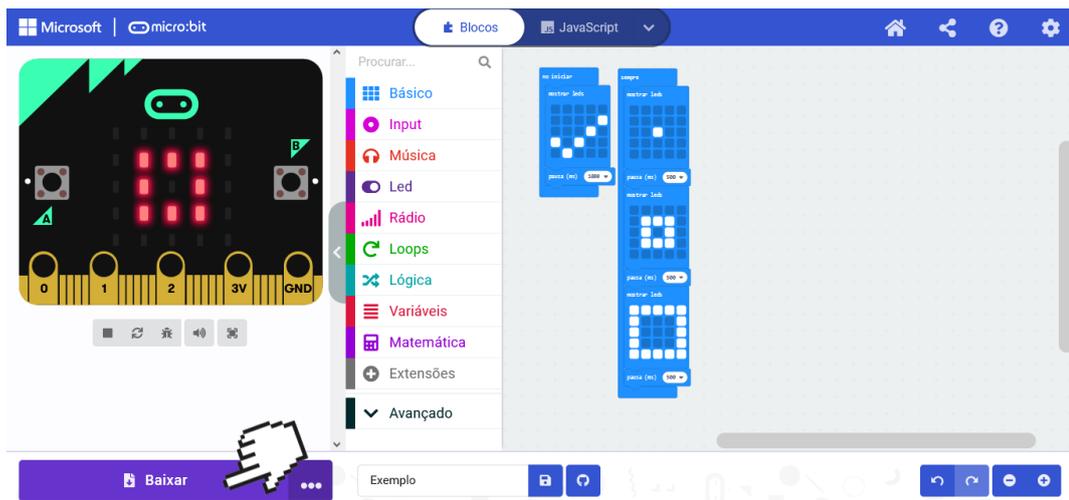


Figura 7.16: Baixando o código para o micro:bit.

[Clique aqui para acessar o código.](#)





Fogos de artifício



# Capítulo 8: Usando os botões

## Input

Agora que já foi aprendido como a primeira aba da biblioteca funciona, conheceremos neste capítulo novos componentes do micro:bit, os botões.

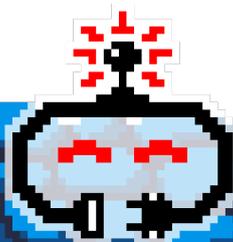
O micro:bit tem dois botões, o botão **A** e o botão **B** que, assim como nos controles remotos, realizam comandos quando são pressionados. Por exemplo, ao apertar o botão vermelho do controle remoto, a TV liga ou desliga. O mesmo acontecerá com o micro:bit.

A partir da aba **Input**, que significa *entrada* e que permite o uso de acessórios da placa, os botões serão usados como ferramentas para que o micro:bit realize o comando que foi programado.

### no botão A pressionado



É o comando usado para dizer o que vai acontecer quando o botão **A** for pressionado. O acionamento pode ser alterado para o botão **B** e também para **A+B**, que representa pressionar os dois botões ao mesmo tempo.



# Usando os botões

## Passo a passo:

1. No bloco **sempre**, coloque o comando **mostrar ícone**, da aba **Básico** e escolha a **carinha feliz**, como mostram as Figuras 8.1, 8.2 e 8.3;



Figura 8.1: Selecionando um comando

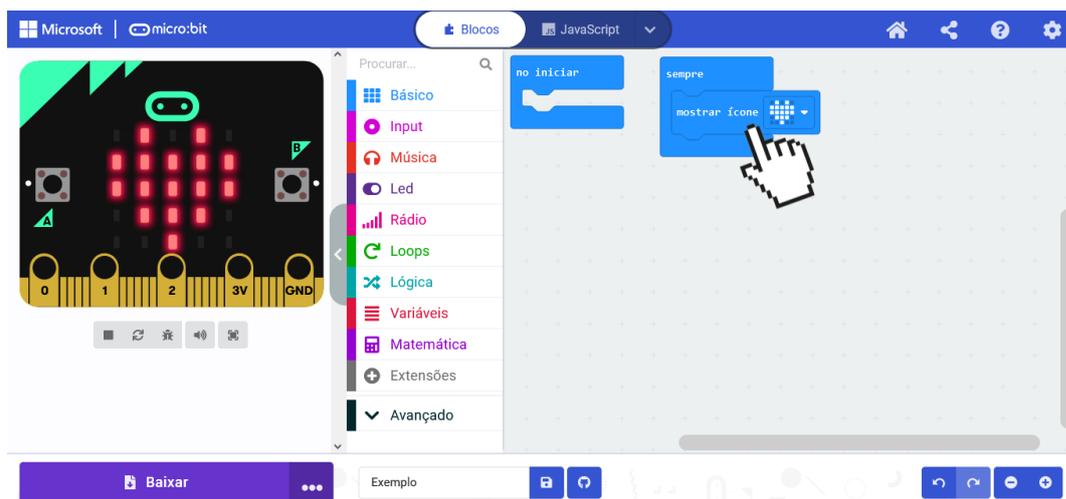


Figura 8.2: Mostrando um ícone.

# Usando os botões

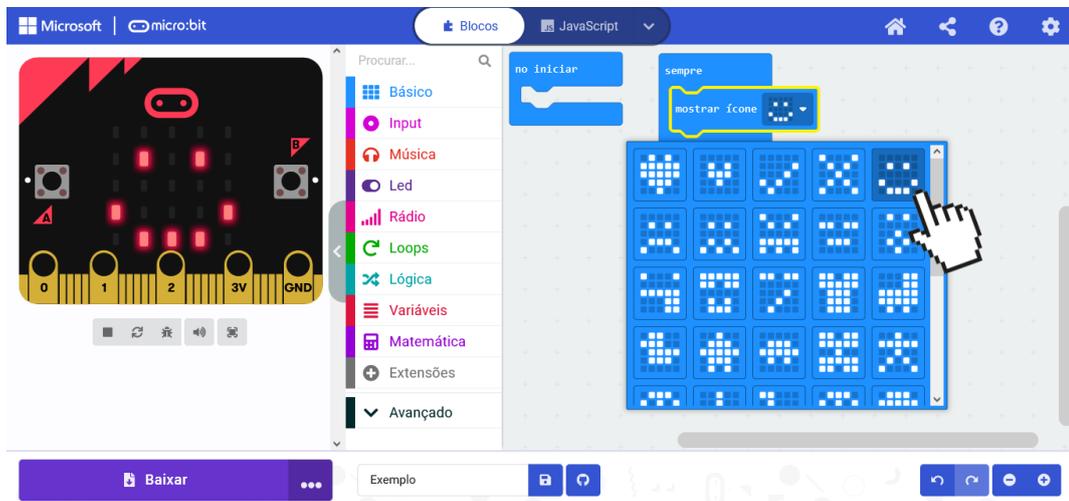


Figura 8.3: Alterando um ícone.

2. Em seguida, vá até a biblioteca e selecione, na aba **Input**, o bloco **no botão A pressionado**, como na Figura 8.4;

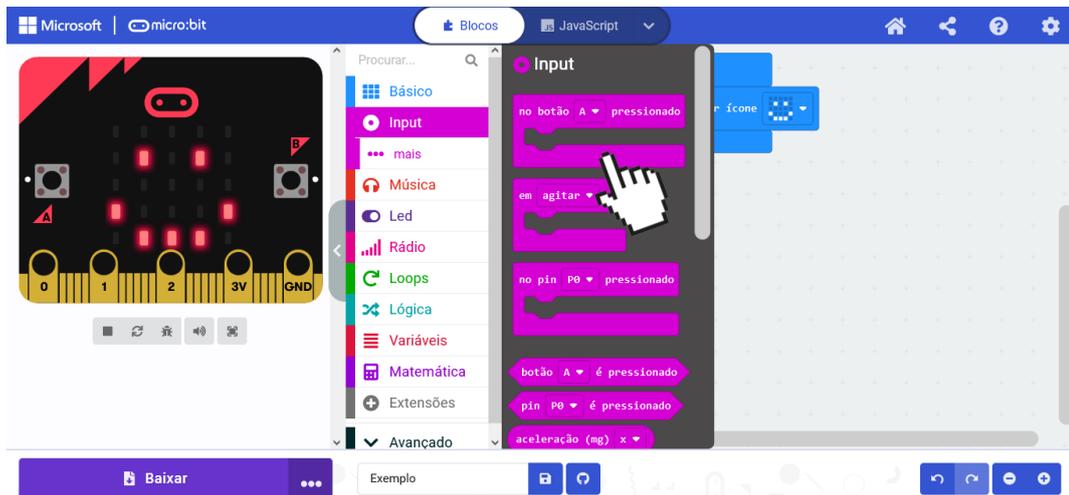


Figura 8.4: Selecionando um bloco de acionamento.



## Usando os botões

3. Pegue o comando **mostrar leds** e coloque-o no bloco **no botão A pressionado**. Desenhe, como em uma das atividades anteriores, a letra inicial do seu nome para que ela apareça nos LEDs do micro:bit. As Figuras 8.5, 8.6 e 8.7 indicam a sequência da montagem;

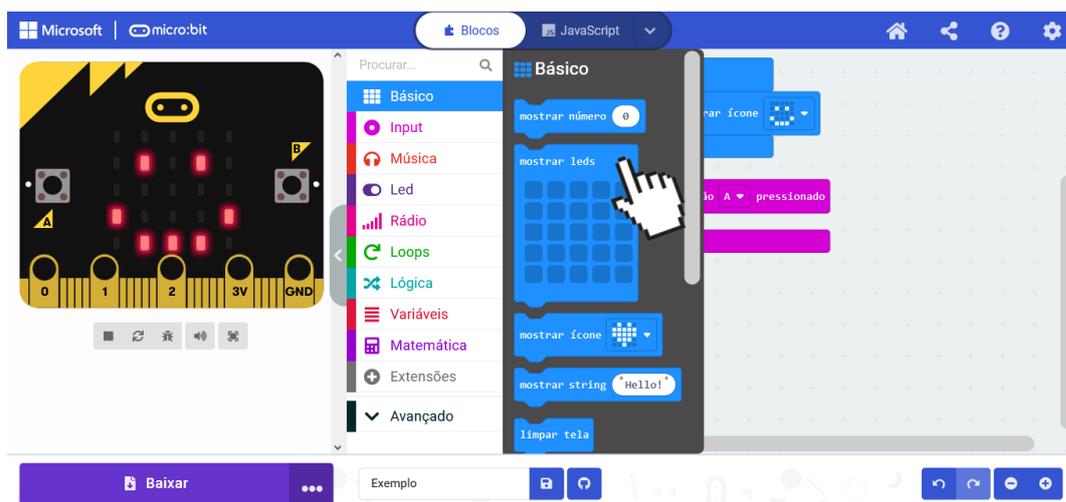


Figura 8.5: Selecionando um comando.



Figura 8.6: Montando um comando.

# Usando os botões



Figura 8.7: Desenhando uma letra no mostrar leds.

**Baixe o programa para o micro:bit, como na Figura 8.8, e teste o botão A.**



Figura 8.8: Baixando o código para o micro:bit.

[Clique aqui para acessar o código.](#)





Usando os botões



## Capítulo 9: Quadro mágico



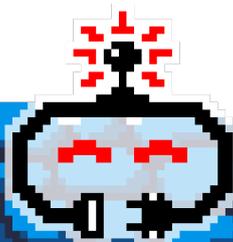
### Limpendo a tela

Depois de aprendermos o comando usado para ativar botões do micro:bit, que tal aprender agora a apagar os desenhos que aparecerem nos LEDs? Para isso, é necessário conhecer um novo comando da aba **Básico**, o **limpar tela**.

limpar tela

limpar tela

É a borracha dos LEDs que, quando aplicada, deve apagar tudo o que aparecia anteriormente.



# Quadro mágico

## Passo a passo:

1. Vá até a biblioteca e selecione, na aba **Input**, o bloco **no botão A pressionado**.  
Figuras 9.1 e 9.2;



Figura 9.1: Selecionando um bloco de acionamento.

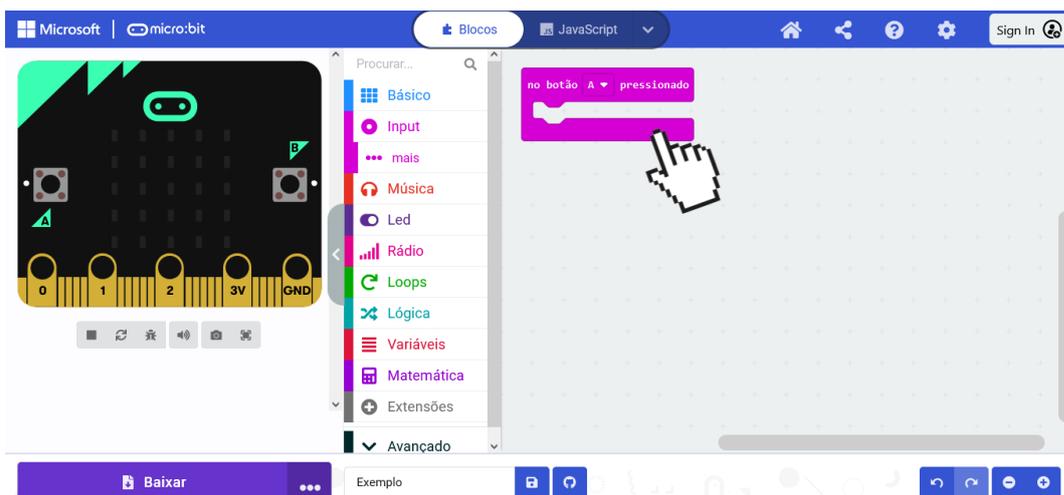


Figura 9.2: Acionamento selecionado.

# Quadro mágico

2. Selecione o comando **mostrar leds** e desenhe um sorriso, como nas Figuras 9.3, 9.4 e 9.5, para que ele apareça nos LEDs do micro:bit;

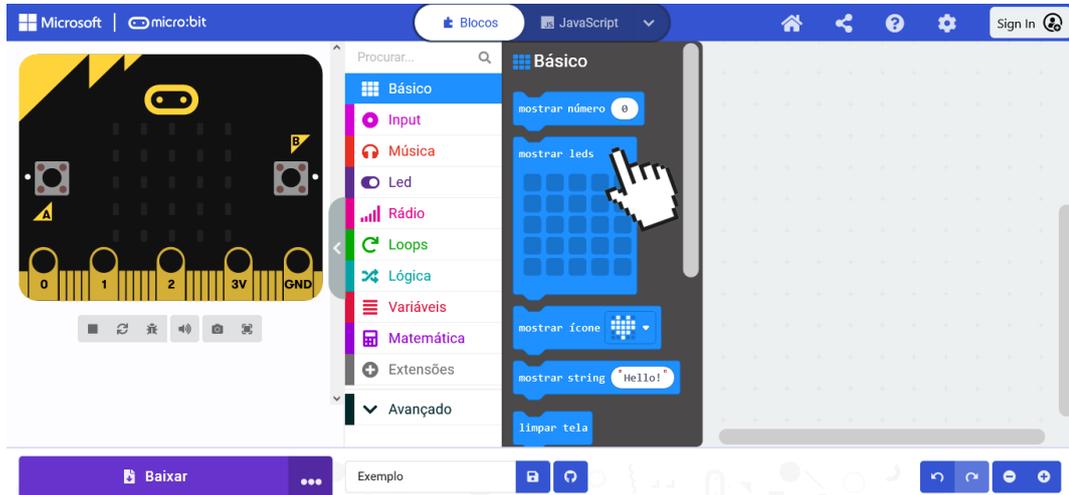


Figura 9.3: Selecionando o comando mostrar leds.

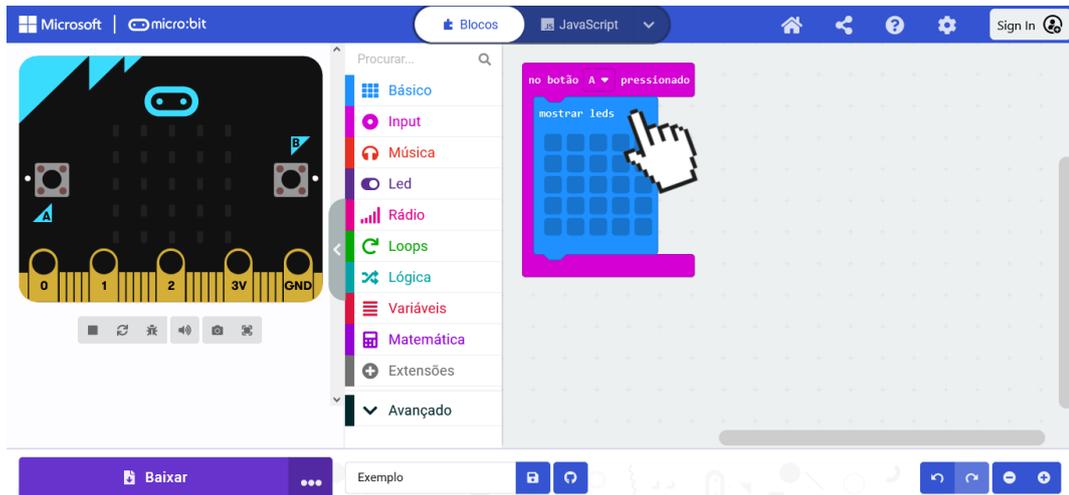


Figura 9.4: Montando o comando mostrar leds.



# Quadro mágico

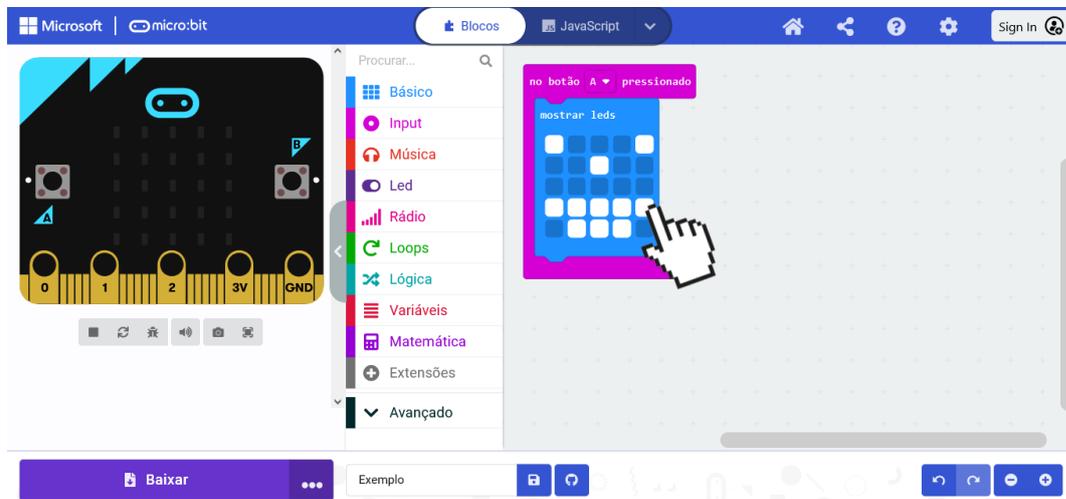


Figura 9.5: Desenhando no mostrar leds.

**Baixe o programa para o micro:bit!**

Conseguiu ver sua obra?  
Quando?  
Mas ela some?  
Como faremos para apagá-la?

# Quadro mágico

## Passo a passo:

1. Para limpar a tela, pegue mais um bloco de acionamento **no botão A pressionado**, Figuras 9.6 e 9.7;



Figura 9.6: Selecionando um bloco de acionamento.

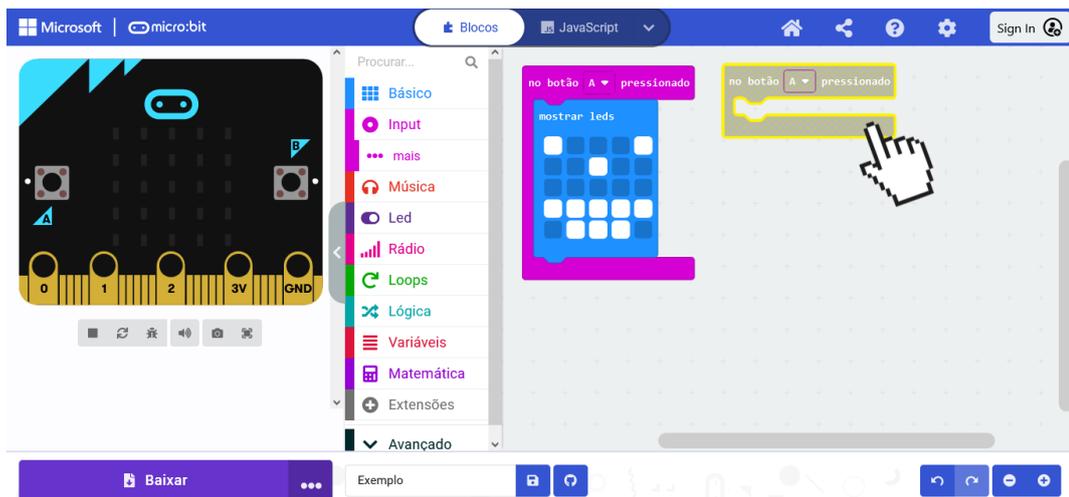


Figura 9.7: Montando um comando.



# Quadro mágico

2. Agora, defina o botão para a letra **B**, clicando na seta à direita da letra **A**, como na Figura 9.8;

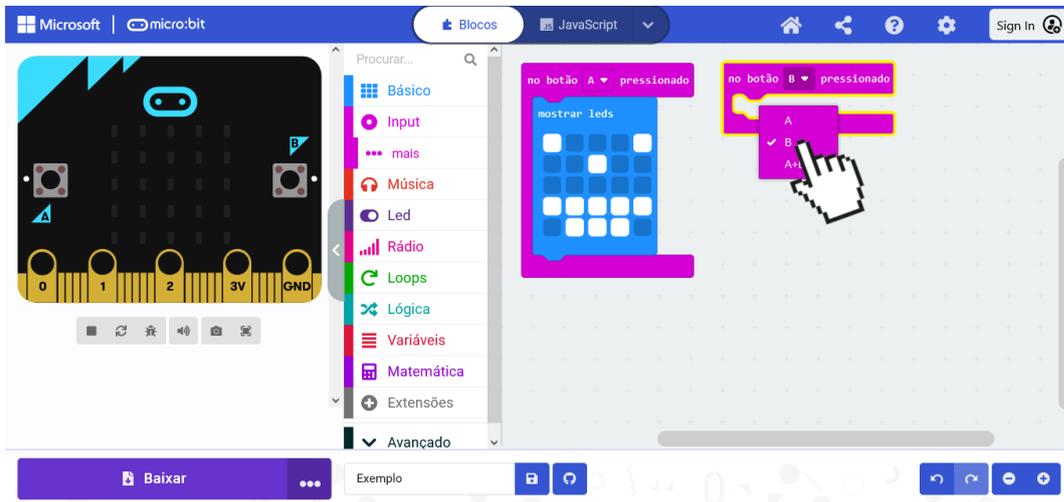


Figura 9.8: Selecionando o botão B.

3. Dentro desse bloco, coloque o comando **limpar tela**, que está na aba **Básico**, Figuras 9.9 e 9.10. Assim, toda vez que o botão **B** for pressionado a tela vai ser limpa.

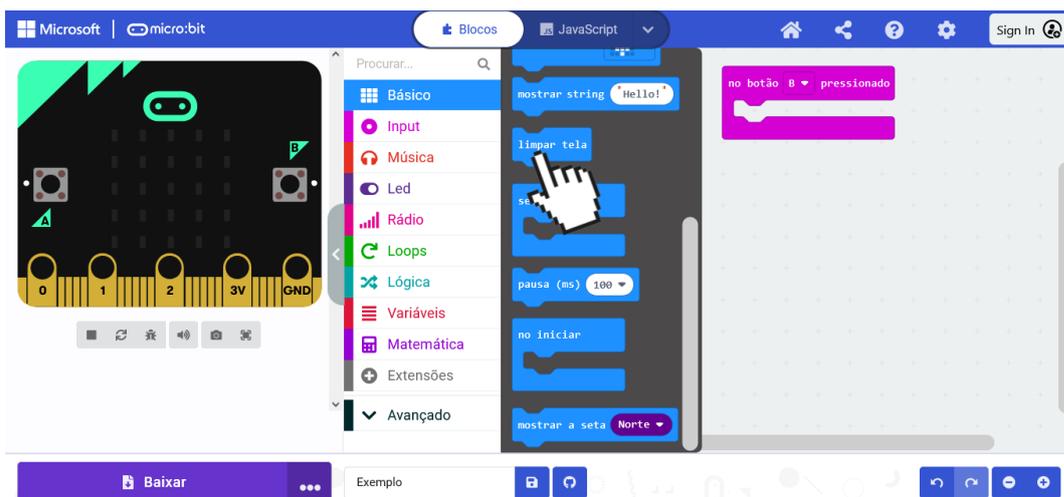


Figura 9.9: Selecionando um comando.

# Quadro mágico

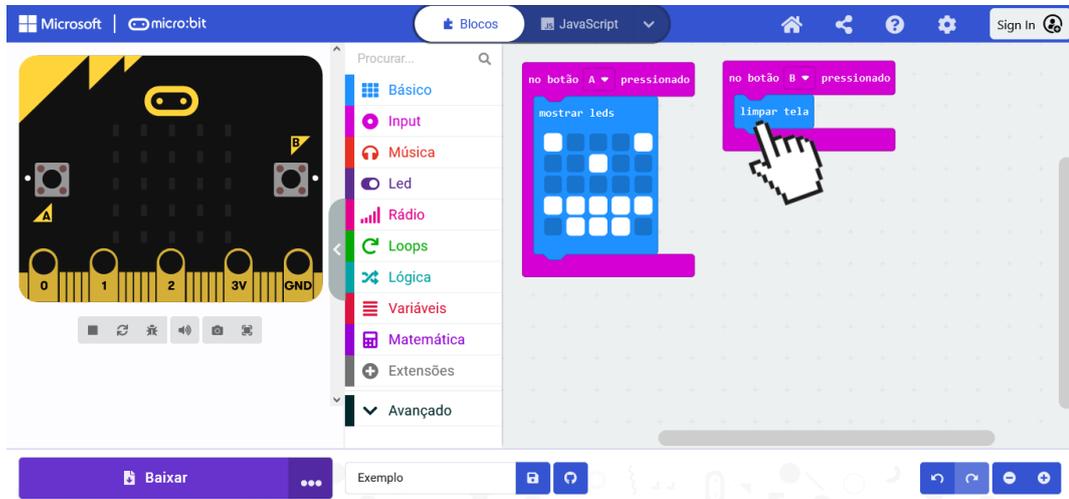


Figura 9.10: Montando um comando.

Baixe o programa para o micro:bit e veja seu desenho surgir e sumir quando os botões **A** e **B** forem pressionados alternadamente.

Você notou que quando pressionamos o mesmo botão duas vezes seguidas, os LEDs do micro:bit não mudam? Isso acontece porque foi definido para cada botão apenas um comando, ou seja, o botão **A** acende os LEDs e o botão **B** limpa a tela, apagando os LEDs.

[Clique aqui para acessar o código.](#)





Quadro mágico



## Capítulo 10: Os sons das emoções



Será que dá para mostrar as emoções acima com sons? Com a versão 2 do micro:bit, que é a que está sendo usada neste livro, é possível mostrar um ícone e emitir um som, simultaneamente, usando a aba **Música** da biblioteca de funções.

Continuaremos usando os botões para acionar o código, mas agora eles terão novas funções com os comandos **play sound**, **risadinha** e **until done**.

**play sound risadinha until done**

play sound

risadinha ▼

until done ▼

Este comando é usado para tocar um som escolhido sempre que ativado. Além de escolher o som, pode-se escolher se ele será tocado apenas uma vez até que acabe, **until done**, ou como segundo plano, **in background**, na execução do código. No segundo caso, o som que estiver sendo tocado será interrompido quando o comando de outro som for acionado.



# Os sons das emoções

## Passo a passo:

1. Vá para a aba **Input** e selecione dois blocos de acionamento **no botão A pressionado**, como nas Figuras 10.1, 10.2, 10.3 e 10.4;

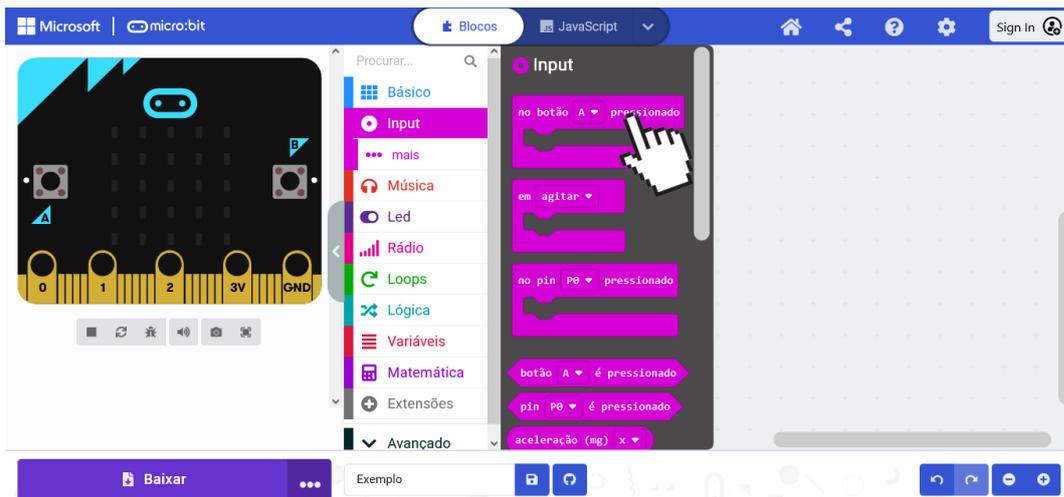


Figura 10.1: Selecionando um bloco de acionamento.

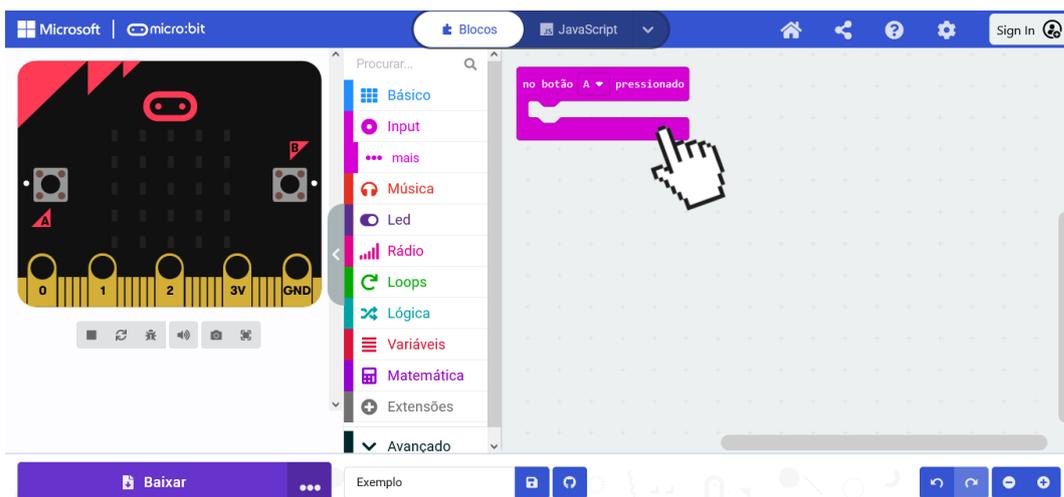


Figura 10.2: Montando um bloco de acionamento.

# Os sons das emoções

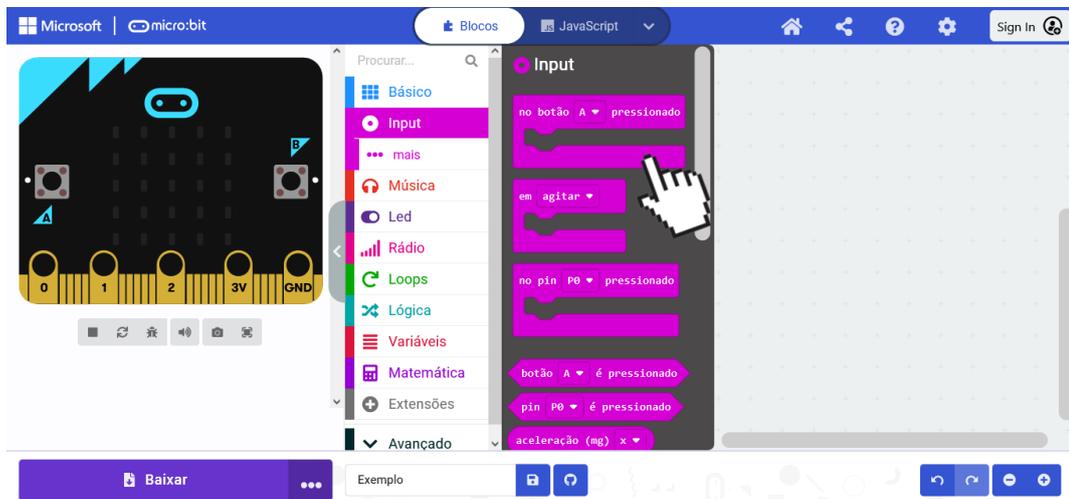


Figura 10.3: Selecionando um bloco de acionamento.

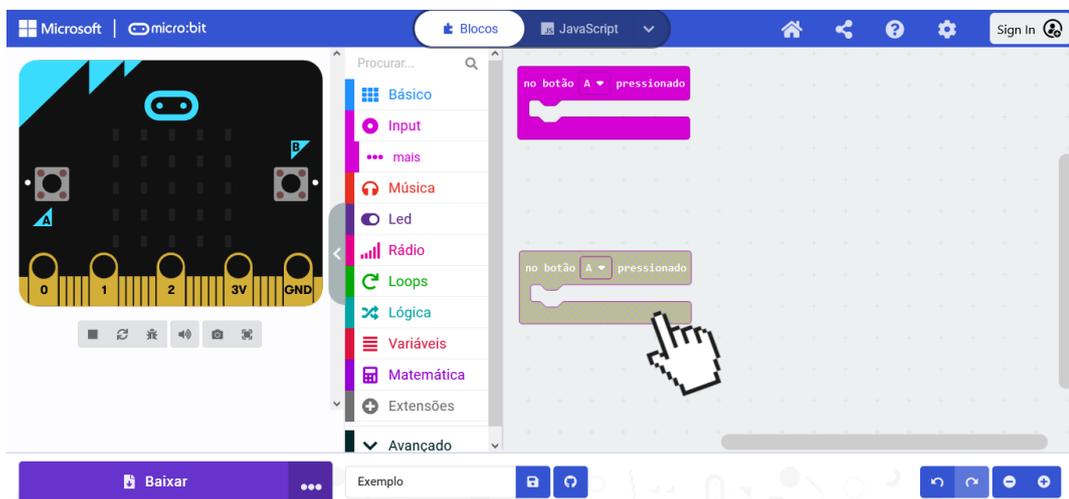


Figura 10.4: Montando um bloco de acionamento.



# Os sons das emoções

2. Substitua, no bloco que ficou na cor cinza, o botão **A** pelo botão **B**, como na Figura 10.5;

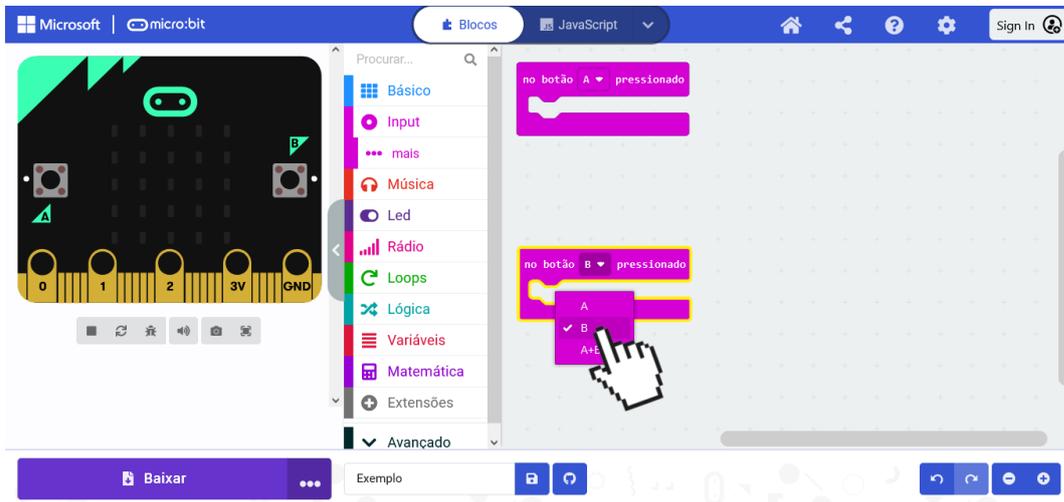


Figura 10.5: Alterando botões de comando.

3. Para acionar um botão e mostrar uma reação feliz e o outro botão uma reação triste, use o **mostrar ícone**, da aba **Básico**, veja as Figuras 10.6, 10.7, 10.8 e 10.9;

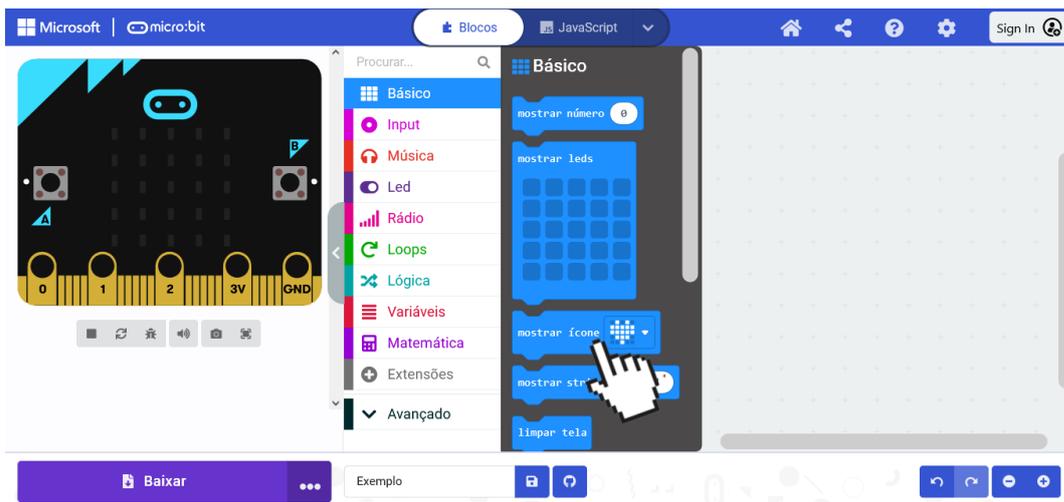


Figura 10.6: Selecionado o comando mostrar ícone.

# Os sons das emoções

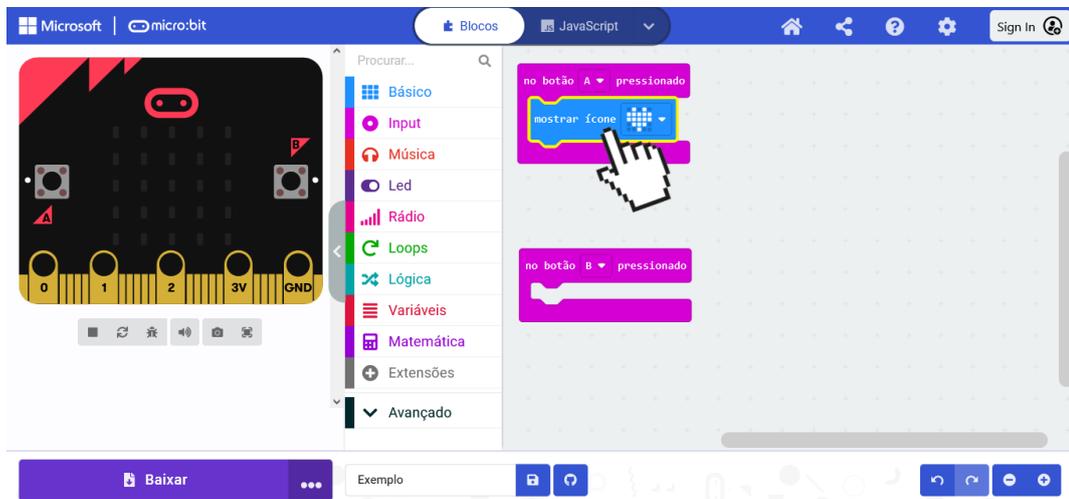


Figura 10.7: Montando um comando.

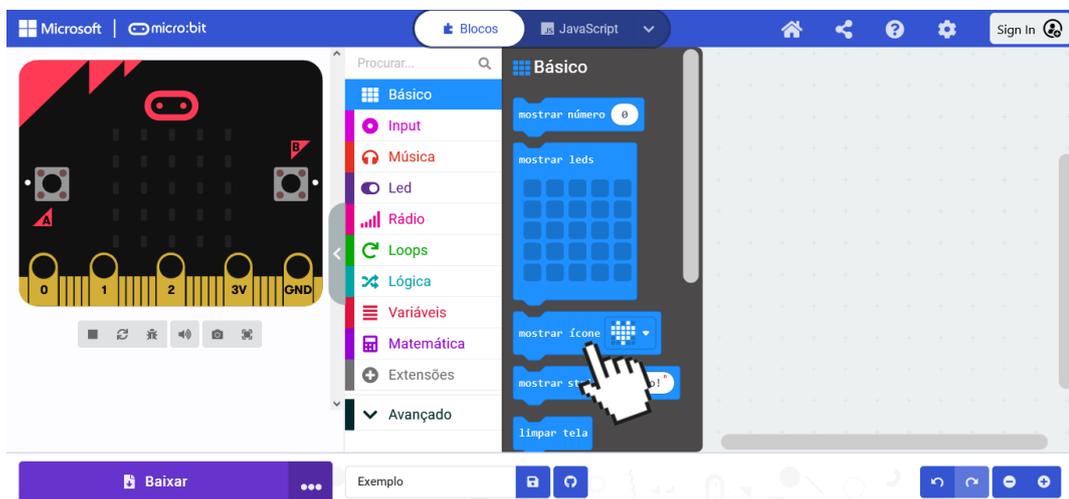


Figura 10.8: Selecionado o comando mostrar ícone.



# Os sons das emoções

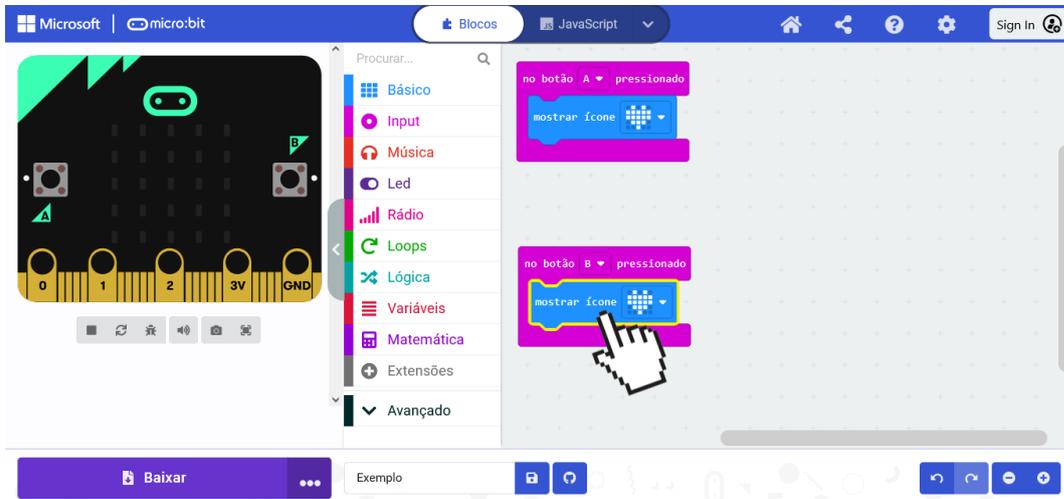


Figura 10.9: Montando um comando.

4. Altere o ícone do coração para o da carinha **feliz**, para o botão **A**, e carinha **triste** para o botão **B**, Figuras 10.10 e 10.11;

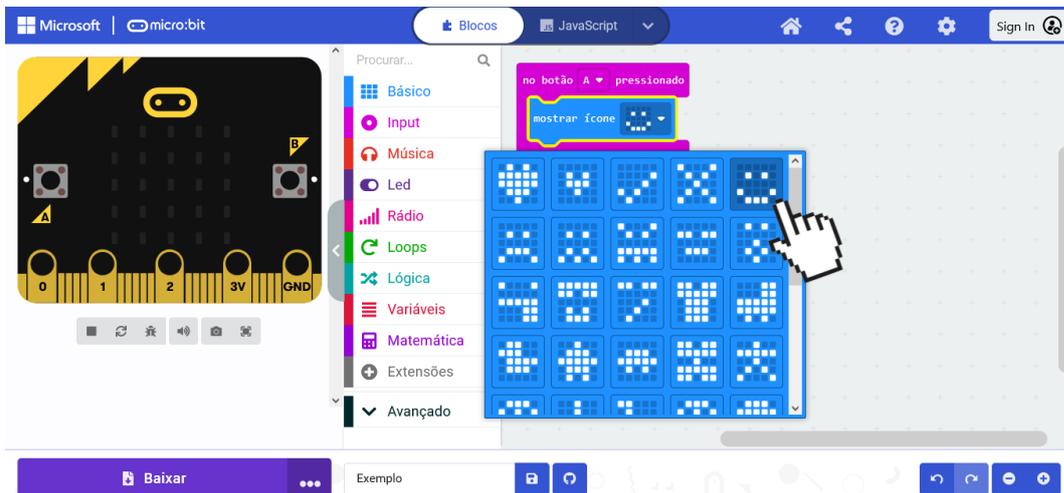


Figura 10.10: Alterando um ícone.

# Os sons das emoções

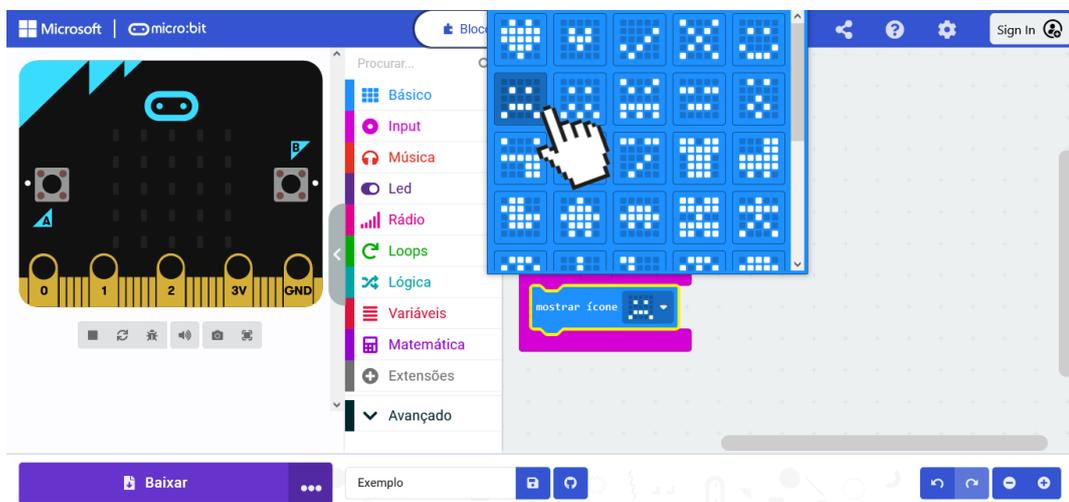


Figura 10.11: Alterando um ícone.

5. Para que os sons sejam tocados junto com as carinhas, na aba **Música**, selecione o comando **play sound risadinha until done**, Figura 10.12;

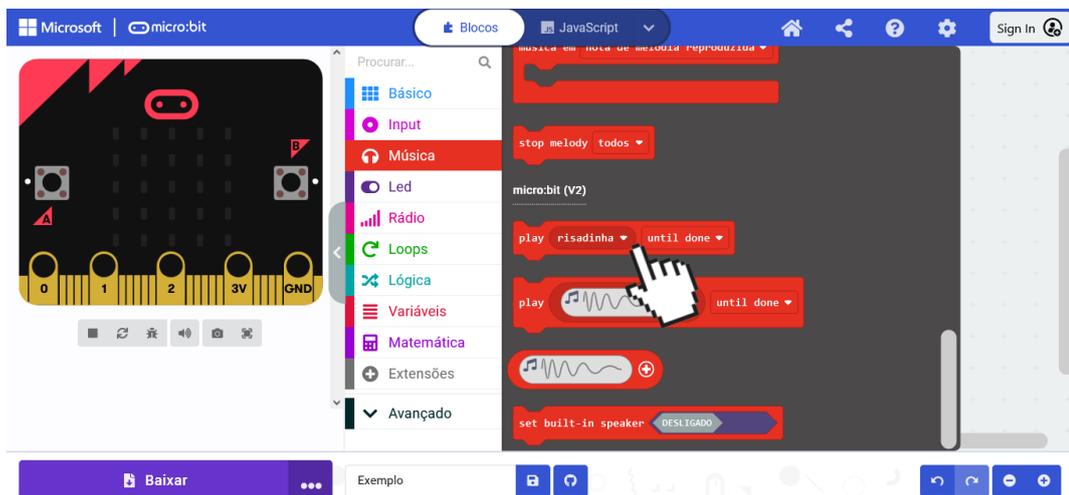


Figura 10.12: Selecionando um comando de música.



# Os sons das emoções

6. Coloque na sequência de cada carinha, um **play sound risadinha until done**, como nas Figuras 10.13, 10.14 e 10.15;

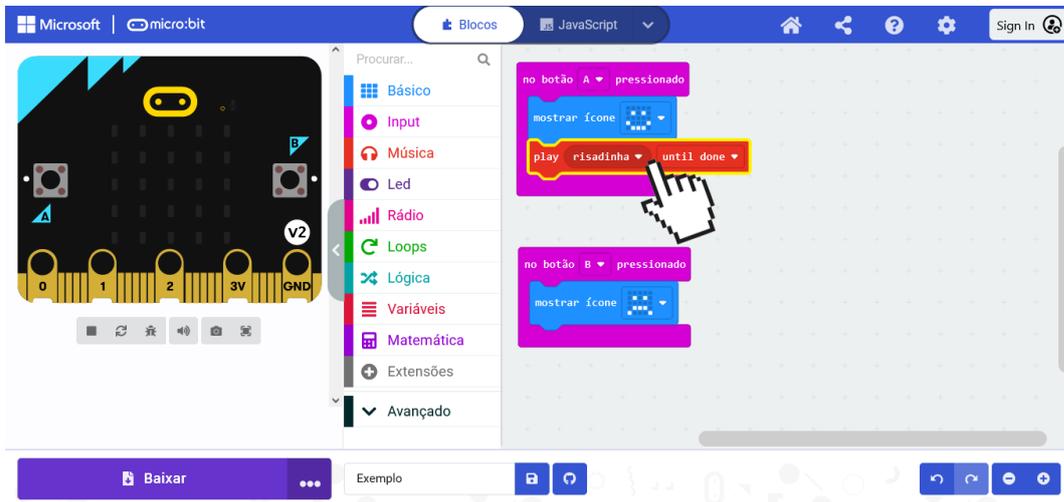


Figura 10.13: Montando um comando.

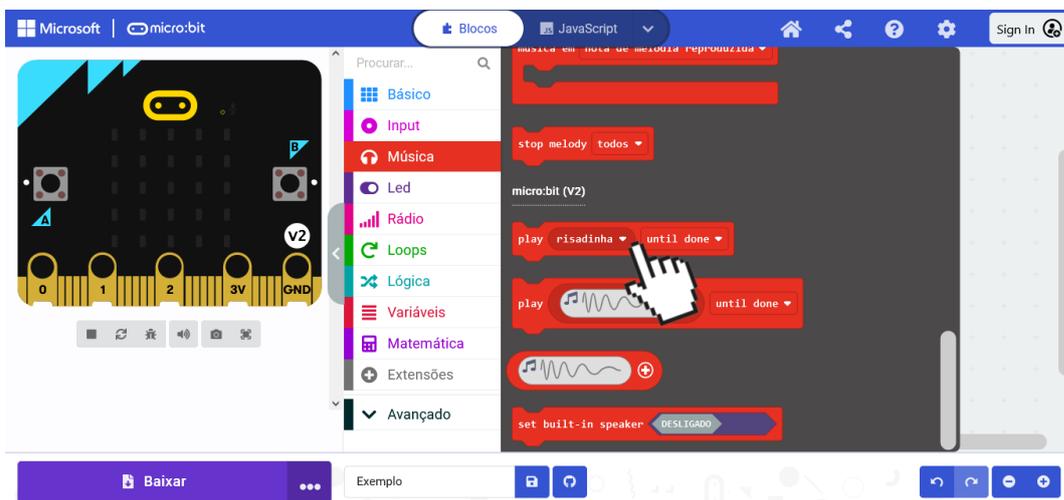


Figura 10.14: Selecionando um comando de música.

# Os sons das emoções

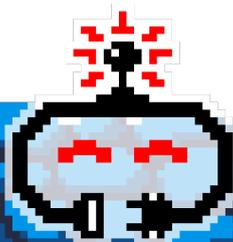


Figura 10.15: Montando um comando.

7. Para a reação **feliz**, mantenha o som da **risadinha** e para a **triste** seleccione o som **triste**, Figuras 10.16 e 10.17.



Figura 10.16: Selecionando uma música.



## Os sons das emoções

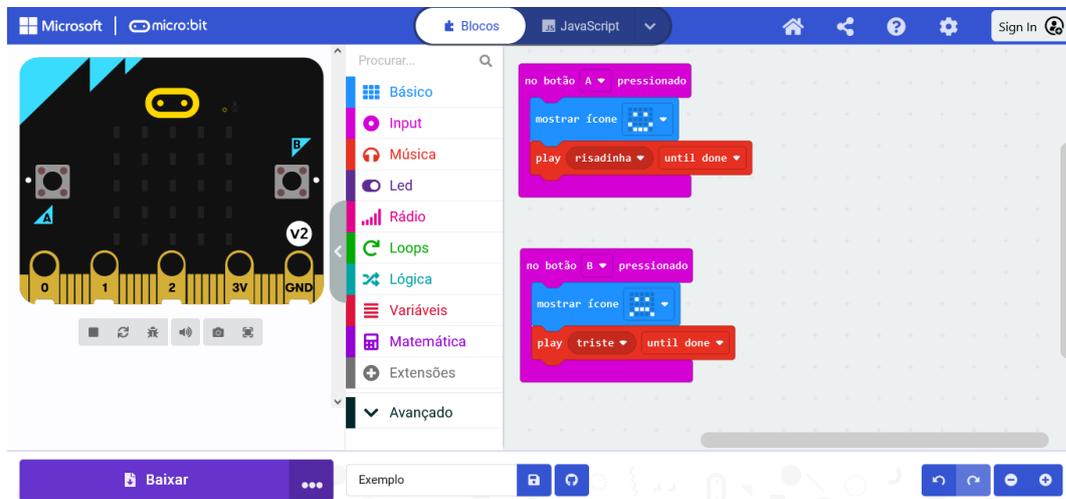


Figura 10.17: Alterando uma música.

**Baixe o programa para o micro:bit!**

Esses sons fazem sentido com as carinhas escolhidas? Agora que aprendemos a mostrar um ícone com um som, que tal falar sobre o que te deixa triste ou feliz?

## Os sons das emoções

### MICRO: DICA

Podemos acrescentar mais uma carinha ao nosso projeto. Para isso, basta repetir os passos anteriores e incluir a opção dos botões **A+B** pressionados. Uma sugestão é escolher o ícone **adormecido** e o som de **bocejo**.



[Clique aqui para acessar o código.](#)



Os sons das emoções



# Capítulo 11: Dado Digital

## Montando um dado

Entre as diversas atribuições que o micro:bit pode ter, uma das mais versáteis é o dado. O micro:bit pode substituir um dado simples de 6 lados de forma prática, sorteando um número aleatório apenas com o apertar de um botão.

Para isso, é preciso conhecer novos comandos, o **mostrar número**, da aba **Básico**, e o **escolher aleatório 0 até 10**, que está na aba **Matemática**. Essa nova aba contém as principais operações matemáticas, como algumas formas de contagem e o sorteio para nosso dado.

### mostrar número

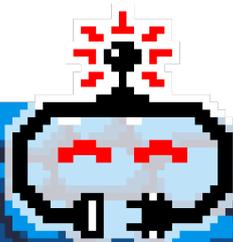
Este comando é usado para acender os LEDs que representam o número escolhido. Para alterar o número, clique sobre o zero e digite o número desejado.

mostrar número 0

### escolher aleatório 0 até 10

É a função usada para escolher números de forma aleatória, como em um sorteio. Para isso, define-se o intervalo dos números que serão sorteados, isto é, o número mínimo e o máximo da sequência.

escolher aleatório 0 até 10



# Dado Digital

## Passo a passo:

1. Comece selecionando o bloco **no botão A pressionado**, da aba **Input**, como nas Figuras 11.1 e 11.2;

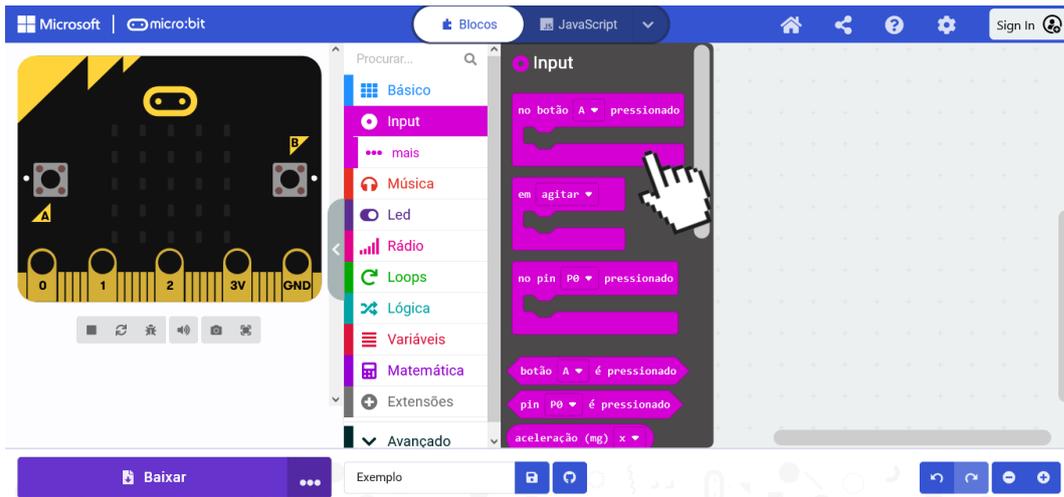


Figura 11.1: Selecionando um bloco de acionamento.

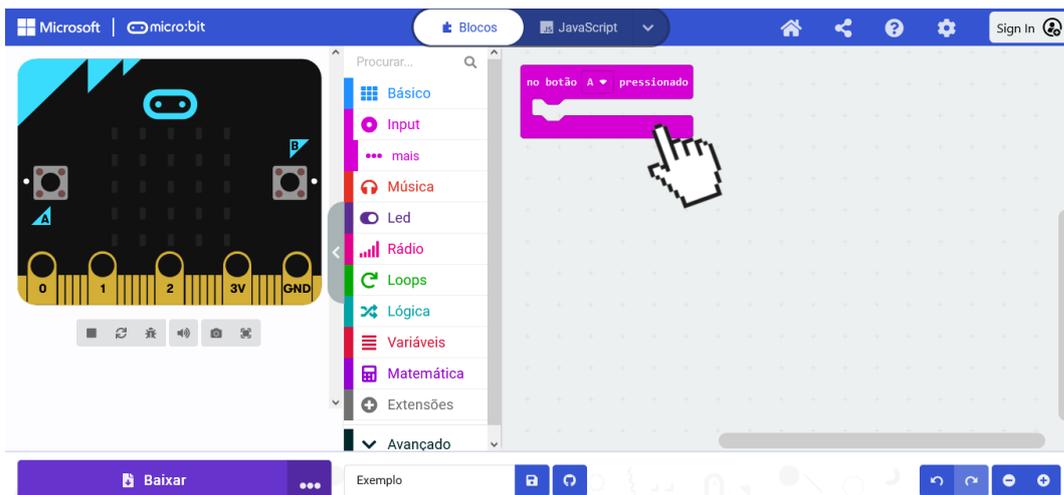


Figura 11.2: Acionamento selecionado.

# Dado Digital

2. Dentro do bloco, coloque a função **limpar tela**, encontrada na aba **Básico**, Figuras 11.3 e 11.4;

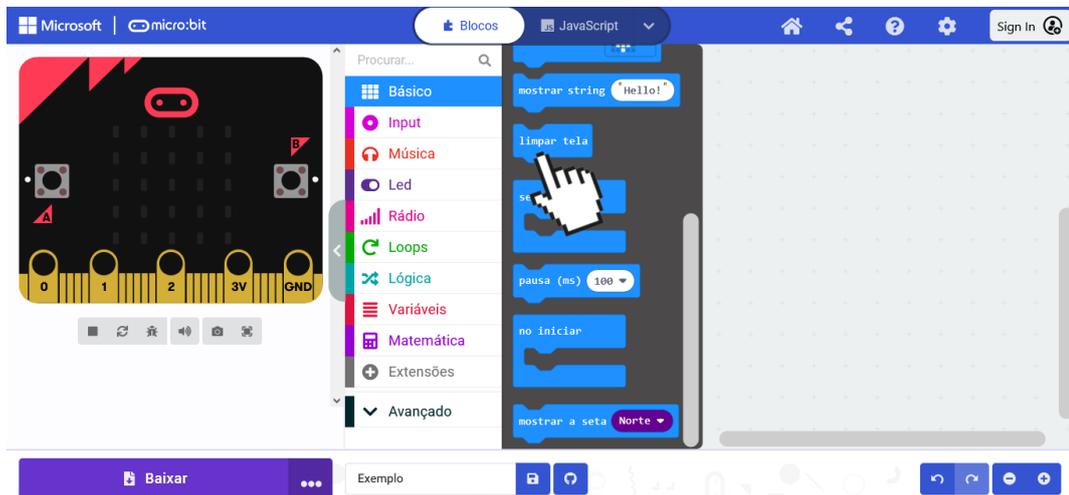


Figura 11.3: Selecionando um comando.



Figura 11.4: Montando um comando.



# Dado Digital

- Agora, coloque uma **pausa (ms)**, da aba **Básico**, e altere-a para 1 segundo, como nas Figuras 11.5, 11.6 e 11.7;

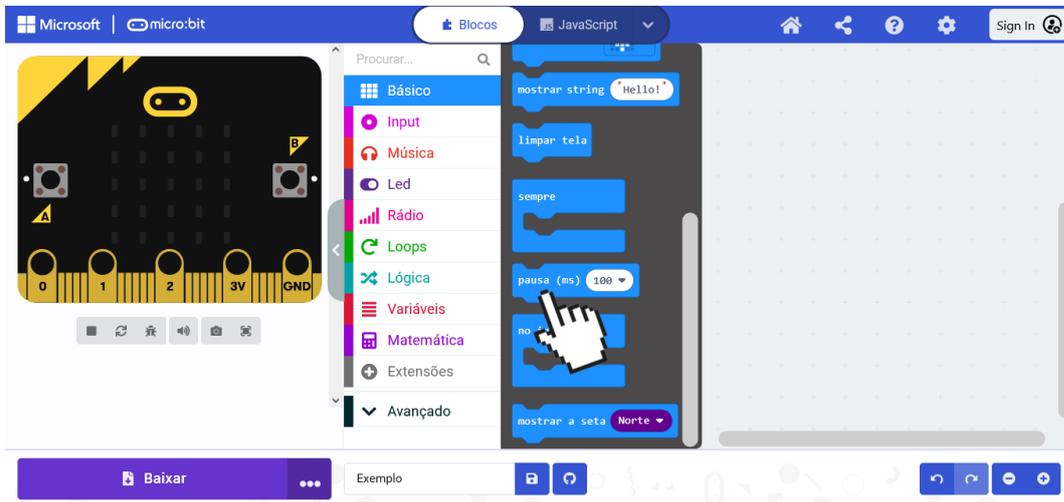


Figura 11.5: Selecionando uma pausa.

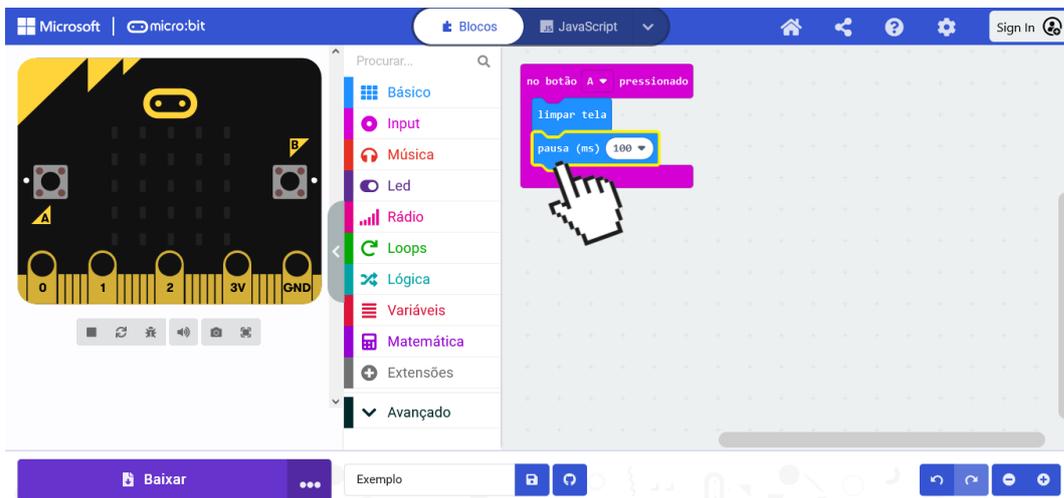


Figura 11.6: Montando o comando de pausa.

# Dado Digital

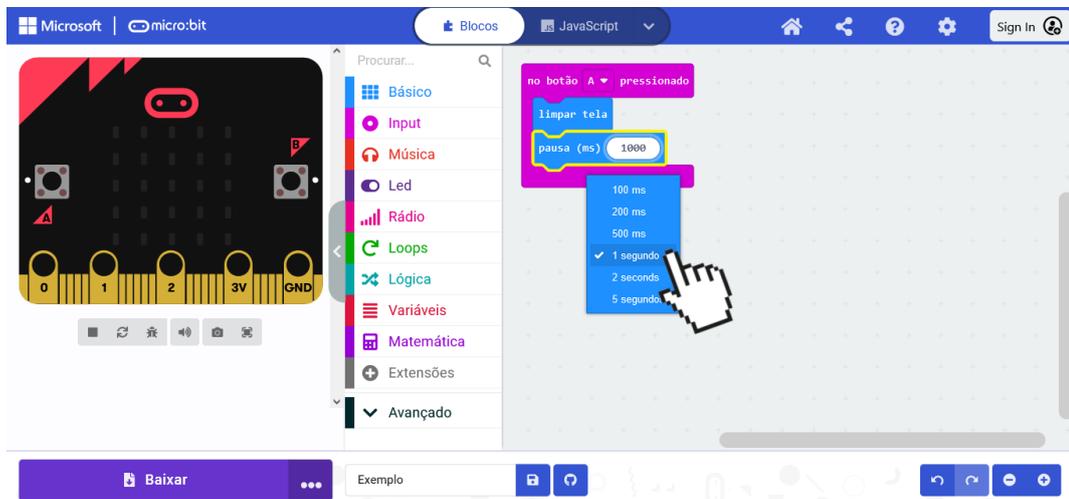


Figura 11.7: Alterando o tempo de uma pausa.

4. Vá até a aba **Básico** novamente e selecione o comando **mostrar número**, Figuras 11.8 e 11.9;



Figura 11.8: Selecionando o comando mostrar número.



# Dado Digital

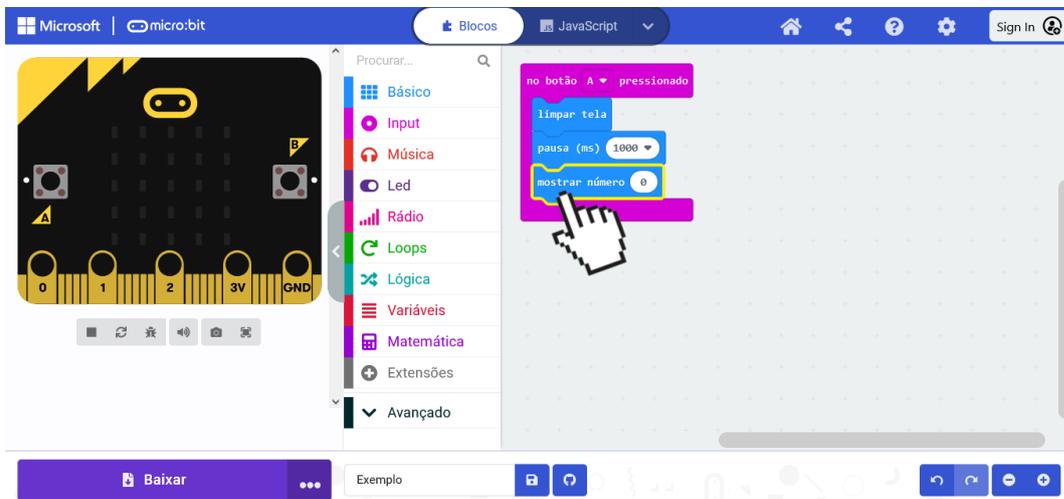


Figura 11.9: Montando um comando.

5. Pegue, na aba **Matemática**, o comando **escolher aleatório 0 até 10** e coloque-o sobre o zero de **mostrar número**, Figuras 11.10 e 11.11;

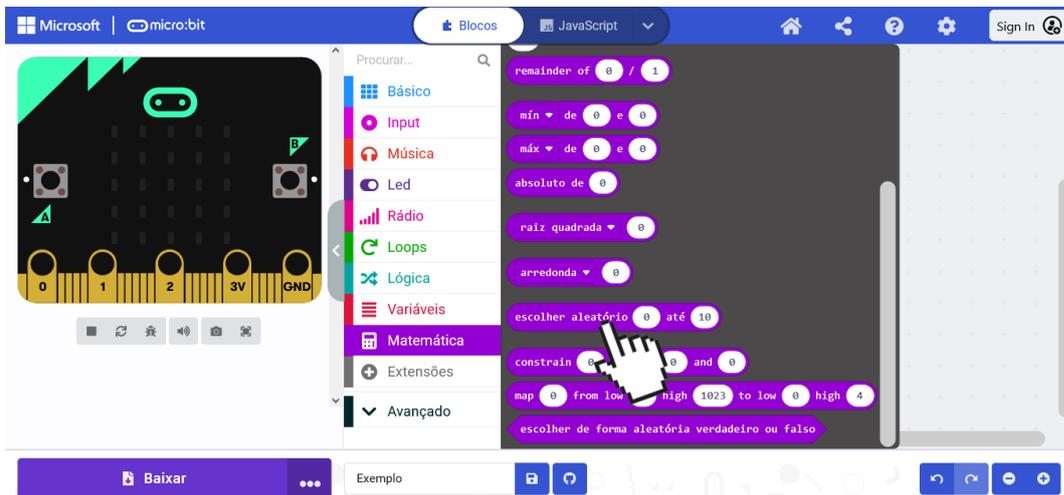


Figura 11.10: Selecionando um comando.

# Dado Digital

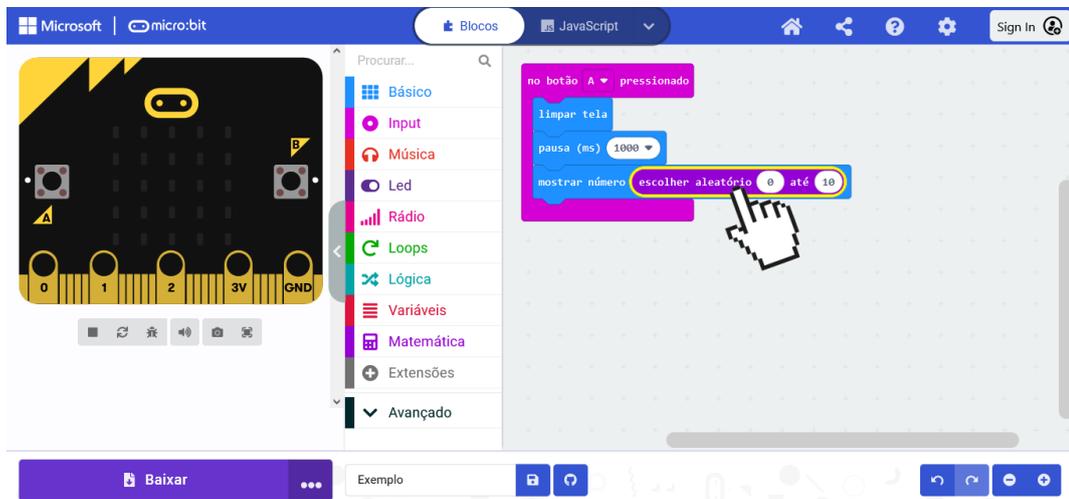


Figura 11.11: Montando um comando.

6. Por último, escolha os números que poderão ser sorteados. Quantos lados tem um dado simples? Ele tem **6** lados. Então, altere os números para **1** até **6**, como nas Figura 11.12 e 11.13.

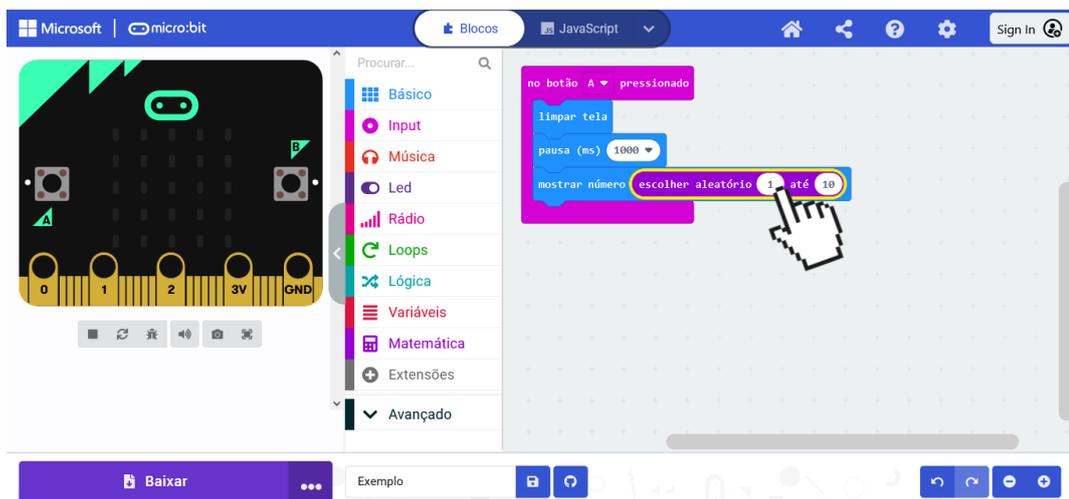


Figura 11.12: Alterando um número.



# Dado Digital

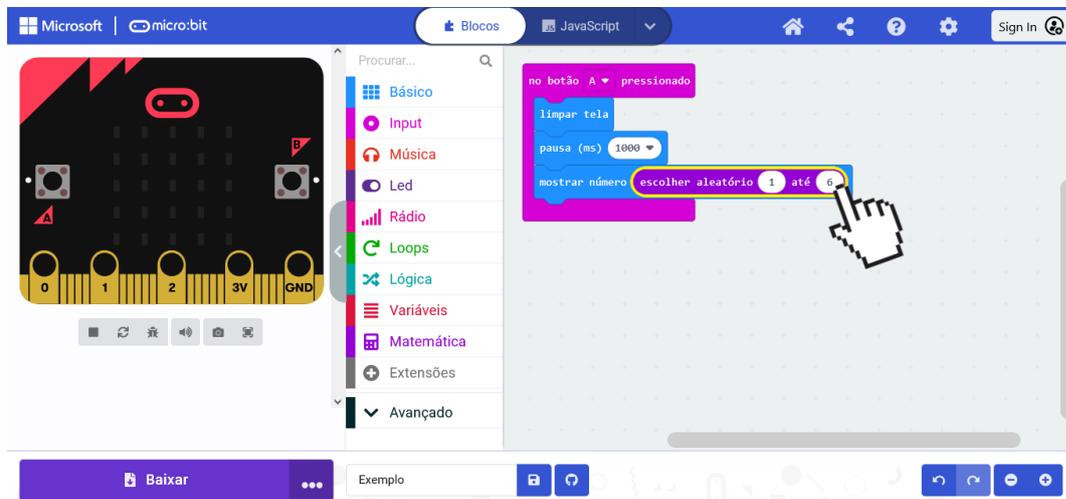


Figura 11.13: Alterando um número.

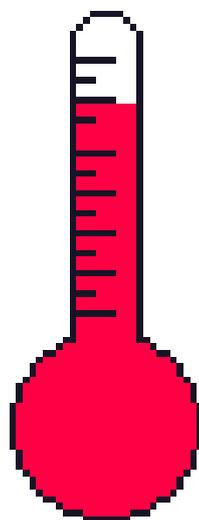
**Baixe o programa para o micro:bit para  
brincar com seu dado!**

Agora que temos um dado, podemos usá-lo em diversos jogos. Para um jogo de tabuleiro, podemos alterar os números para 0 até 6 e, quando o zero for sorteado, a pessoa perde a vez na jogada.

[Clique aqui para acessar o código.](#)



# Capítulo 12: Montando um termômetro



O que é um termômetro e o que ele faz? O termômetro é um instrumento que mede temperatura por meio de um sensor. O micro:bit também consegue fazer isso e pode ser programado usando o que já foi aprendido. Com a nova função **temperatura (°C)**, da aba **Input**, é possível fazer um termômetro.

## temperatura

É a função que usa o sensor de temperatura do micro:bit para medir a temperatura do ambiente. Essa temperatura é dada em graus na escala Celsius.

temperatura (°C)



# Montando um termômetro

## Passo a passo:

1. Selecione o bloco de acionamento **no botão A pressionado**, localizado na aba **Input**, como nas Figuras 12.1 e 12.2;

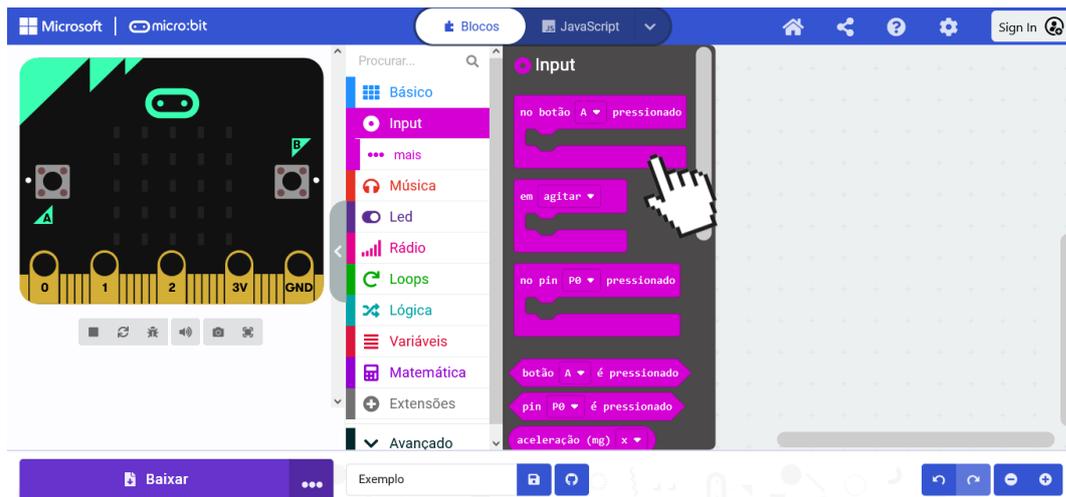


Figura 12.1: Selecionando um bloco de acionamento.

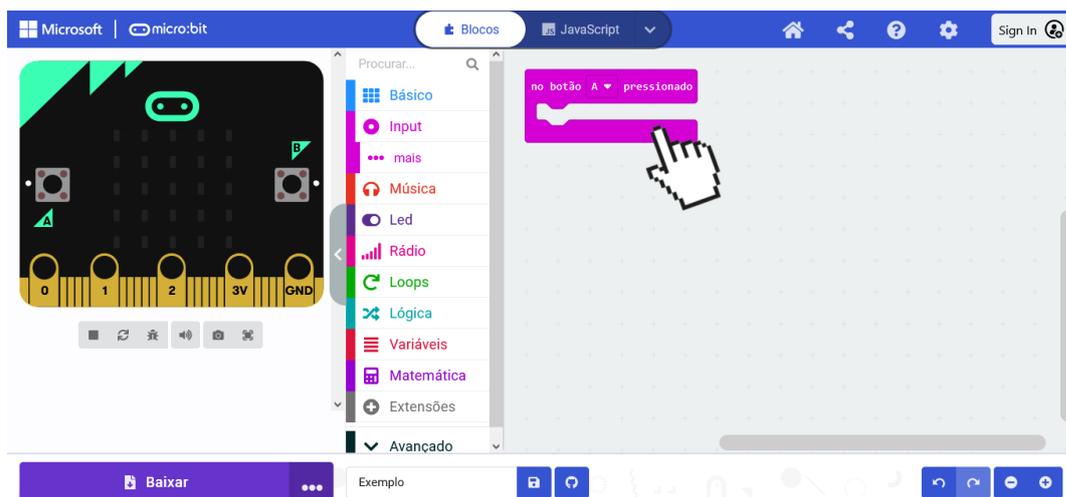


Figura 12.2: Montando um comando.

# Montando um termômetro

2. Após selecionado, pegue o comando **mostrar número**, localizado na aba **Básico**, Figura 12.3, e coloque-o dentro do bloco **no botão A pressionado**, Figura 12.4;

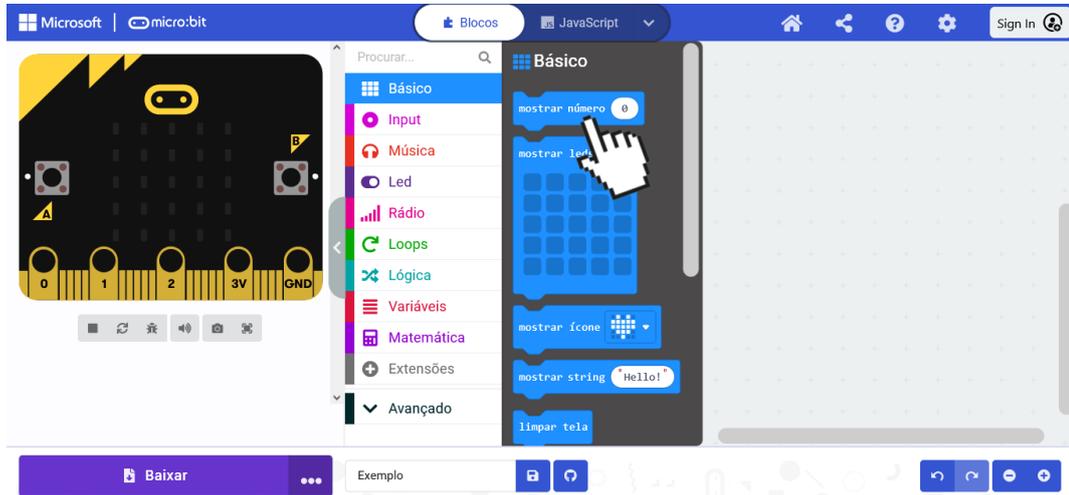


Figura 12.3: Selecionando um comando.

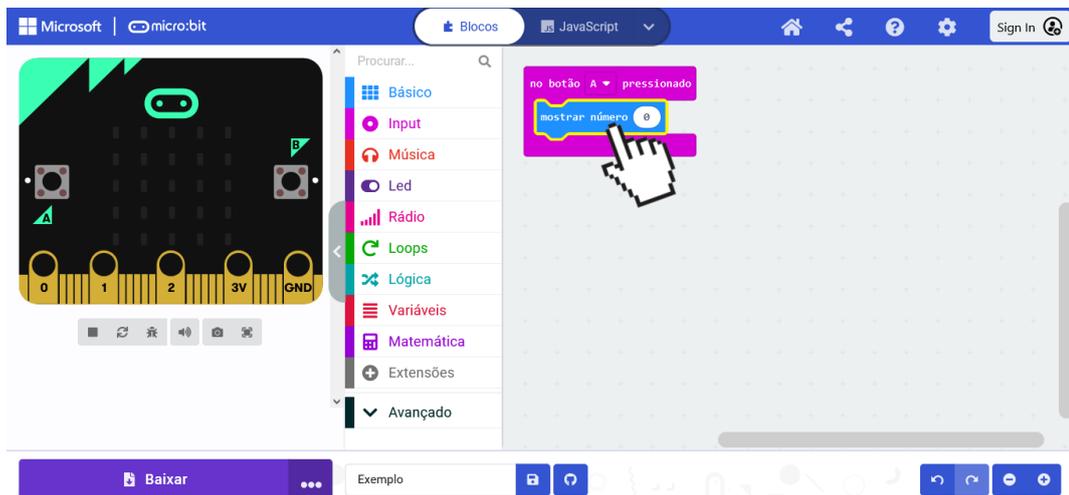
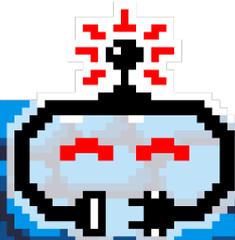


Figura 12.4: Montando um comando.



# Montando um termômetro

3. Antes de montar o comando para mostrar a temperatura, altere o número **0** pela sua idade e aperte o botão **A** no simulador para ver como os números aparecerão nos LEDs do micro:bit, Figura 12.5;

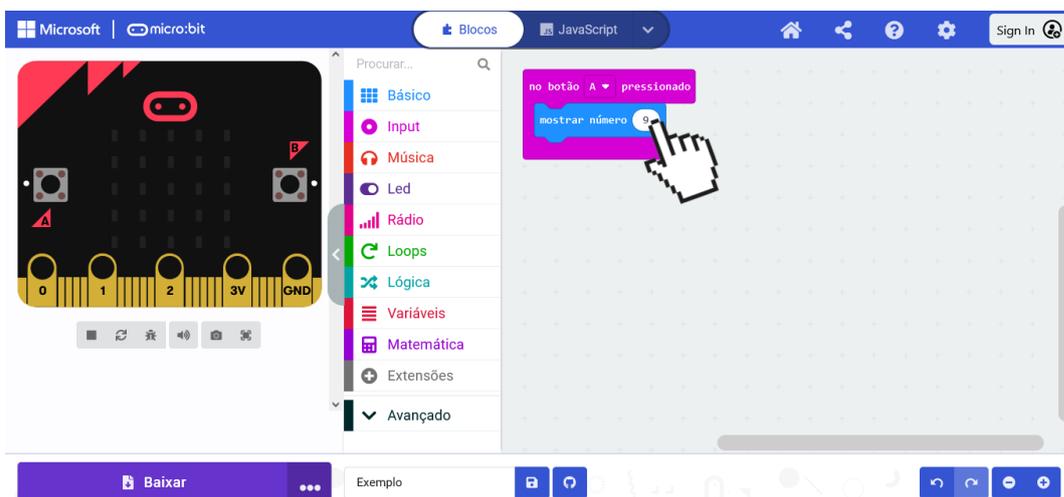


Figura 12.5: Montando um comando.

4. Para mostrar uma temperatura nos LEDs do micro:bit, clique sobre a aba **Input** e selecione a função **temperatura (°C)**, como na Figura 12.6;

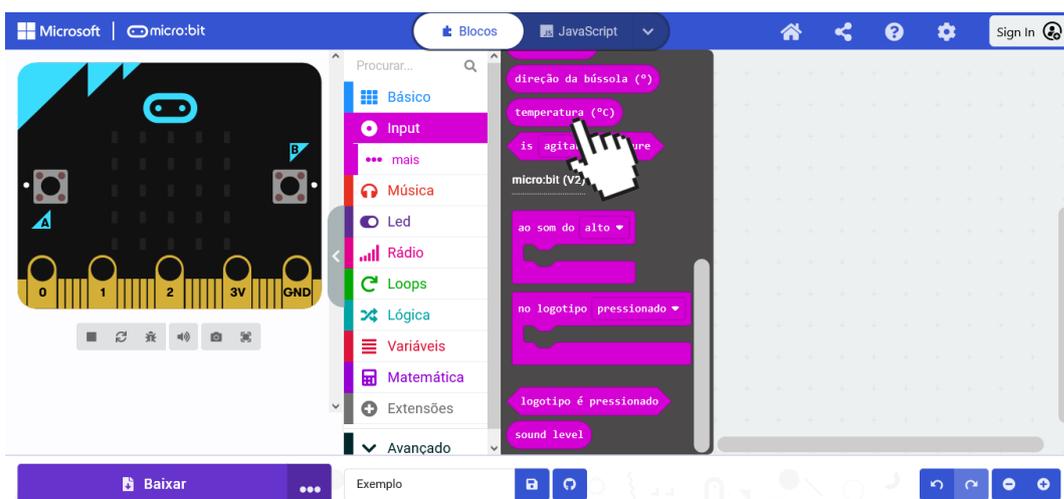


Figura 12.6: Selecionando um comando.

## Montando um termômetro

5. Para terminar, coloque a função **temperatura (°C)** no lugar do número, como mostra a Figura 12.7.

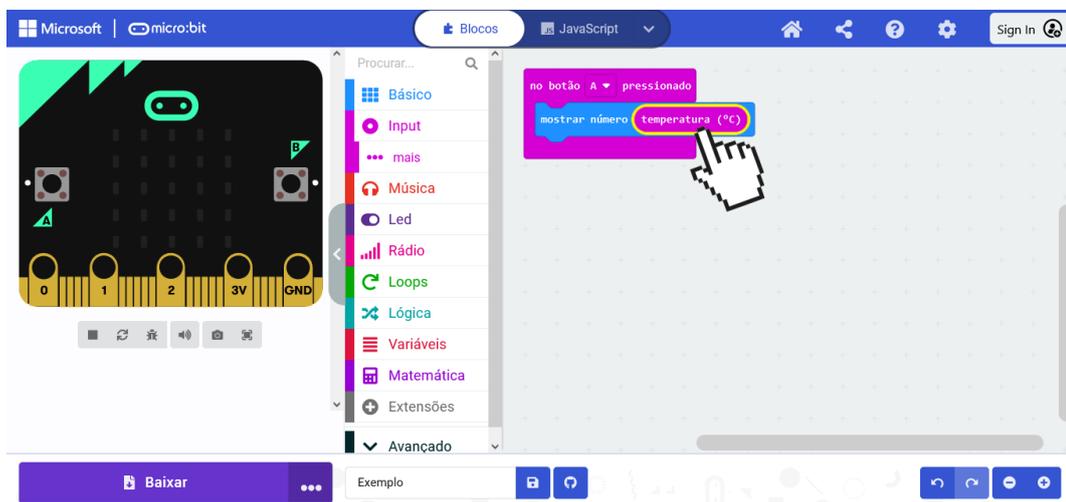


Figura 12.7: Montando um comando.

**Baixe o programa para o micro:bit para medir temperaturas!**

O termômetro funciona?  
A temperatura dos colegas é a mesma que a sua?  
Agora, que tal comparar a temperatura da sua mão com a das mãos dos colegas?  
Para isso localize o sensor do micro:bit e o cubra com um dedo.  
Ainda é a mesma temperatura que a do seu colega?

[Clique aqui para acessar o código.](#)

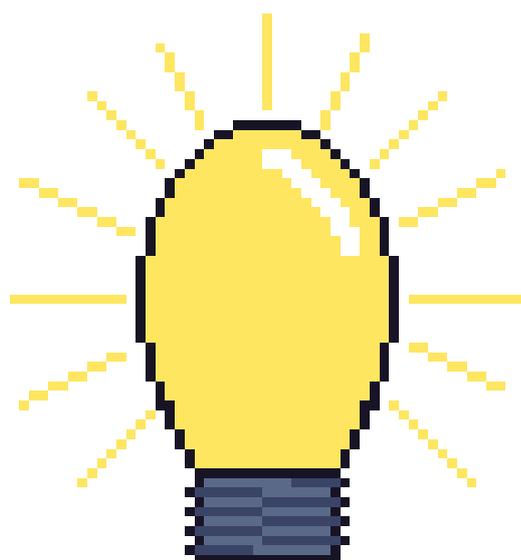




Montando um termômetro



# Capítulo 13: Conhecendo a lógica



A lógica é um conjunto de regras e conceitos que precisamos aplicar para criar códigos que serão interpretados e executados por um computador. Ou seja, é uma ferramenta que dá regras aos comandos do micro:bit. Para entender como isso funciona, serão utilizados os comandos de uma nova aba da biblioteca de funções, a **Lógica**.

Também utilizaremos um novo comando da aba **Matemática**, além de comandos das abas **Input** e **Básico**.



## Conhecendo a lógica

### Novos Blocos escolher de forma aleatória verdadeiro ou falso

#### escolher de forma aleatória verdadeiro ou falso

Assim como o **escolher aleatório 0 até 10** sorteia um número do intervalo definido, essa função vai escolher de forma aleatória os comandos verdadeiro ou falso.

#### se verdadeiro então



Essa função determina o que acontecerá se a condição para que o programa seja executado for satisfeita. Assim, se a condição for verdadeira o comando será executado.

#### se verdadeiro então, senão

É uma função que pode ser entendida da seguinte maneira: ao sair de casa olha-se para o céu; **se** o céu estiver nublado, **então** leva-se um guarda chuva, **senão** não leva nada.



# Conhecendo a lógica

## Passo a passo:

1. Na aba **Input**, selecione o bloco de comando **no botão A pressionado**, Figuras 13.1 e 13.2;

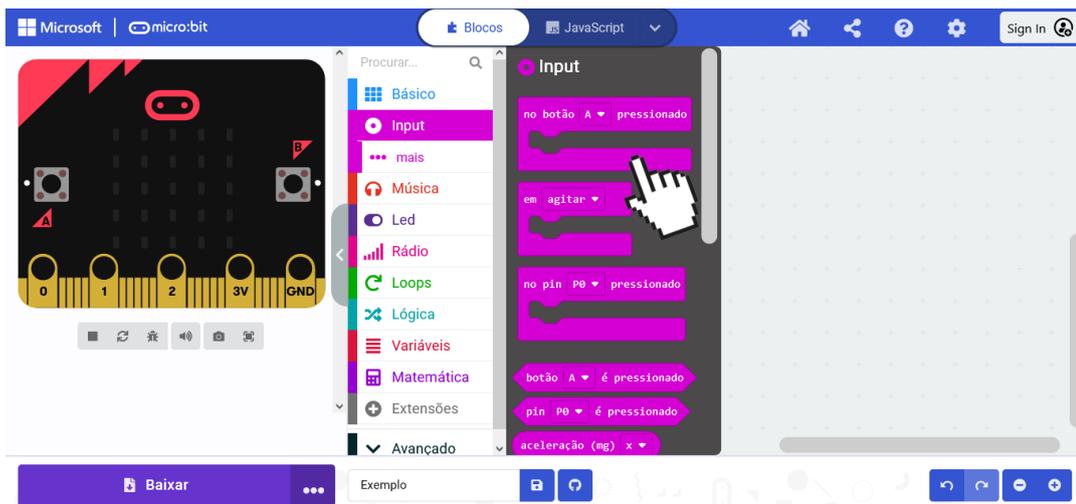
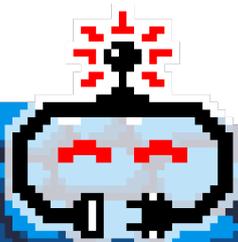


Figura 13.1: Selecionando um bloco de acionamento.



Figura 13.2: Montando um bloco de acionamento.



# Conhecendo a lógica

2. Na aba **Lógica**, selecione o bloco **se verdadeiro então**, Figura 13.3 ;

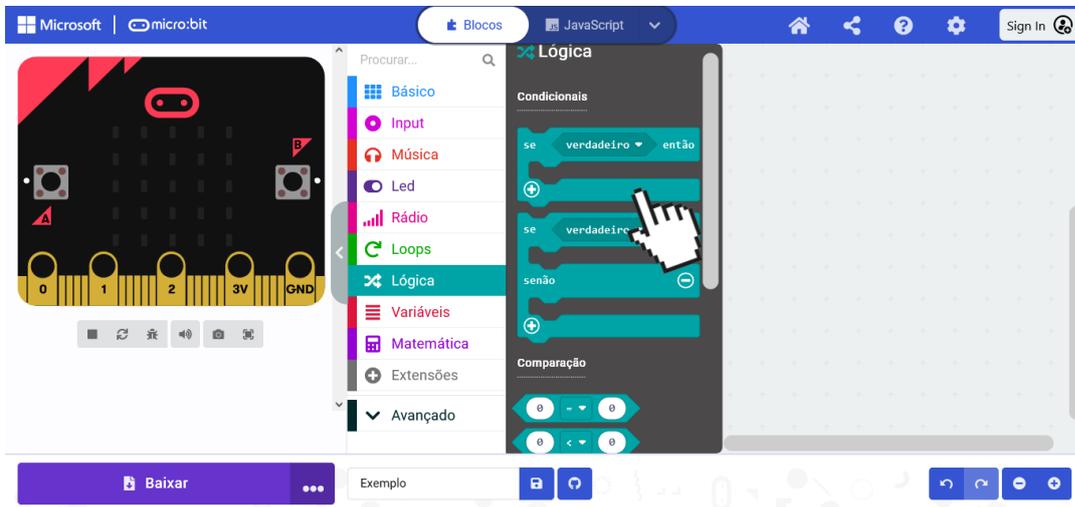


Figura 13.3: Selecionando um comando.

3. Coloque-o dentro do bloco **no botão A pressionado**, Figura 13.4;

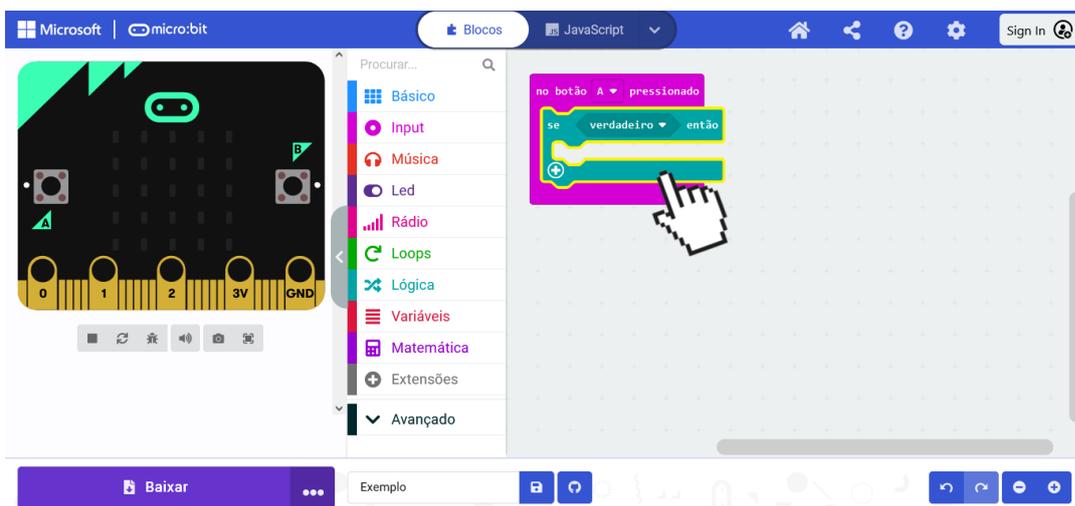


Figura 13.4: Montando um comando.

# Conhecendo a lógica

4. Na aba **Matemática**, selecione o bloco **escolher de forma aleatória verdadeiro ou falso** ou **falso**, Figura 13.5;

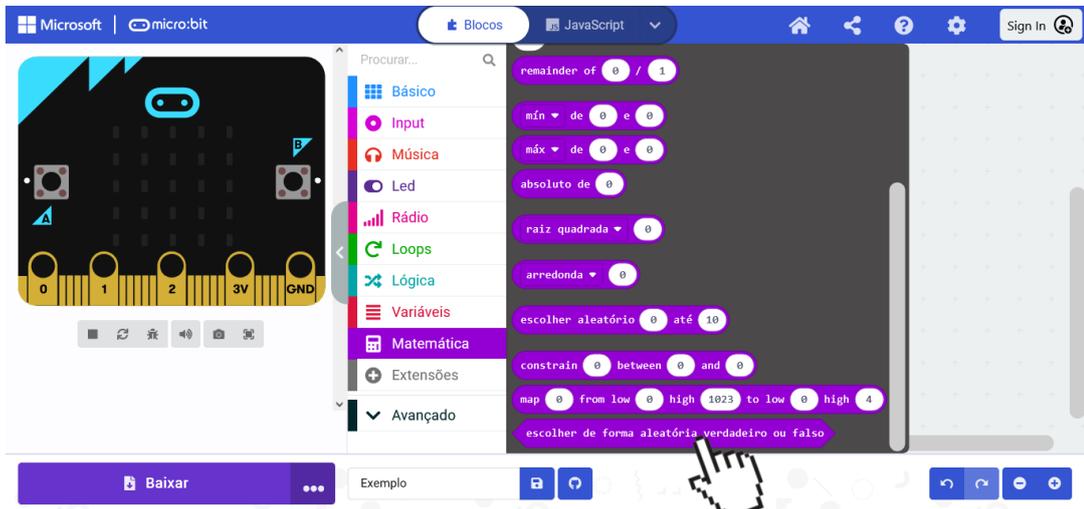


Figura 13.5: Selecionando um comando.

5. Coloque o bloco selecionado sobre a palavra **verdadeiro**. A montagem deve ficar como na Figura 13.6;

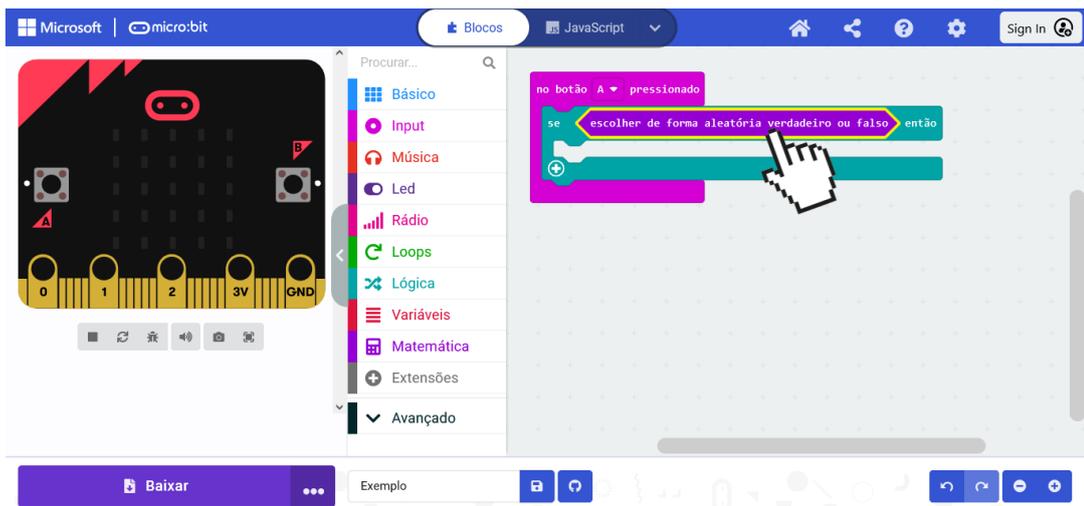


Figura 13.6: Montando um comando.



# Conhecendo a lógica

6. Na aba **Básico**, selecione o comando **mostrar ícone**, Figura 13.7;

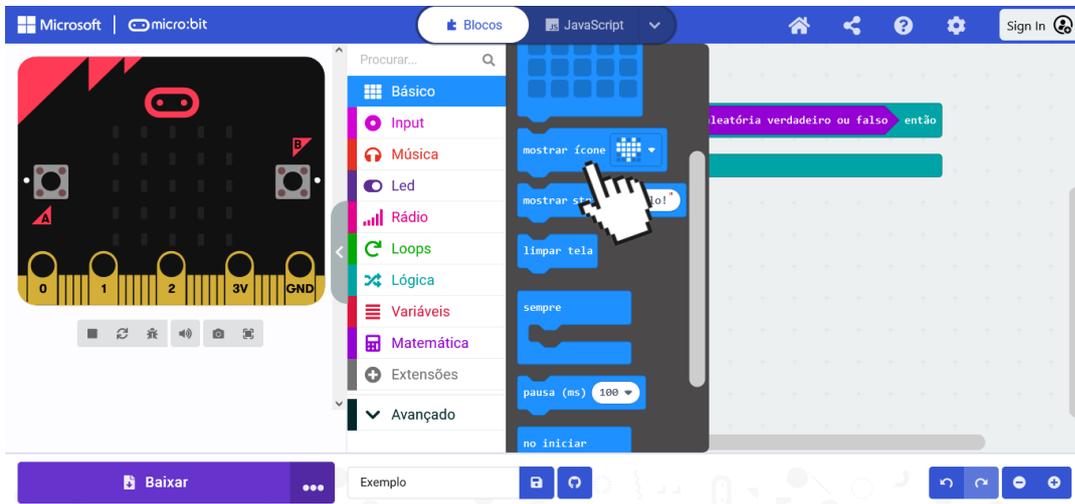


Figura 13.7: Selecionando um comando.

7. Coloque o comando **mostrar ícone** dentro do bloco de **Lógica** e escolha o ícone desejado, Figuras 13.8 e 13.9;

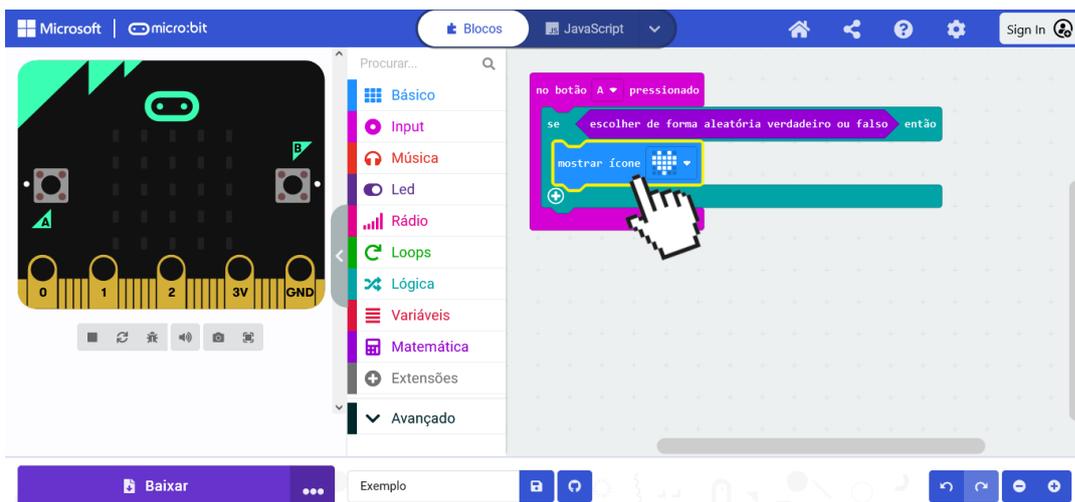


Figura 13.8: Montando um comando de lógica.

# Conhecendo a lógica

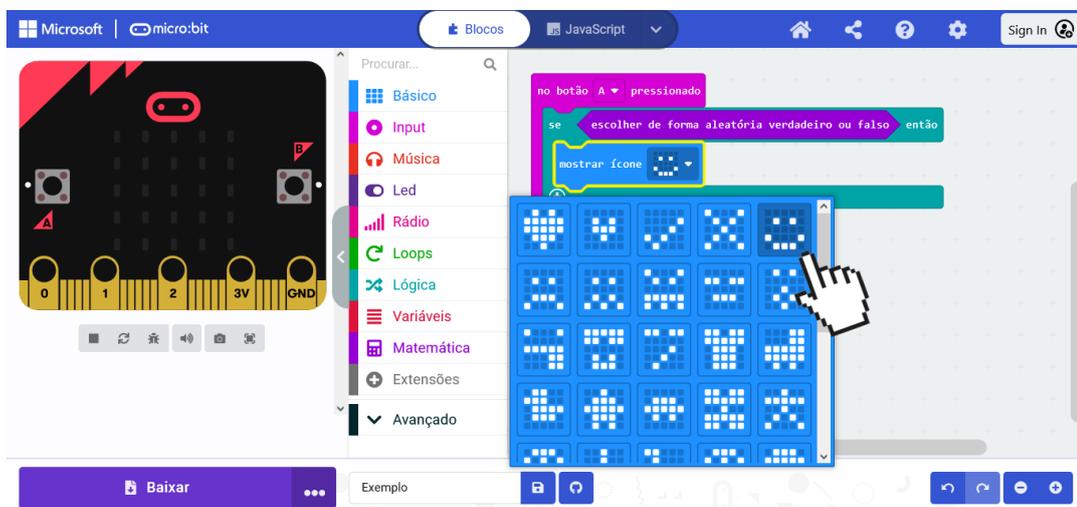


Figura 13.9: Alterando um ícone.

- Na aba **Básico**, selecione o comando **limpar tela**, Figura 13.10;

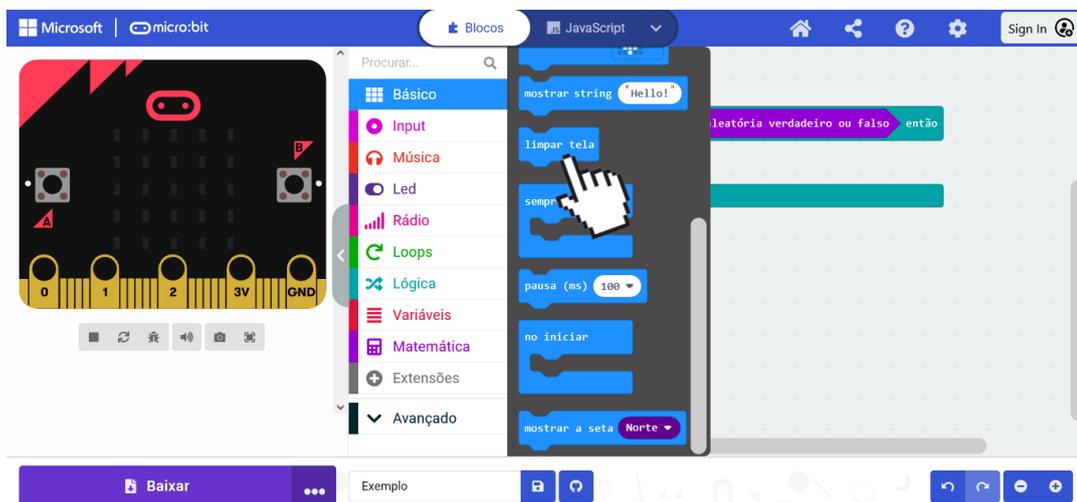
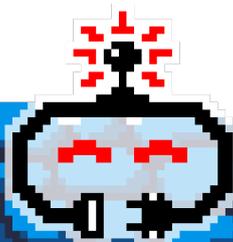


Figura 13.10: Selecionando um comando.



## Conhecendo a lógica

9. Insira o comando **limpar tela**, logo abaixo do **mostrar ícone**, dentro do bloco de **Lógica**, como na Figura 13.11.



Figura 13.11: Montando um comando.

**Baixe o programa para o micro:bit e veja como a lógica funciona.**

**Conseguiu entender o que acontece? Quando o micro:bit escolhe verdadeiro, a imagem é mostrada. Quando escolhe falso, nada acontece.**

[clique aqui para acessar o código](#)



# Capítulo 14: Pedra, papel e tesoura



O micro:bit possui diversos recursos. Dentre eles, é possível criar variáveis e montar diferentes tipos de situações usando a lógica. Neste capítulo, utilizando os conhecimentos dos capítulos anteriores, criaremos uma nova forma de jogar pedra, papel e tesoura usando um novo bloco de acionamento, **em agitar**, da aba **Input**.

## em agitar:

Quando usado, faz com que o micro:bit execute os comandos selecionados todas as vezes em que a placa é agitada.



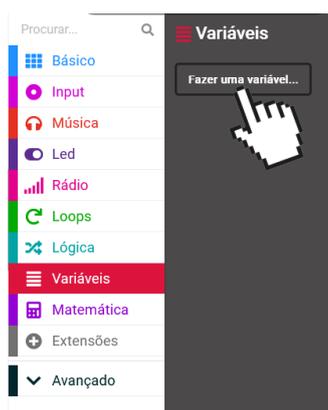
# Pedra, papel e tesoura

## Variáveis

Antes de fazer o jogo, é preciso entender o que são variáveis, na programação, e como criá-las para montar os comandos do código.

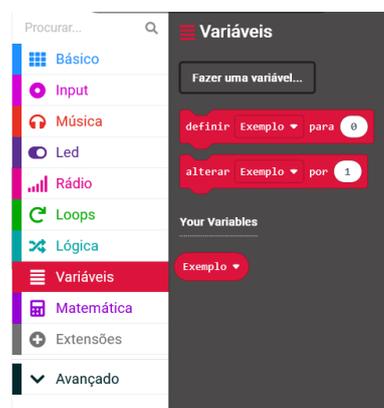
Uma variável é um espaço na memória de um computador que armazena um valor. Pode-se compará-la a uma caixinha que guarda números. Esses números podem ser inicialmente definidos e alterados a partir de determinados comandos. Por exemplo, é possível definir uma variável para o número zero e, apertando o botão A, alterá-la por 2. Então, cada vez que o botão A for pressionado, o número 2 será somado ao resultado anterior.

Para criar uma variável no micro:bit, use a aba **Variáveis** da biblioteca de funções. Nesta aba pode-se criar, alterar ou substituir variáveis. Para isso, basta clicar em **Fazer uma variável** e escolher um nome para esta variável.



**Importante:** no nome da variável não pode colocar espaço entre letras e palavras, usar caracteres especiais ou começar com um número.

Depois de criada, pode-se substituir uma variável por algum valor numérico, alterá-lo em determinado momento e utilizá-lo em comandos.



# Pedra, papel e tesoura

O Jogo:

## Passo a passo:

1. Comece criando a variável que será. Na aba **Variáveis**, clique em **Fazer uma variável...**, como na Figura 14.1;

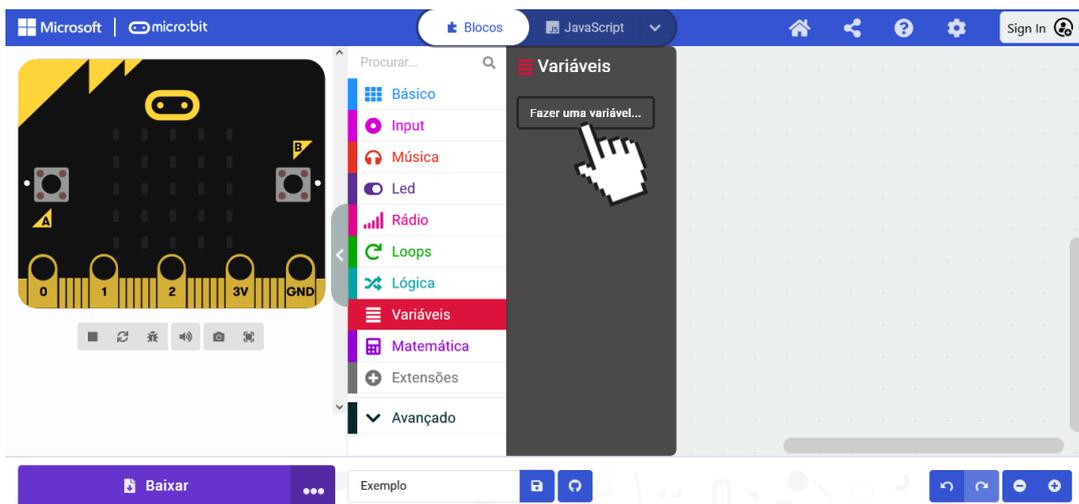


Figura 14.1: Fazendo uma variável.

2. Dê um nome para a variável. Aqui chamada de **sorteio**, Figura 14.2;

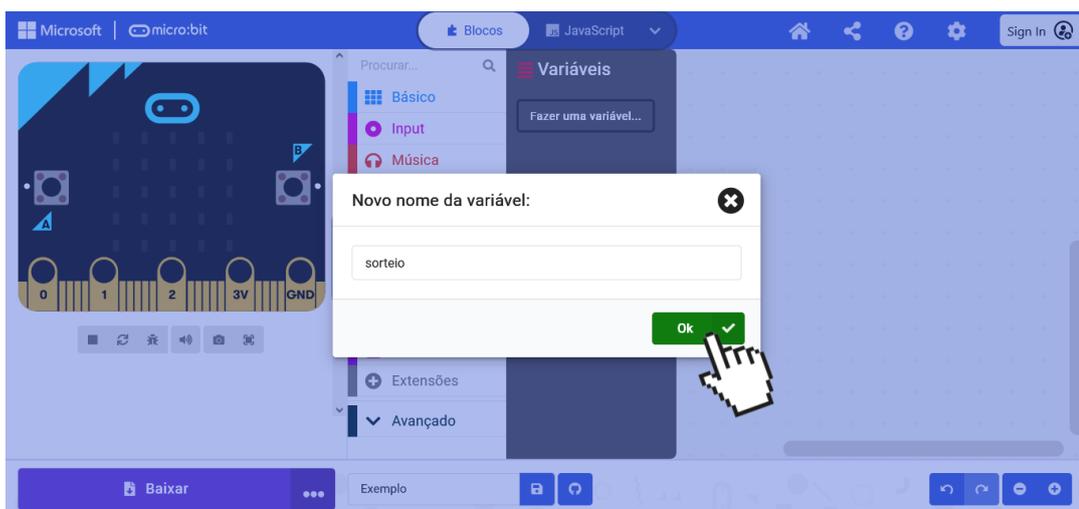


Figura 14.2: Dando um nome a uma variável.



# Pedra, papel e tesoura

3. Na aba **Input**, selecione o bloco de acionamento **em agitar**, Figuras 14.3 e 14.4;

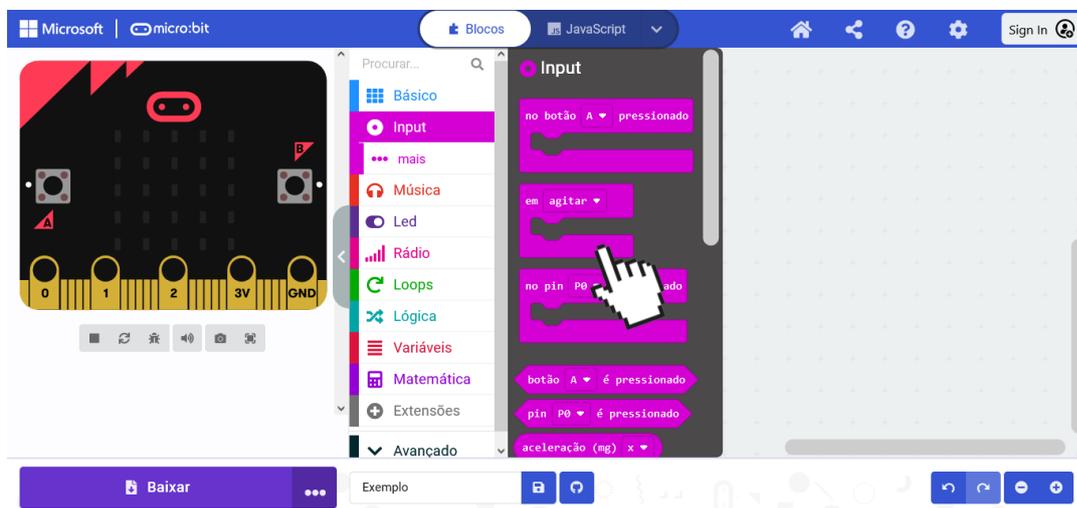


Figura 14.3: Selecionando um bloco de acionamento.

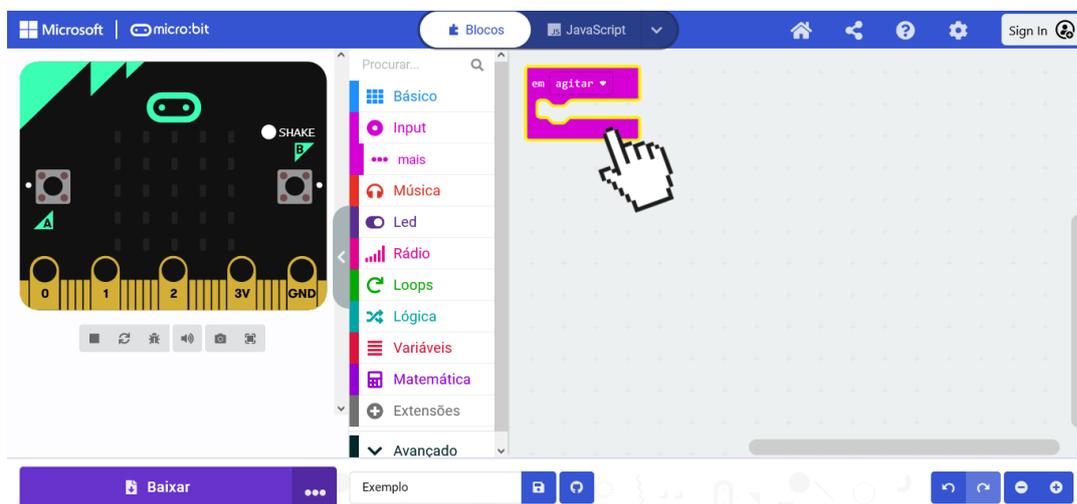


Figura 14.4: Montando um comando.

# Pedra, papel e tesoura

4. Para mostrar que o micro:bit foi realmente agitado, crie um padrão. Para isso, na aba **Básico**, selecione **mostrar ícone**, e arraste-o para **em agitar**, Figuras 14.5 e 14.6;

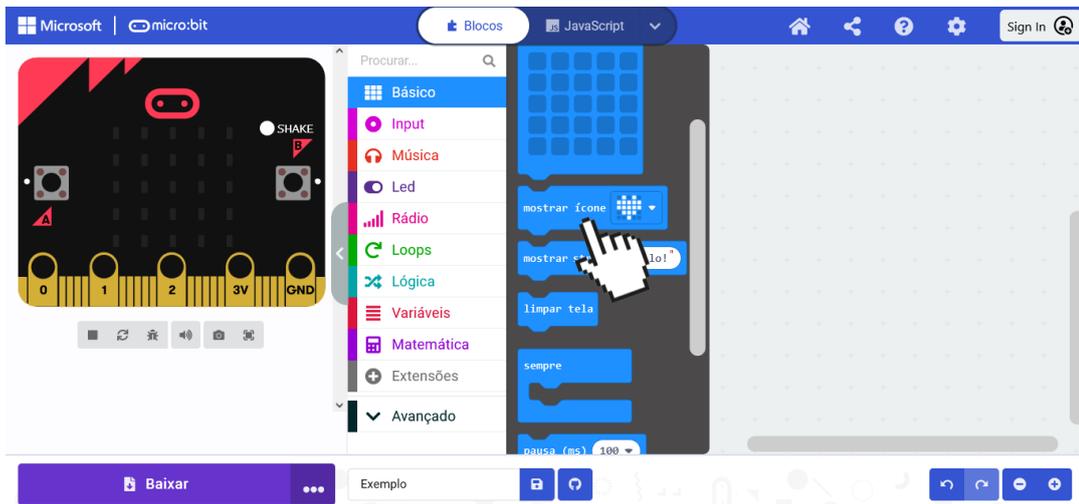


Figura 14.5: Selecionando um comando.

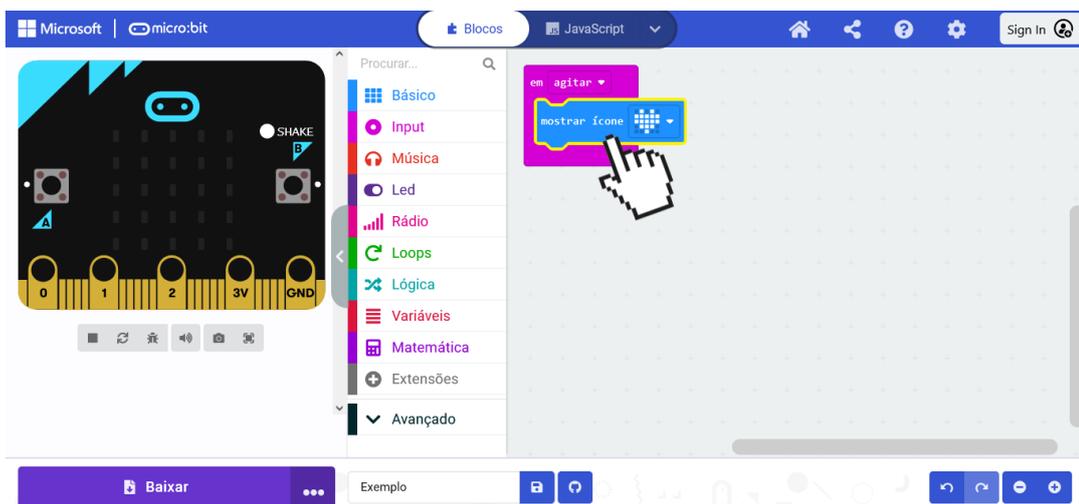
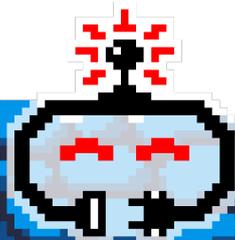


Figura 14.6: Montando um comando.



# Pedra, papel e tesoura

5. Selecione o ícone **diamante**, Figura 14.7;

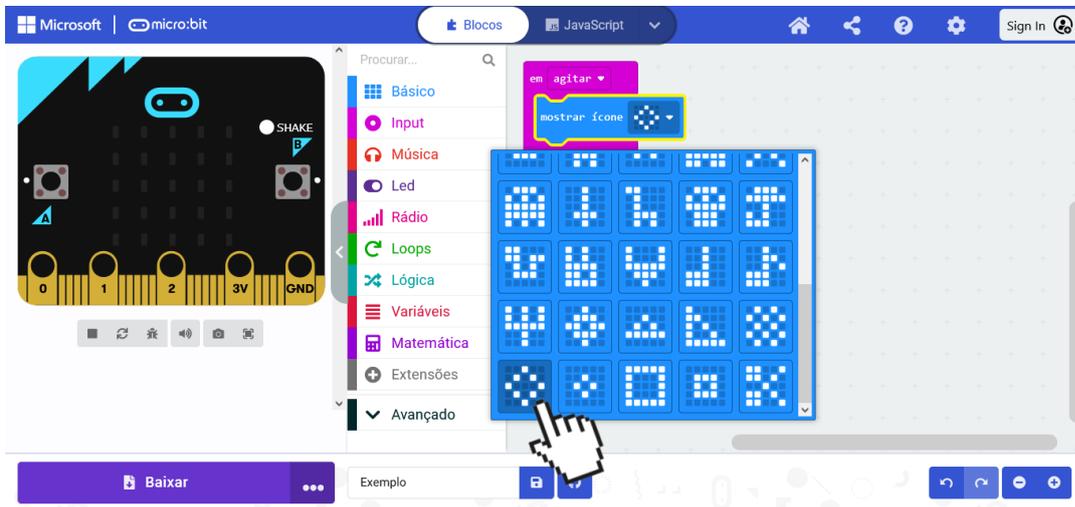


Figura 14.7: Alterando um ícone.

6. Na aba **Básico**, selecione novamente o comando **mostrar ícone**, Figura 14.8;

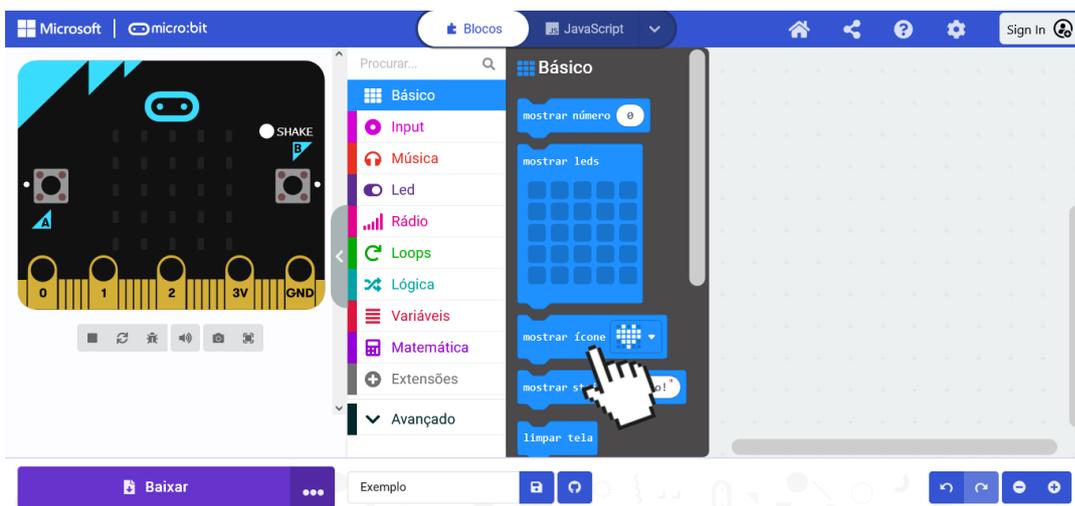


Figura 14.8: Selecionando um comando.

# Pedra, papel e tesoura

7. Coloque o último comando selecionado dentro do bloco **em agitar**, logo abaixo do outro **mostrar ícone**, Figura 14.9;

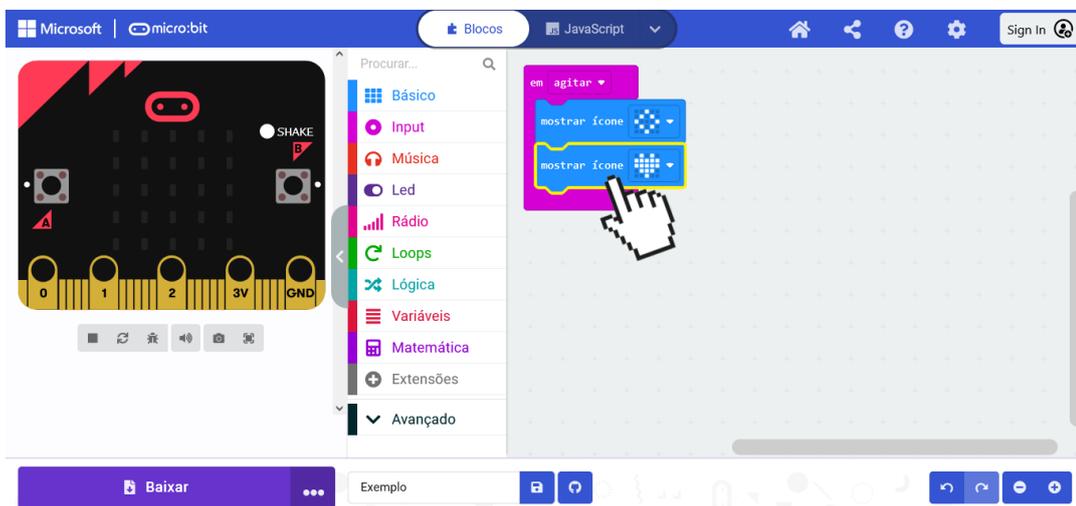


Figura 14.9: Montando um comando.

8. Selecione o ícone **diamante pequeno**, Figura 14.10;

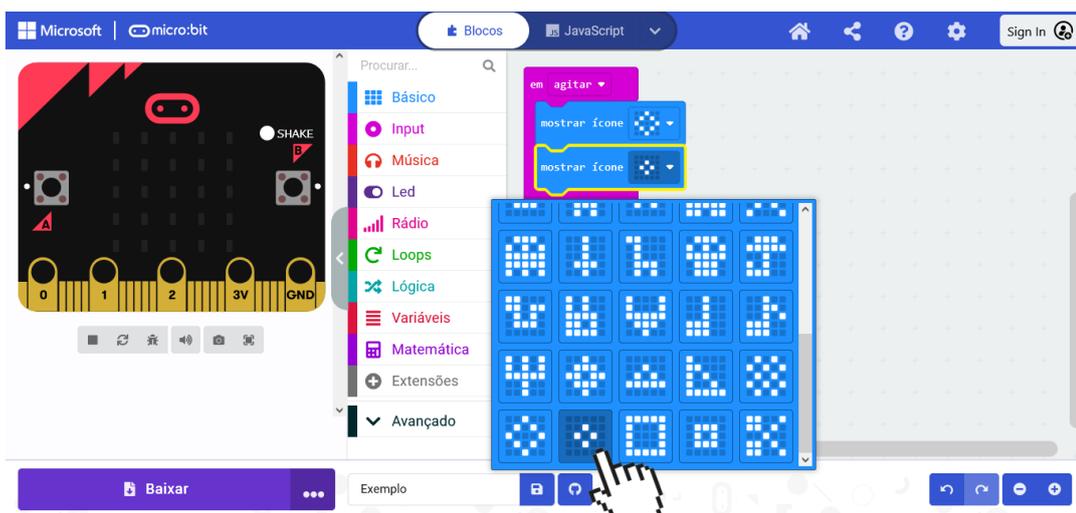
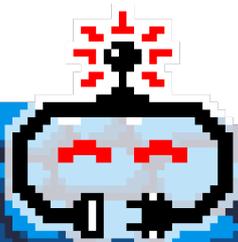


Figura 14.10: Alterando um ícone.



# Pedra, papel e tesoura

9. Na aba **Básico**, selecione o comando **mostrar leds**, Figura 14.11;

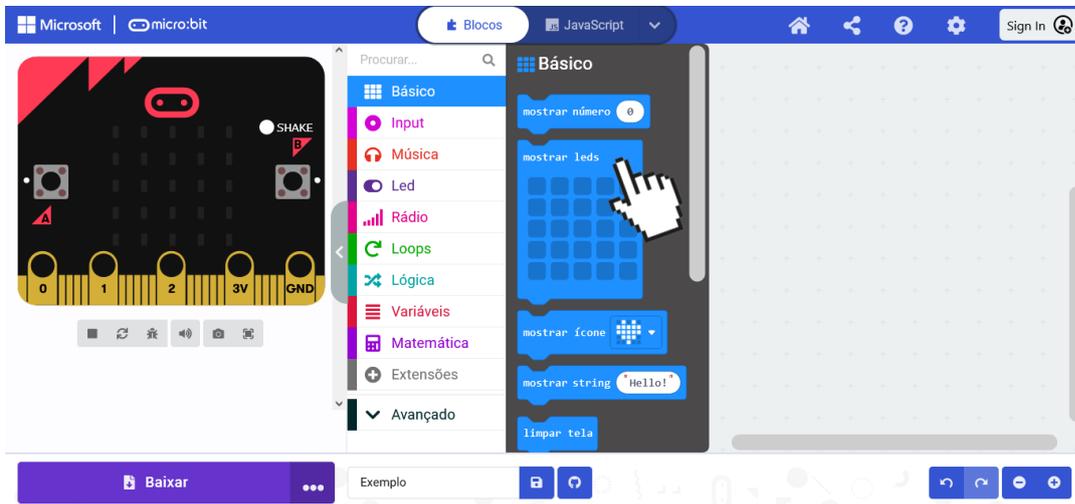


Figura 14.11: Selecionando um comando.

10. Coloque o **mostrar leds** embaixo do último **mostrar ícone**, Figura 14.12;

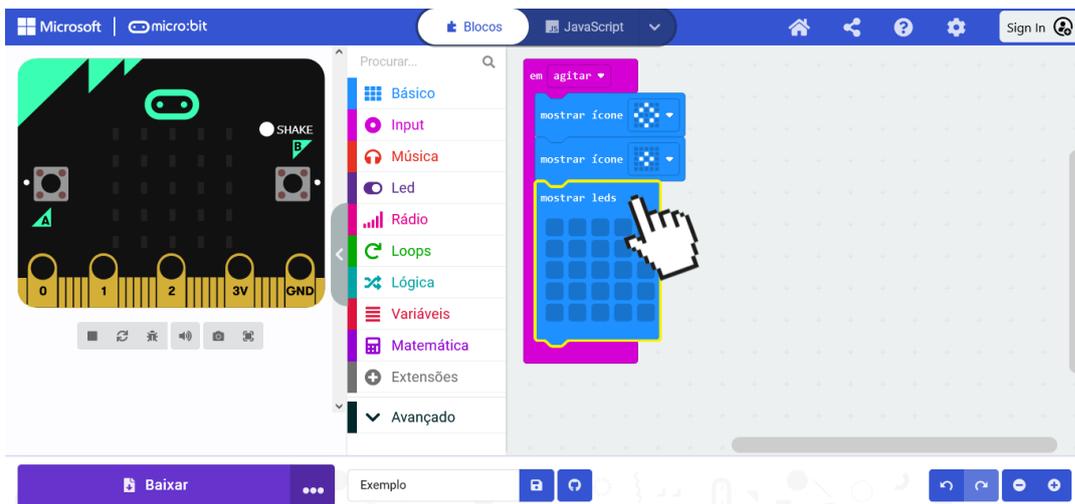


Figura 14.12: Montando um comando.

# Pedra, papel e tesoura

**11.** Selecione o LED central, Figura 14.13;

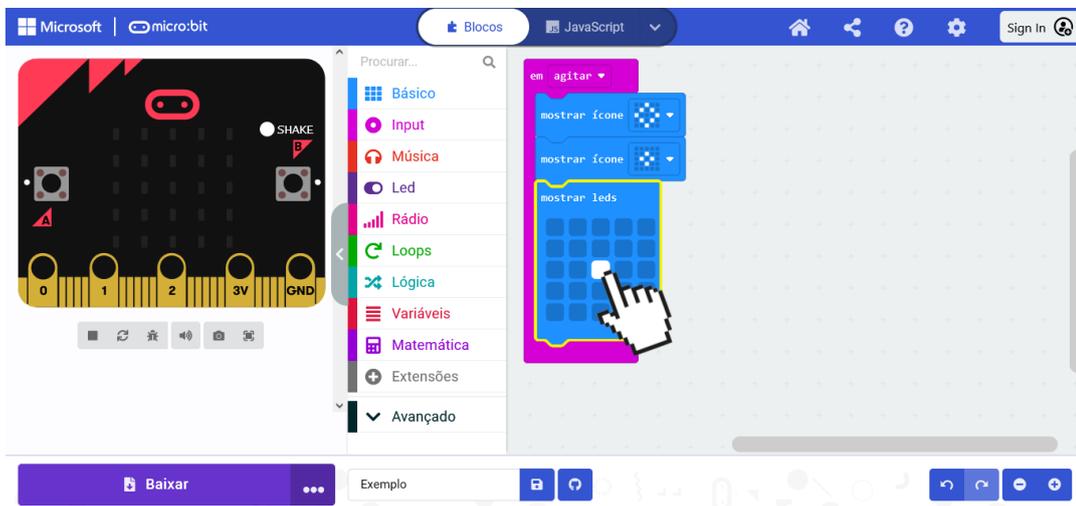


Figura 14.13: Acendendo um LED.

**12.** Na aba **Variáveis**, selecione o comando **definir sorteio para 0**, Figura 14.14;

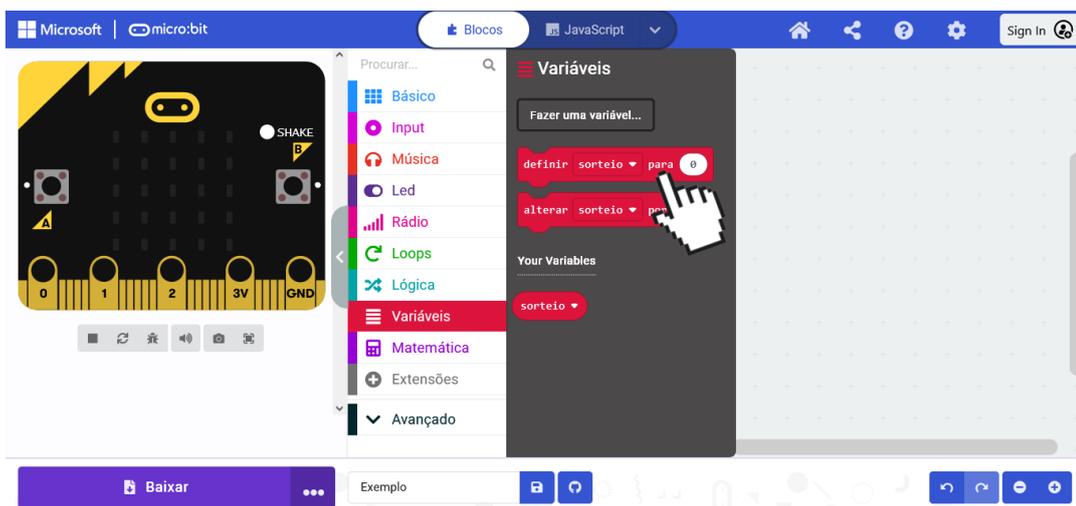


Figura 14.14: Selecionando um comando.



# Pedra, papel e tesoura

- 13.** Insira o último comando selecionado dentro do bloco **em agitar**, logo abaixo de **mostrar leds**, Figura 14.15;



Figura 14.15: Montando um comando.

- 14.** Na aba **Matemática**, selecione o comando **escolher aleatório 0 até 10**, Figura 14.16;



Figura 14.16: Selecionando um comando.

# Pedra, papel e tesoura

- 15.** Coloque o **escolher aleatório 0 até 10** no lugar do número **0** do comando **definir sorteio para 0**, Figura 14.17;

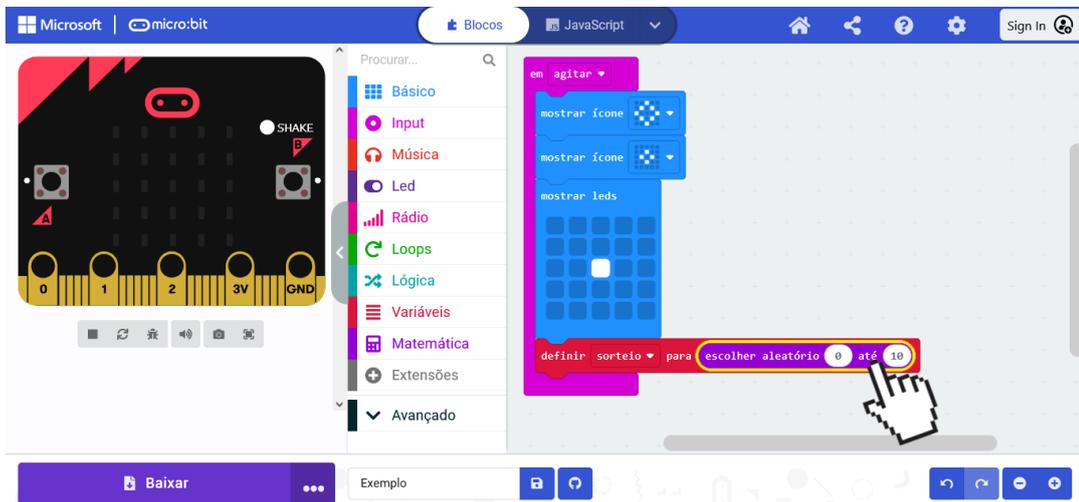


Figura 14.17: Montando um comando.

- 16.** Como só há três possibilidades, pedra, papel ou tesoura, altere o número **0** para **1** em **escolher aleatório 0 até 10**, Figura 14.18;

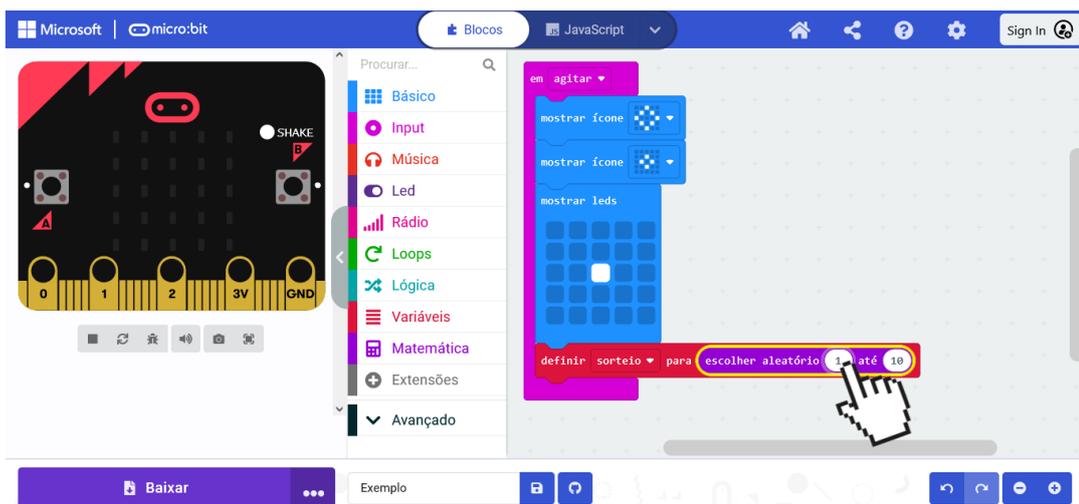


Figura 14.18: Alterando o número de um intervalo.



# Pedra, papel e tesoura

**17.** Ainda no **escolher aleatório 1 até 10**, altere o número **10** para **3**, Figura 14.19;

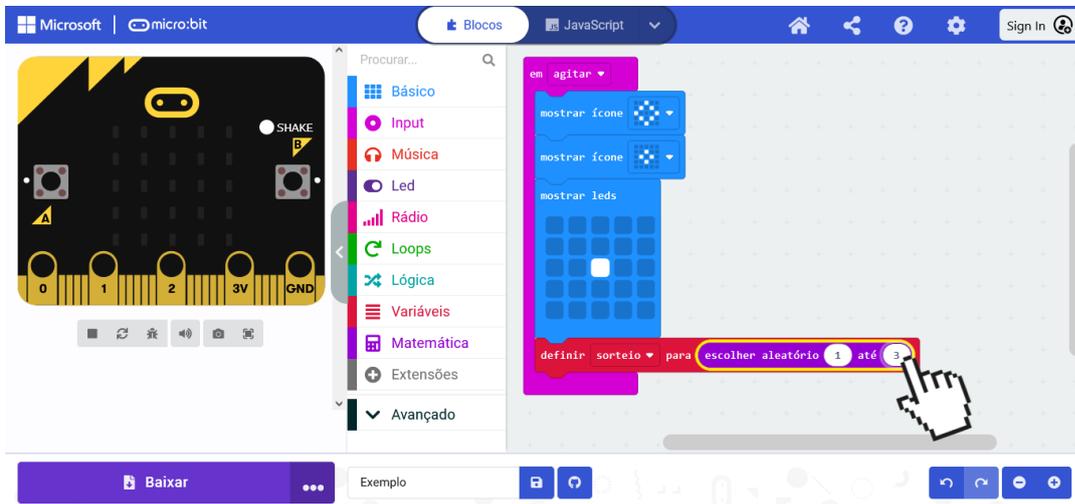


Figura 14.19: Alterando o número de um intervalo.

**18.** Na aba **Lógica**, selecione o bloco **se verdadeiro então**, três vezes, Figura 14.20;

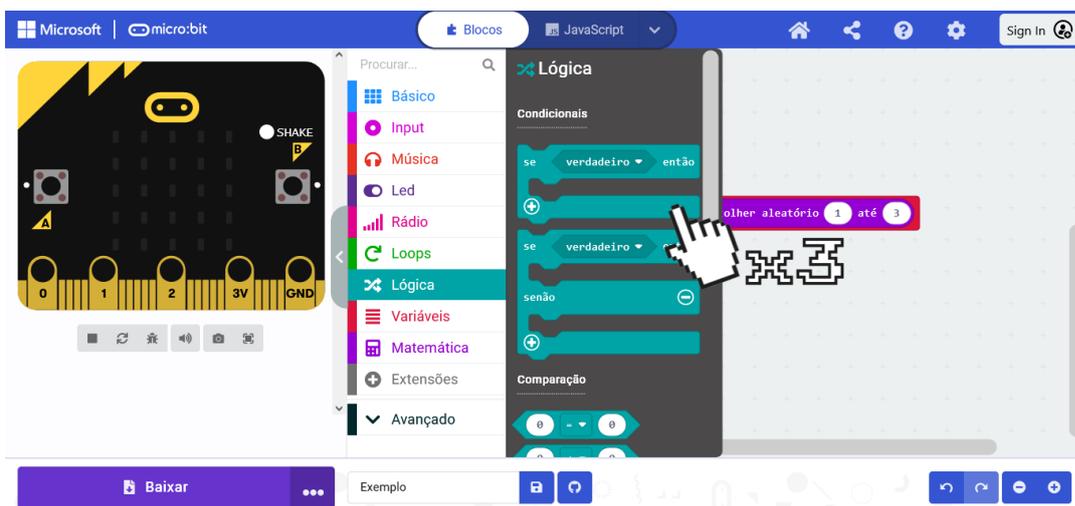


Figura 14.20: Selecionando um comando.

# Pedra, papel e tesoura

19. Insira os três comandos dentro do bloco de acionamento **em agitar**, logo abaixo de **definir sorteio para escolher aleatório 1 até 3**, um abaixo do outro, Figura 14.21;

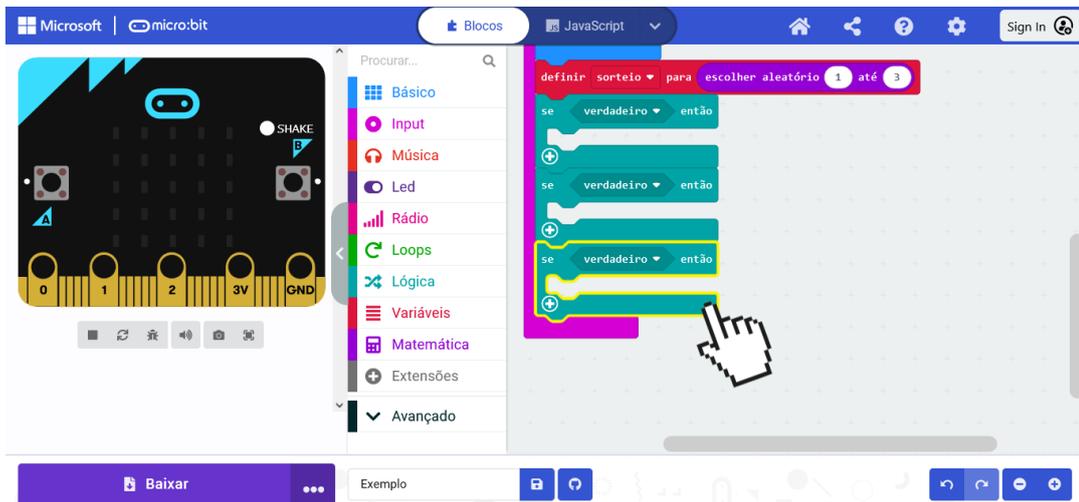


Figura 14.21: Montando comandos.

20. Na aba **Lógica**, selecione três vezes o comando de comparação **0 = 0**, Figura 14.22;

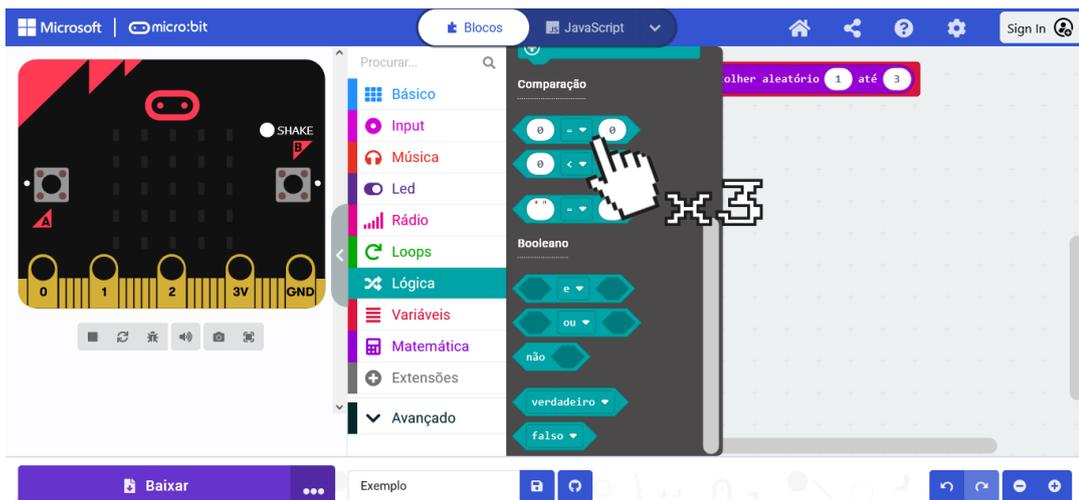


Figura 14.22: Selecionando um comando.



# Pedra, papel e tesoura

- 21.** Insira a comparação **0 = 0** no lugar do **verdadeiro** nos três blocos **se verdadeiro então**. Cada um ficará com a comparação **se 0 = 0 então**, como na Figura 14.23;

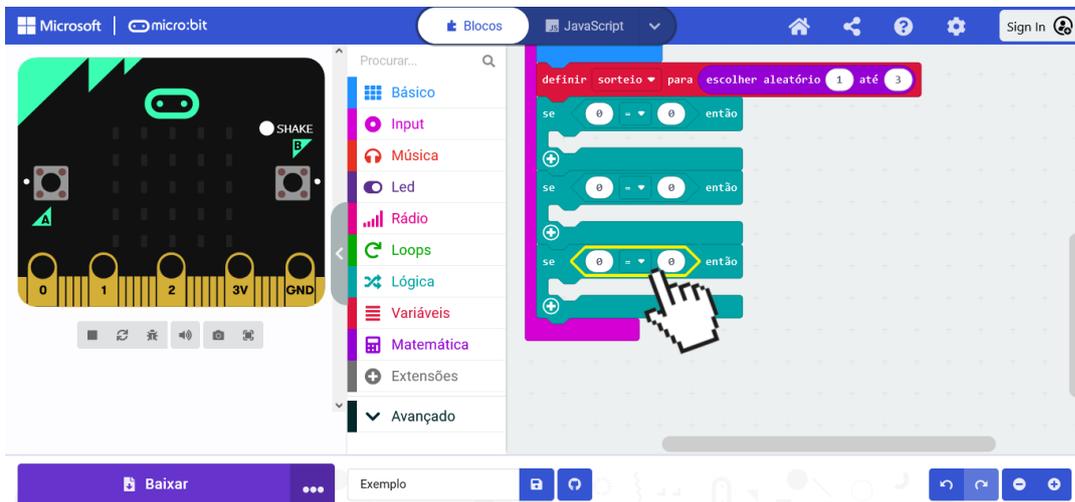


Figura 14.23: Montando comandos.

- 22.** Selecione três vezes a variável **sorteio**, como na Figura 14.24;

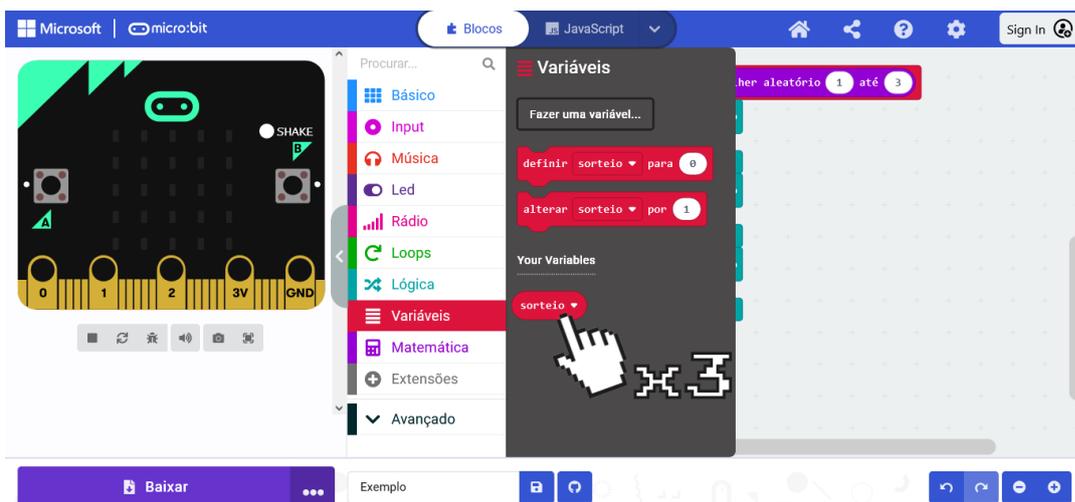


Figura 14.24: Selecionando uma variável.

# Pedra, papel e tesoura

- 23.** Coloque a variável **sorteio** no lugar do primeiro zero em cada comparação de **0 = 0**. Os três blocos ficarão como **se sorteio = 0 então**, Figura 14.25;

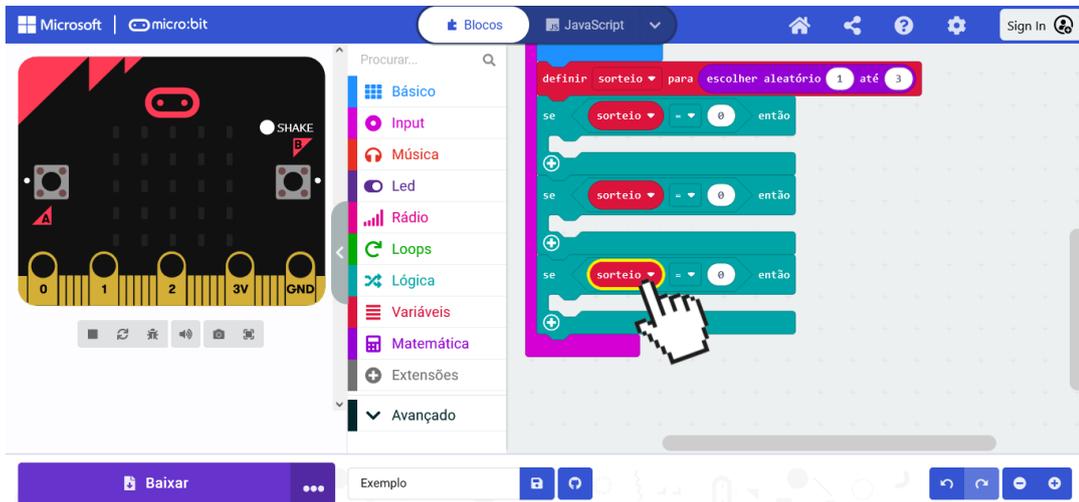


Figura 14.25: Montando comandos de lógica.

- 24.** Na primeira comparação, altere o número **zero** do comando **se sorteio = 0 então** para **1**, Figura 14.26;

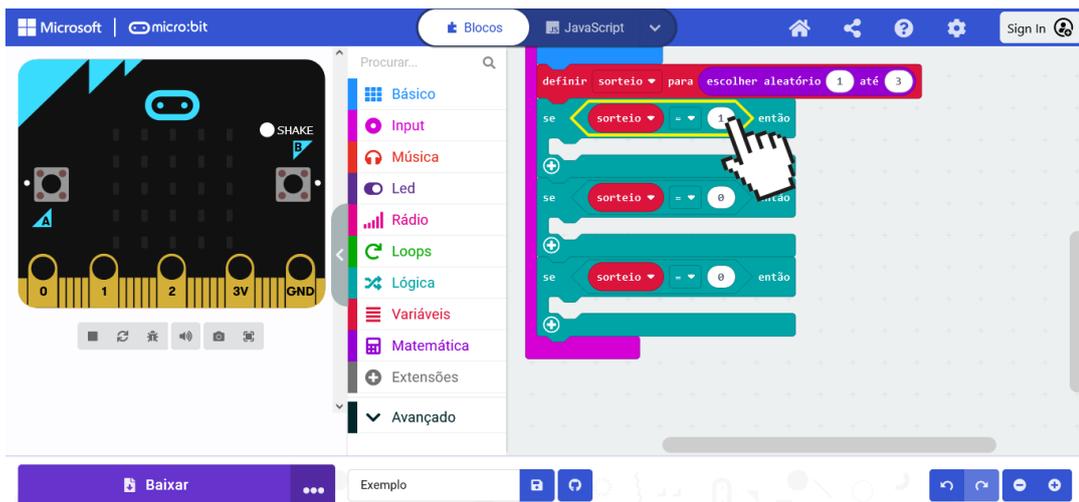


Figura 14.26: Alterando uma comparação.



# Pedra, papel e tesoura

**25.** Na aba **Básico**, selecione o comando **mostrar leds**, Figura 14.27;



Figura 14.27: Selecionando um comando.

**26.** Insira o comando **mostrar leds** selecionado dentro do bloco **se sorteio = 1 então**, Figura 14.28;



Figura 14.28: Montando um comando.

# Pedra, papel e tesoura

**27.** Crie o desenho que representará a pedra, Figura 14.29;

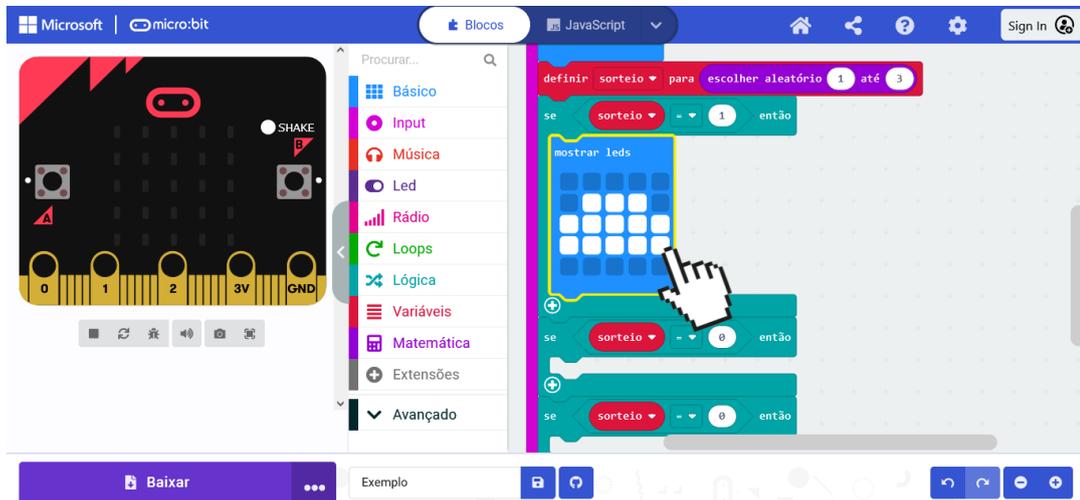


Figura 14.29: Desenhando nos LEDs.

**28.** Na segunda comparação, altere o número **zero** do comando **se sorteio = 0 então** para **2**, como na Figura 14.30;

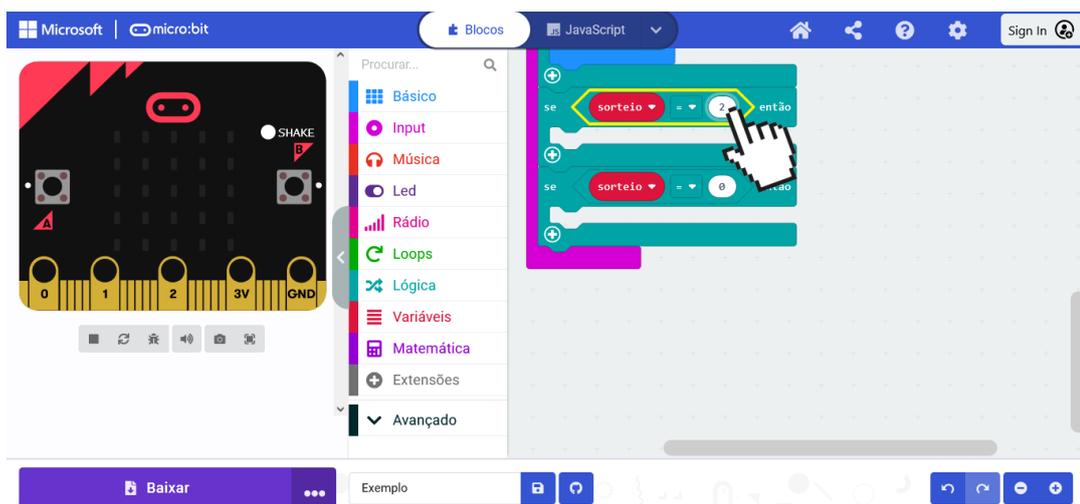


Figura 14.30: Alterando uma comparação.



# Pedra, papel e tesoura

**29.** Na aba **Básico**, selecione o comando **mostrar leds**, Figura 14.31;

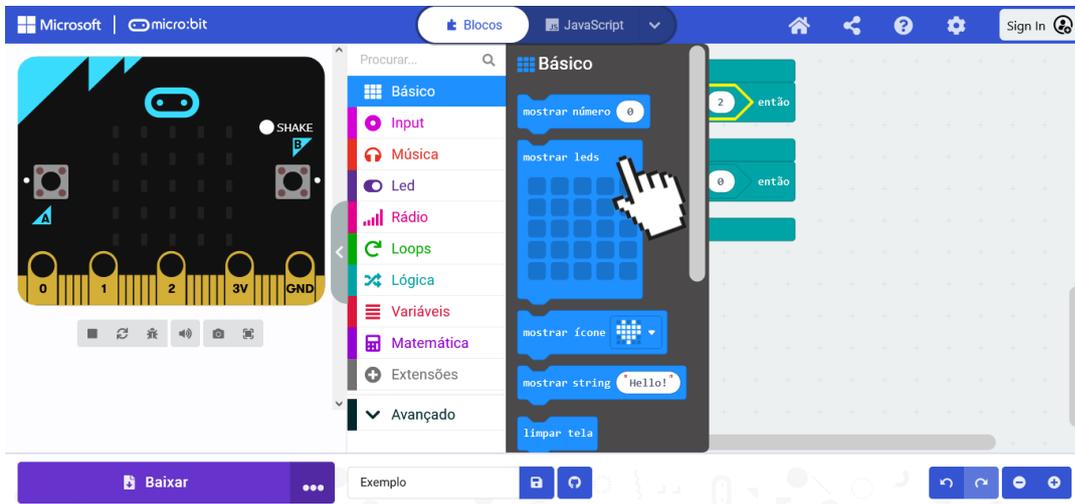


Figura 14.31: Selecionando um comando.

**30.** Insira o comando **mostrar leds** selecionado dentro do bloco **se sorteio = 2 então**, Figura 14.32;



Figura 14.32: Montando um comando.

# Pedra, papel e tesoura

**31.** Crie o desenho que representará o papel, Figura 14.33;

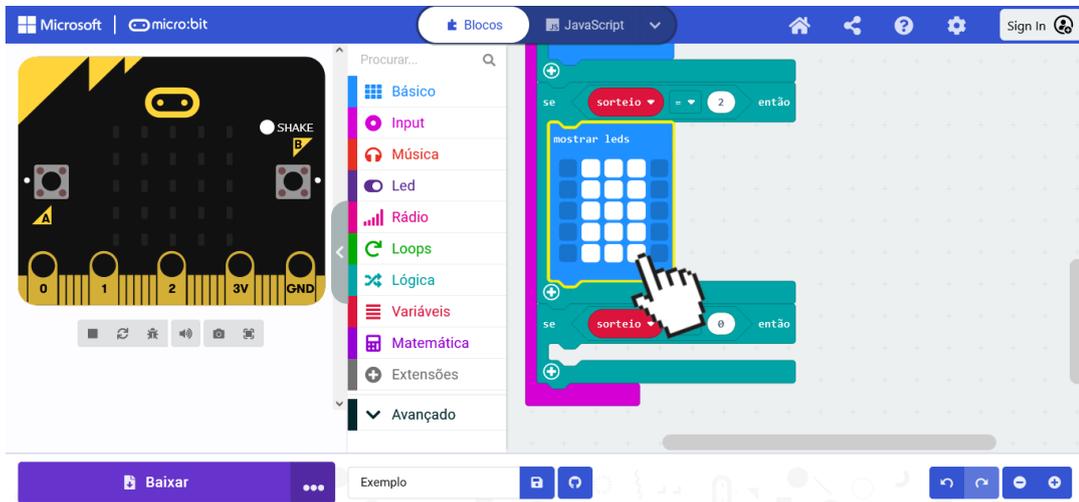
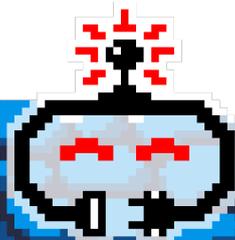


Figura 14.33: Desenhando nos LEDs.

**32.** Na última comparação, altere o número **zero** do comando **se sorteio = 0 então** para **3**, como na Figura 14.34;



Figura 14.34: Alterando uma comparação.



# Pedra, papel e tesoura

**33.** Na aba **Básico**, selecione o comando **mostrar ícone**, Figura 14.35;

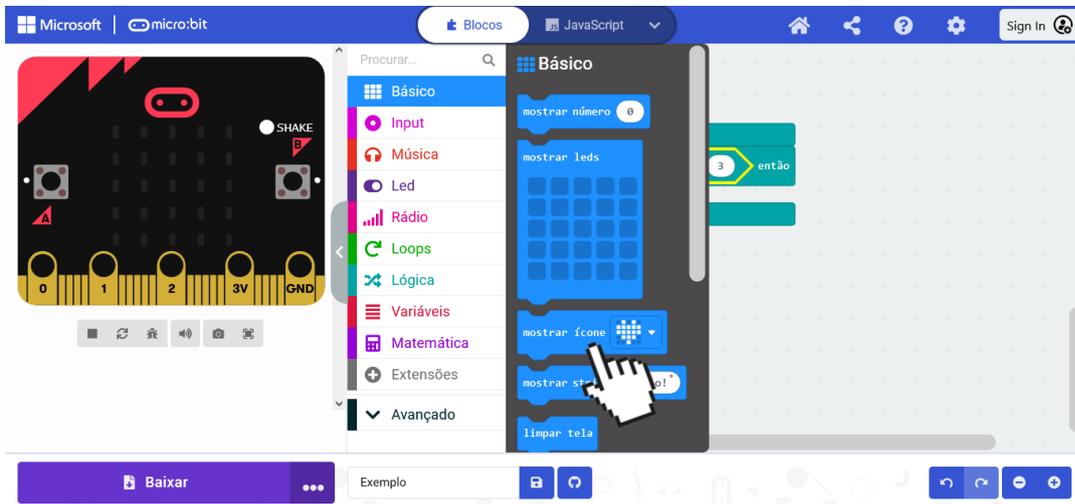


Figura 14.35: Selecionando um comando.

**34.** Insira o comando **mostrar ícone** no último bloco, dentro de **se sorteio = 3 então**, Figura 14.36;



Figura 14.36: Montando um comando.

# Pedra, papel e tesoura

**35.** Selecione o ícone tesoura, como na Figura 14.37.

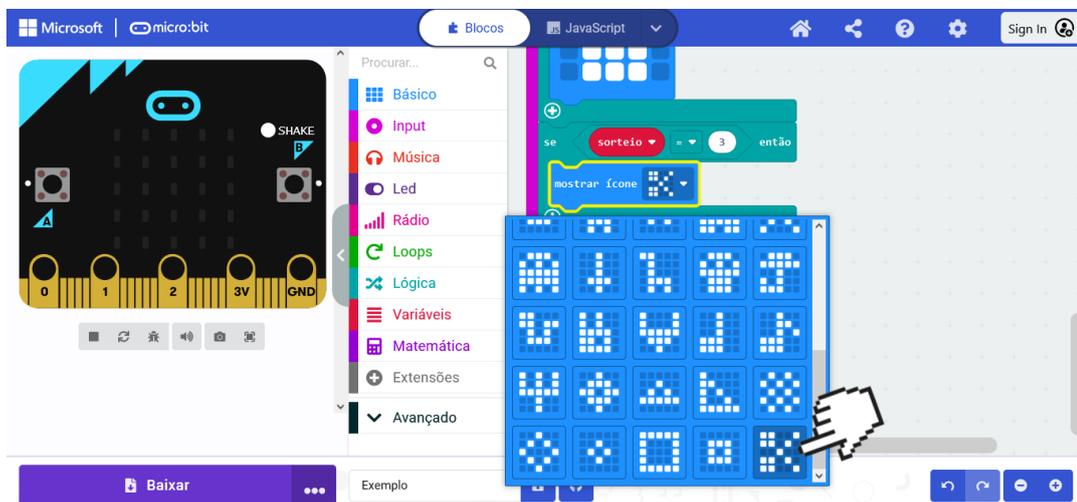


Figura 14.37: Escolhendo um ícone.

**Baixe o programa para o micro:bit e jogue pedra, papel e tesoura agitando ele!**

## MICRO:DICA

Para que haja um maior envolvimento das crianças, propomos uma pequena gincana dividindo a sala em grupos. Crie um caminho por onde as crianças possam seguir em direções opostas e, quando se encontrarem, façam o jogo pedra, papel e tesoura. Quem chegar ao lado oposto primeiro, pontua.



[Clique aqui para acessar o código.](#)

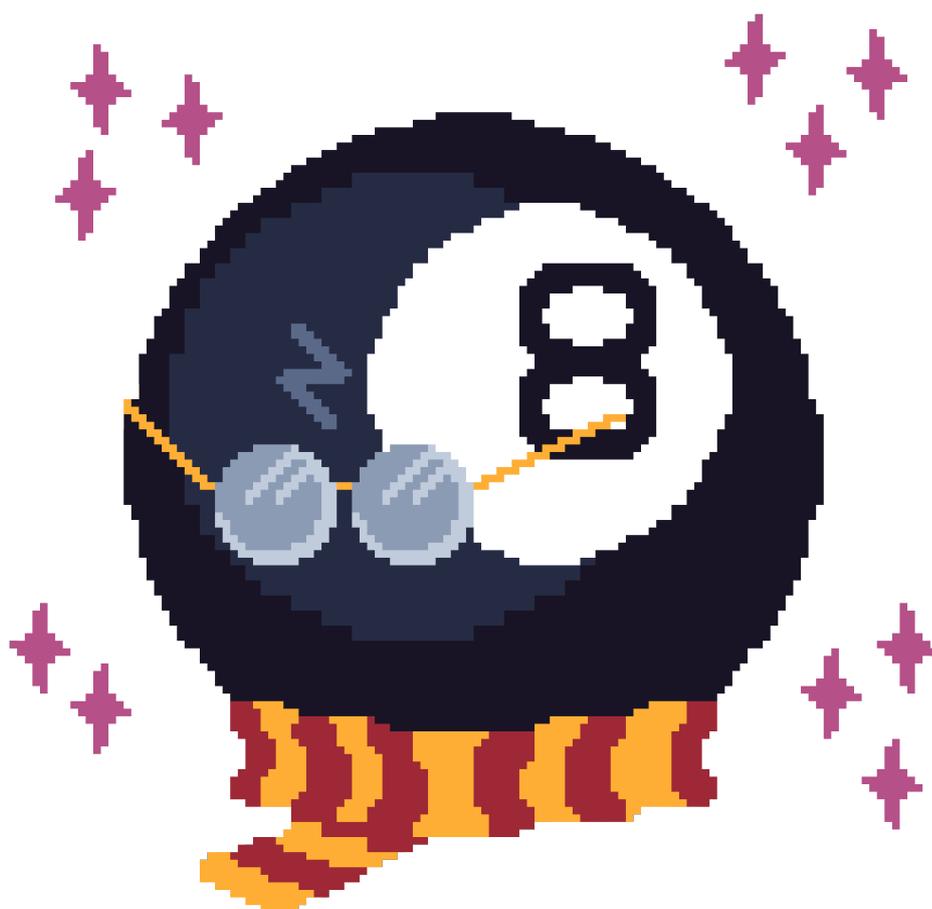




Pedra, papel e tesoura

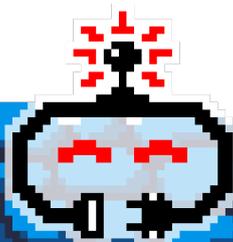


## Capítulo 15: Bola mágica



O que acha de recriar um brinquedo clássico com o micro:bit? A Bola 8 mágica foi inventada nos EUA na década de 1950. O brinquedo é no formato de uma bola de bilhar grande que, quando balançada, mostra uma das 20 respostas diferentes de forma aleatória. As respostas podem ser positivas, negativas ou algo intermediário.

A Bola 8 mágica pode ser feita com o micro:bit usando o mesmo princípio do jogo pedra, papel e tesoura. Mas desta vez, quando o micro:bit for agitado, será mostrado um sim, um não ou um talvez, aleatoriamente.



# Bola mágica

## Passo a passo:

1. Primeiramente, defina uma variável. Neste exemplo, daremos o nome de **resposta**, Figuras 15.1 e 15.2;

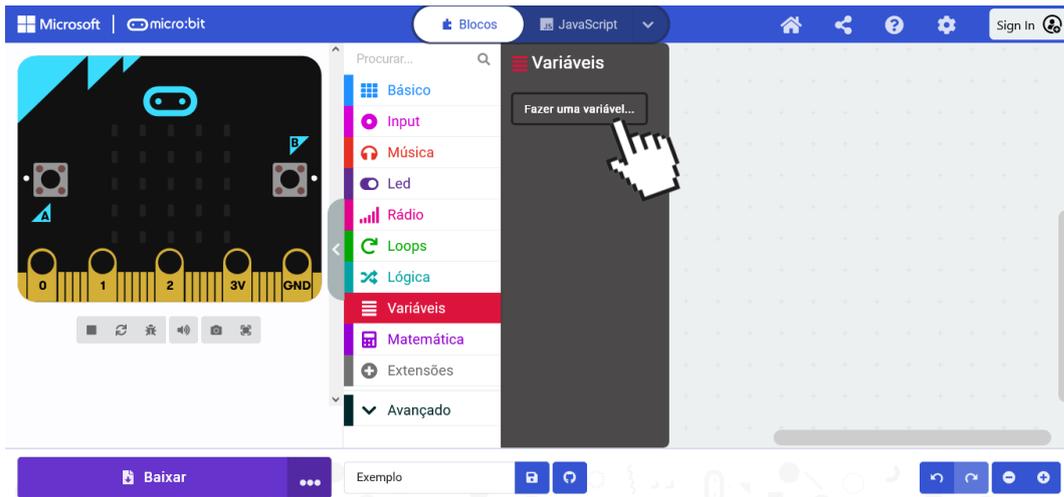


Figura 15.1: Fazendo uma variável.

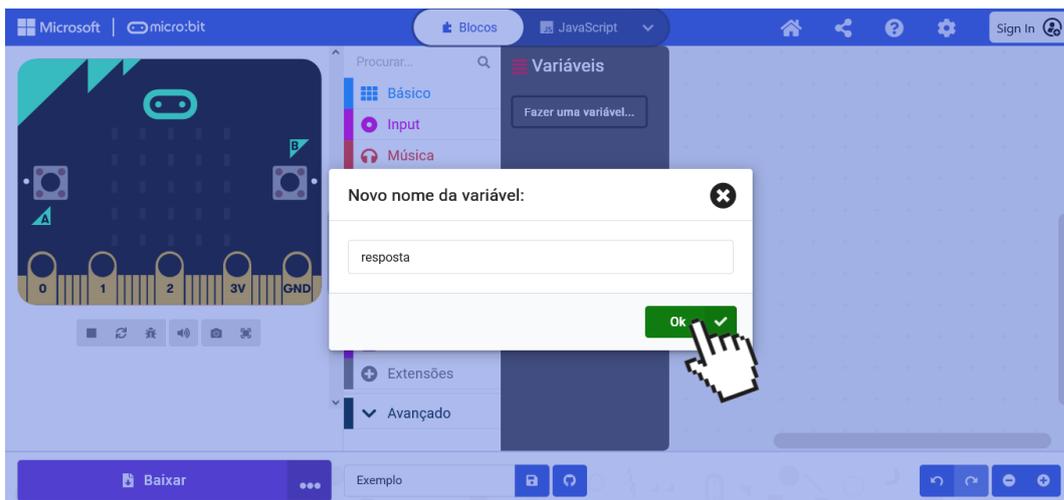


Figura 15.2: Dando nome a uma variável.

# Bola mágica

2. Selecione, na aba **Input**, o comando **em agitar**, Figuras 15.3 e 15.4;

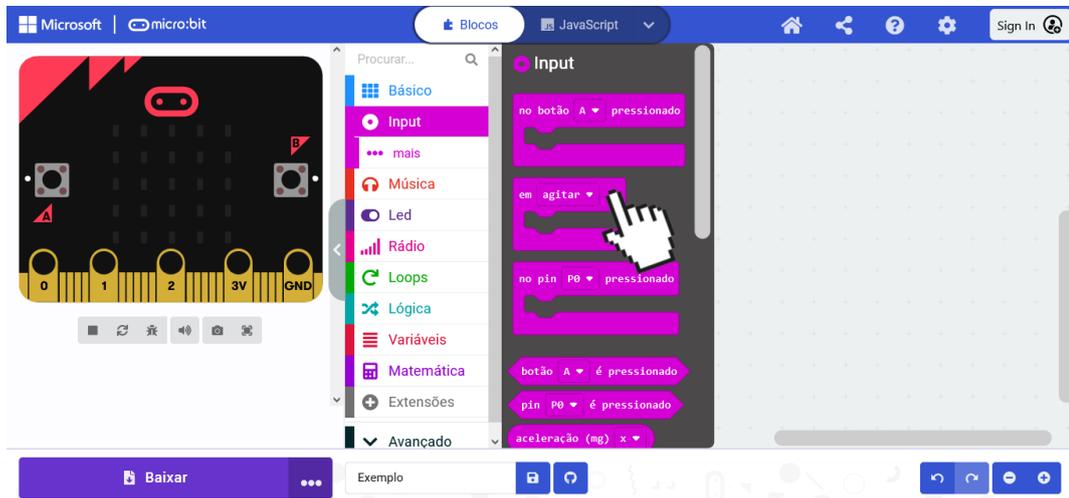


Figura 15.3: Selecionando um bloco de acionamento.

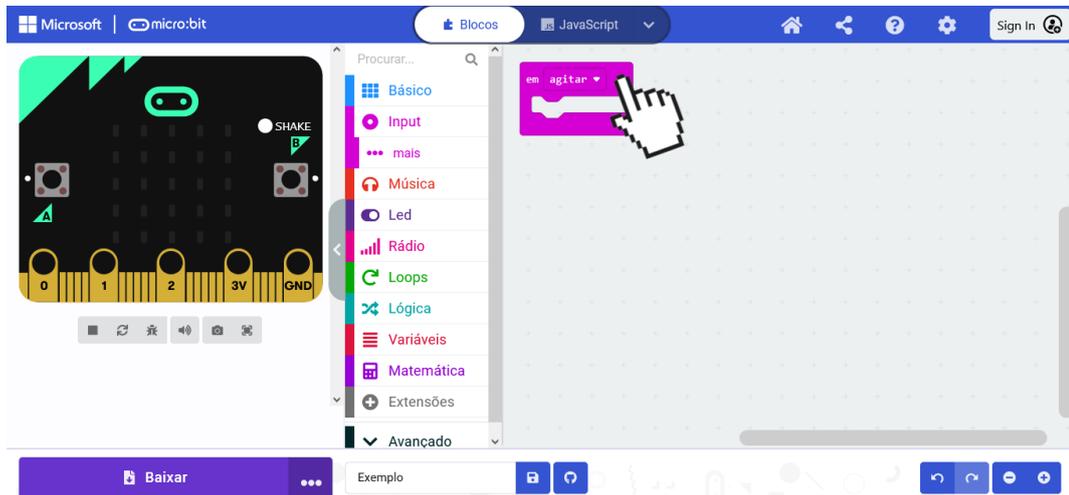


Figura 15.4: Montando um comando.



# Bola mágica

3. Coloque o comando **definir resposta para 0** dentro do bloco **em agitar**, Figuras 15.5 e 15.6;

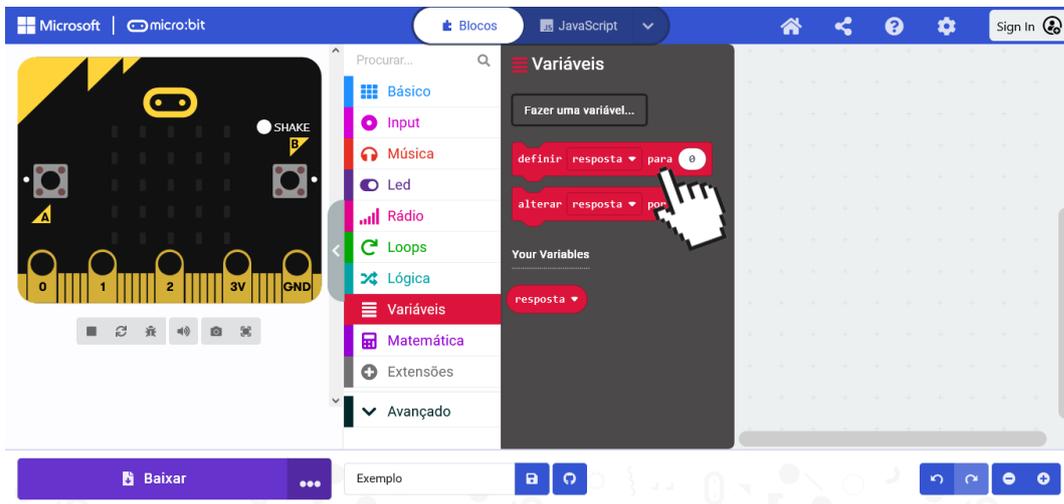


Figura 15.5: Selecionando um comando.

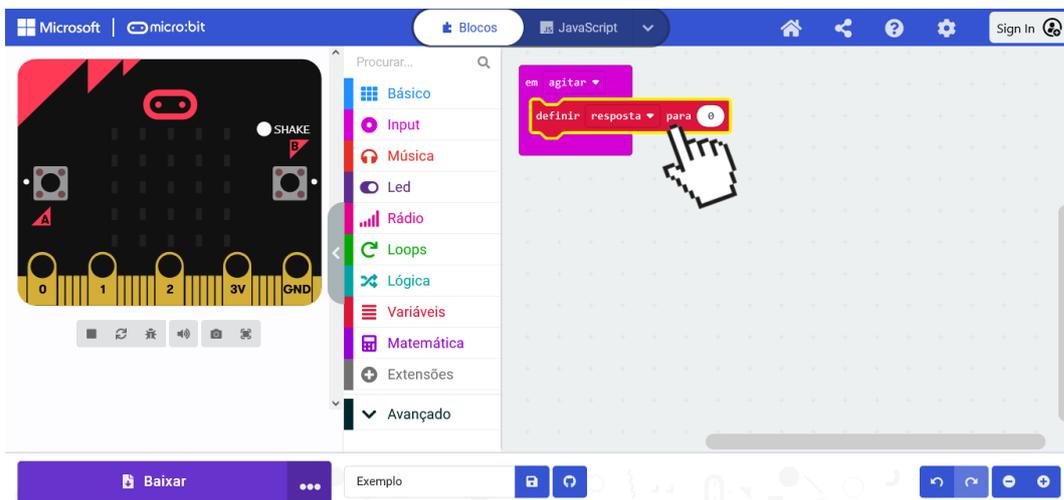


Figura 15.6: Montando um comando.

# Bola mágica

4. Substitua o número 0 pela função matemática **escolher aleatório 0 até 10** e limite os números do intervalo para 1 até 3, Figuras 15.7, 15.8, 15.9 e 15.10;



Figura 15.7: Selecionando um comando.

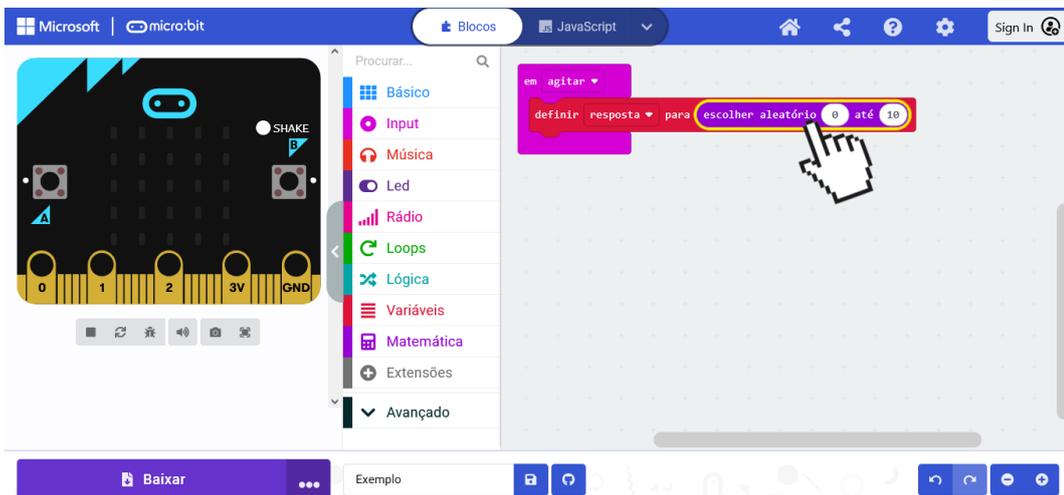


Figura 15.8: Montando uma comando.



# Bola mágica

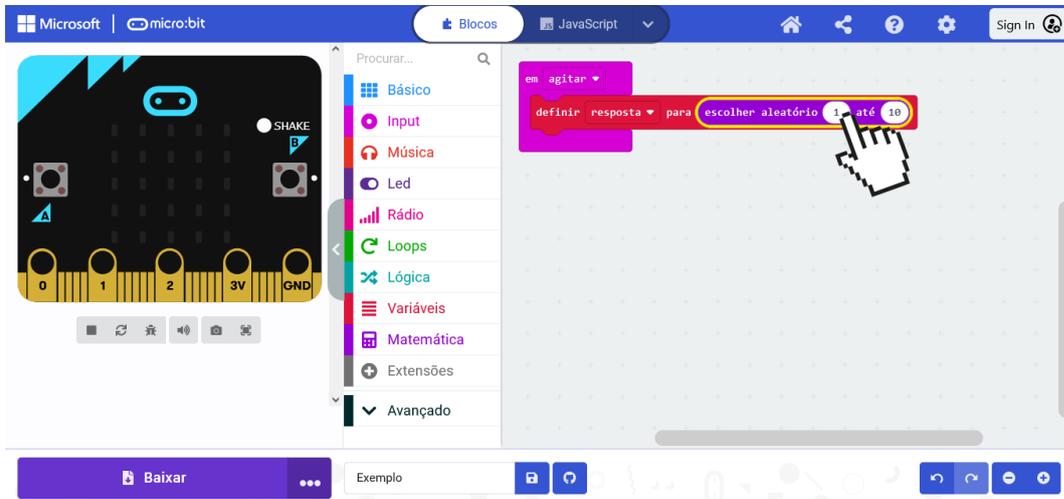


Figura 15.9: Alterando o início de um intervalo.

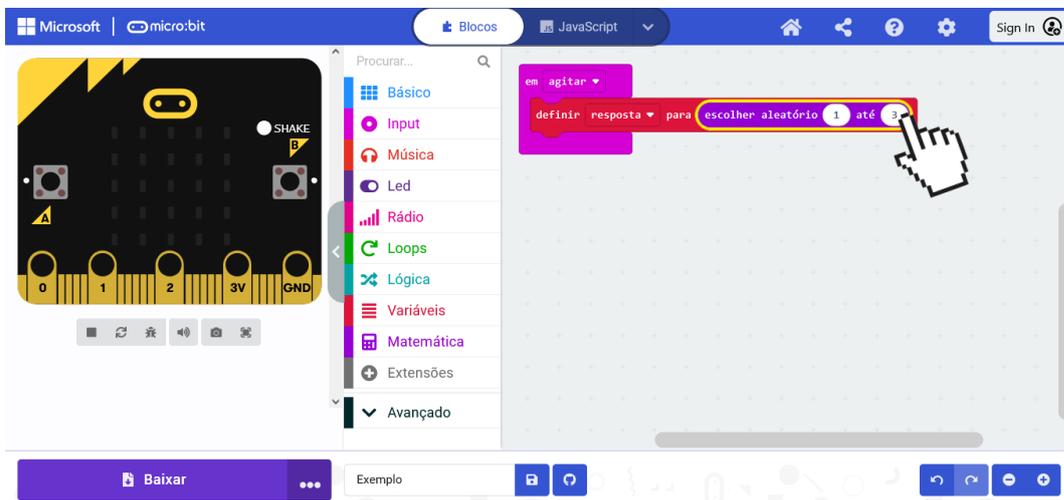


Figura 15.10: Alterando o final de um intervalo.

# Bola mágica

5. Coloque, logo abaixo do comando da **variável**, a função lógica **se verdadeiro então, senão**, Figuras 15.11 e 15.12;

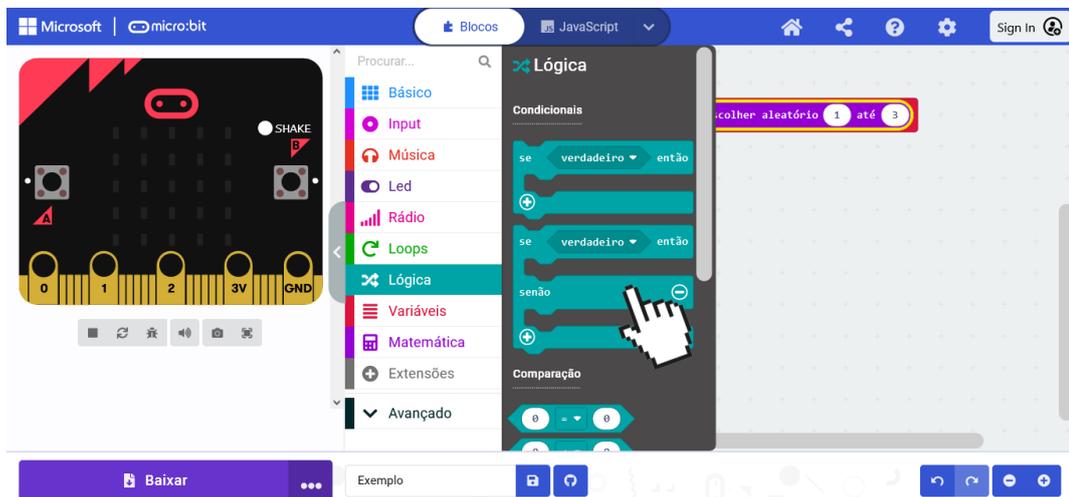


Figura 15.11: Selecionando um comando.



Figura 15.12: Montando um comando.



# Bola mágica

6. Substitua o **verdadeiro** da lógica pela comparação **0 = 0**, Figuras 15.13 e 15.14;

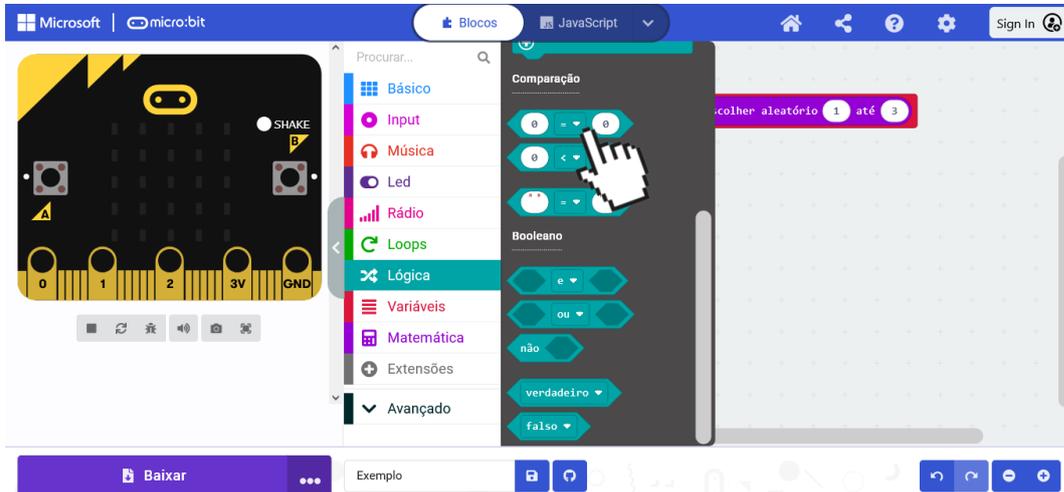


Figura 15.13: Selecionando uma comparação.



Figura 15.14: Montando uma comparação.

# Bola mágica

7. Coloque a variável **resposta** sobre o primeiro número **0** da comparação e troque o segundo pelo número **3**, Figuras 15.15, 15.16 e 15.17;

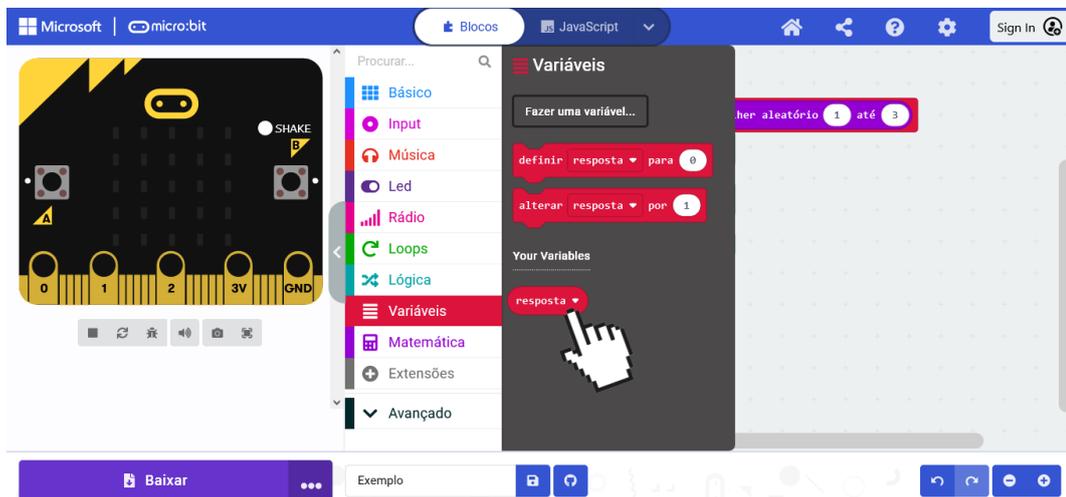


Figura 15.15: Selecionando uma variável.



Figura 15.16: Montando uma comparação.



# Bola mágica



Figura 15.17: Alterando uma comparação.

- Defina o que acontecerá quando a resposta sorteada for o número 3. Para isso, use o **mostrar ícone** e escolha o ícone que representa o **sim**, Figuras 15.18, 15.19 e 15.20;



Figura 15.18: Selecionando um comando.

# Bola mágica

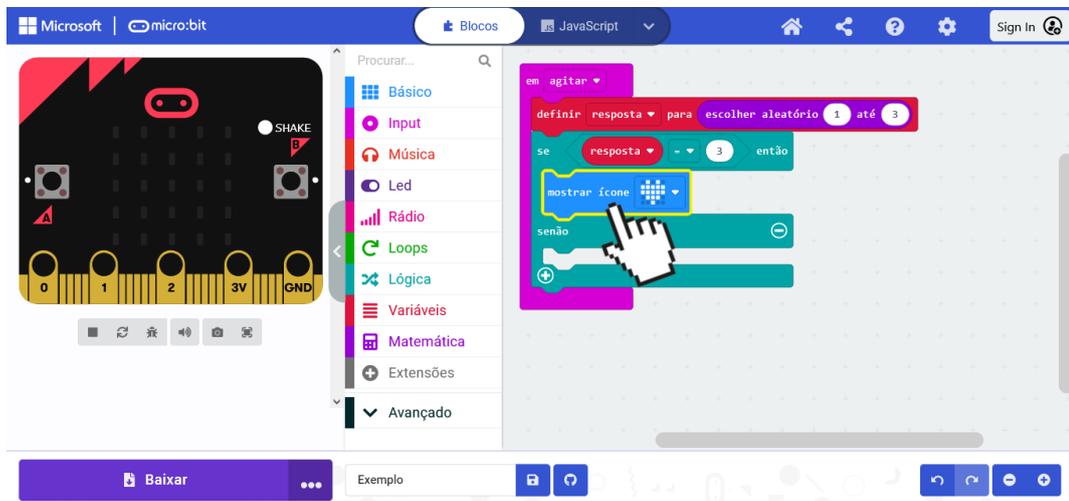


Figura 15.19: Montando um comando.

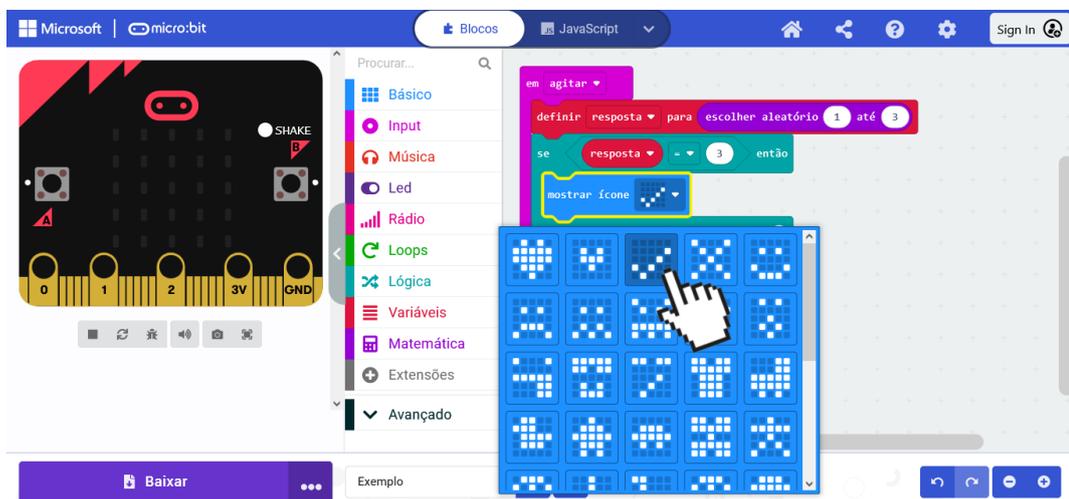
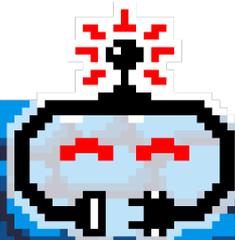


Figura 15.20: Alterando um ícone.



# Bola mágica

9. Adicione mais dois casos para que o jogo tenha as respostas sim, não e talvez. Para isso, clique no sinal de +, no canto inferior esquerdo do bloco de **lógica**, para adicionar mais uma condição, Figura 15.21;

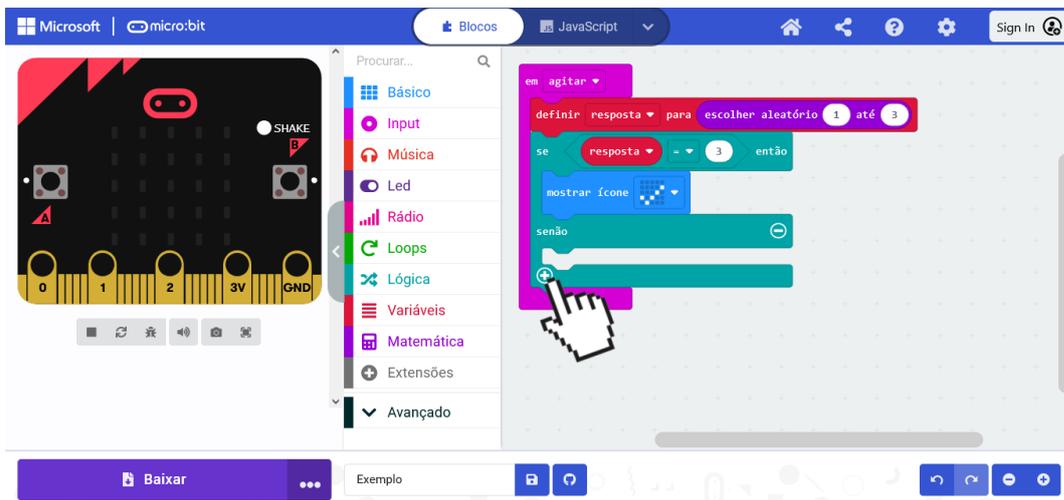


Figura 15.21: Adicionando uma condição.

10. Repita os passos de 6 a 8 para incluir a opção de **não**. Substitua o **verdadeiro** pela comparação  $0 = 0$ , troque o zero da esquerda pela variável **resposta** e altere o zero da direita por **2**, como na Figura 15.22;

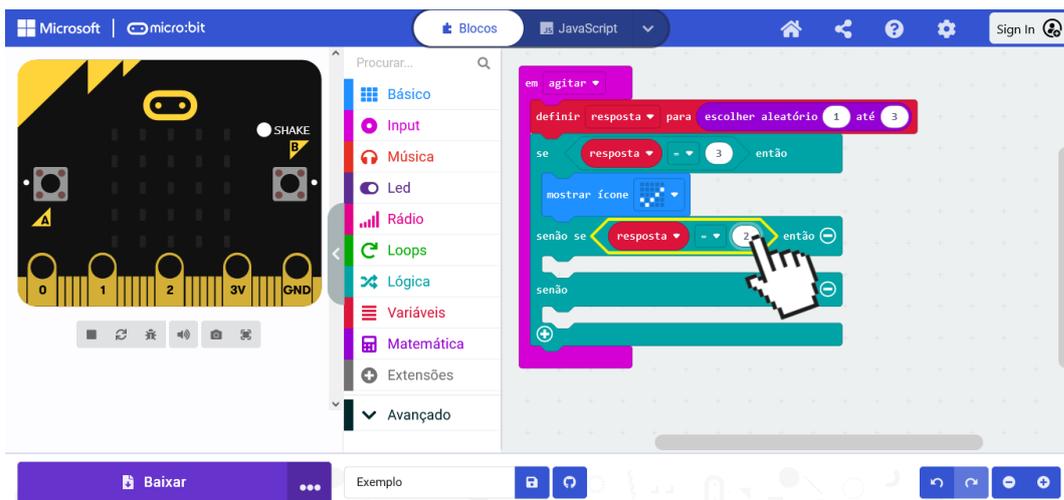


Figura 15.22: Montando e alterando uma comparação.

# Bola mágica

11. Selecione o comando **mostrar ícone** e coloque-o na segunda condição. Depois altere-o para o ícone de **não**, como nas Figuras 15.23 15.24 e 15.25;

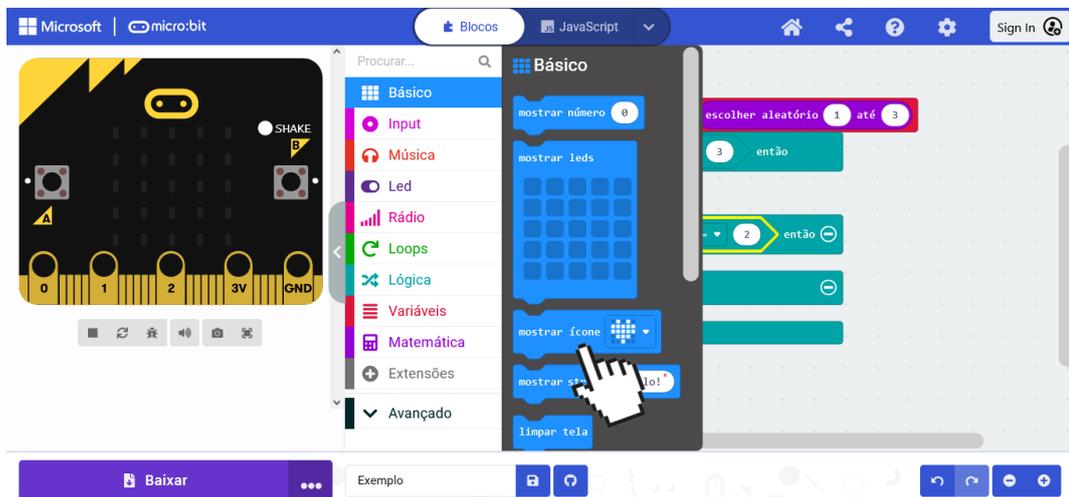


Figura 15.23: Selecionando um ícone.

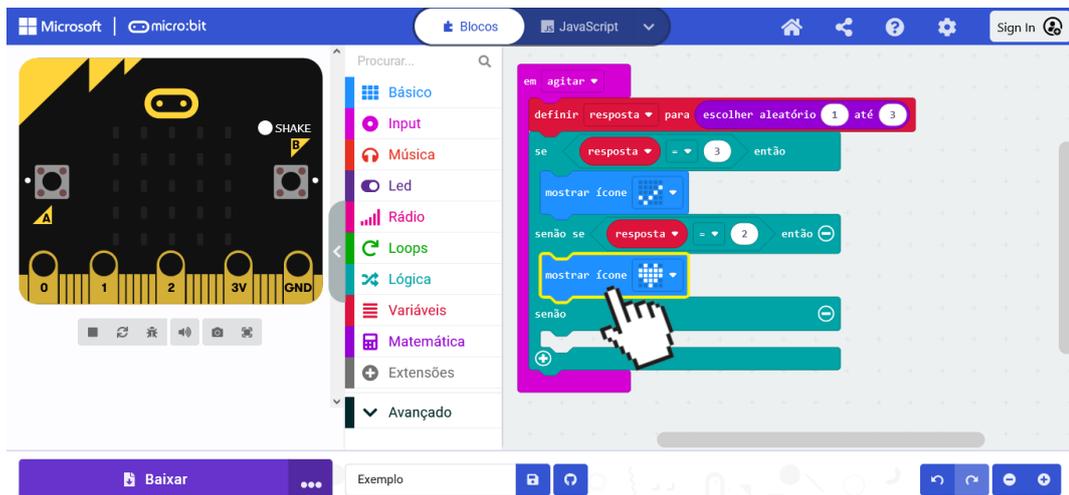


Figura 15.24: Montando um ícone.



# Bola mágica

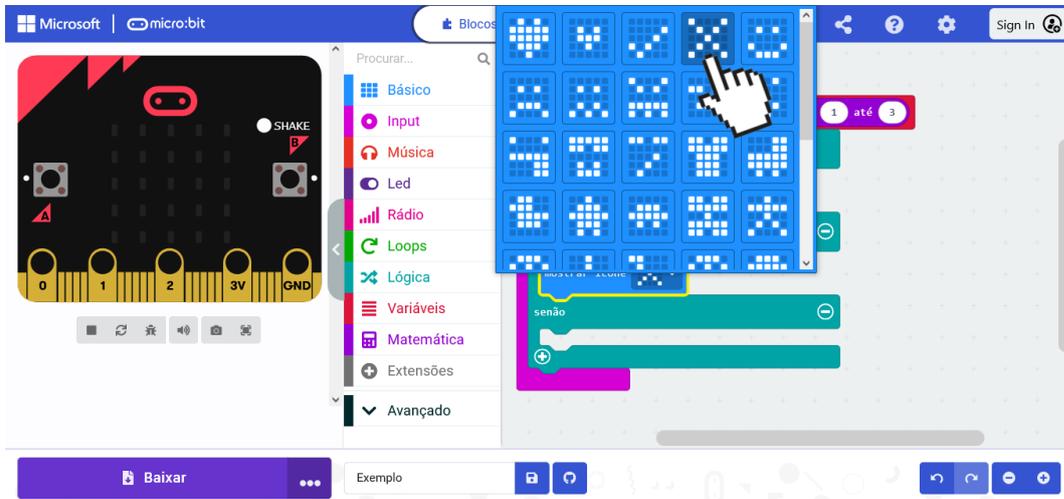


Figura 15.25: Alterando um ícone.

- 12.** Por último, coloque abaixo de **senão** o ícone de **adormecido** (que representará o **talvez**), que aparecerá caso a resposta sorteada for o número **1**, como nas Figuras 15.26 e 15.27.



Figura 15.26: Selecionando um ícone.

# Bola mágica

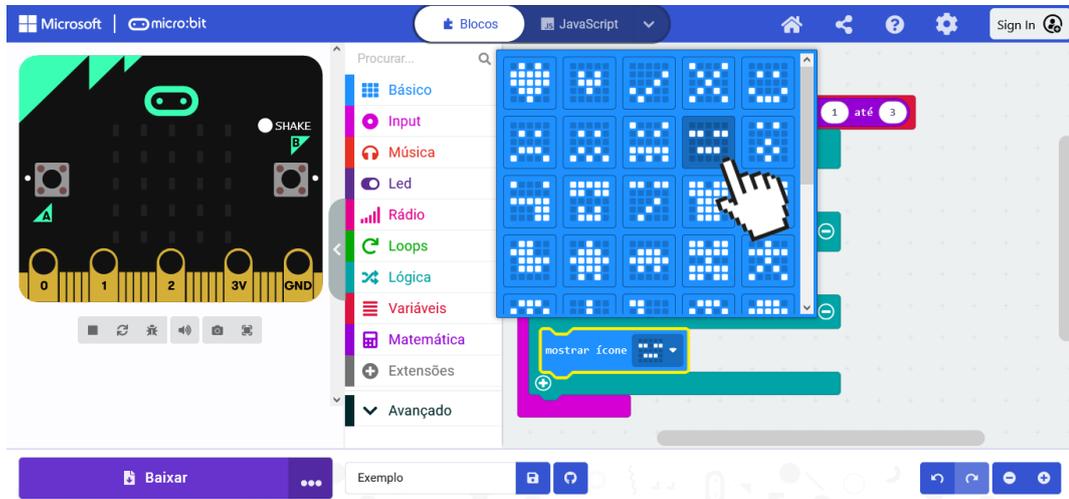


Figura 15.27: Alterando um ícone.

**Baixe o programa para o micro:bit e divirta-se fazendo perguntas!**

[Clique aqui para acessar o código.](#)





Bola mágica



# Capítulo 16: Batata Quente



Você já brincou de batata quente? Então, hoje o micro:bit será programado para essa brincadeira.

O jogo será quase igual ao tradicional, a diferença será que, ao invés de uma pessoa escolher aleatoriamente quando a batata queimou, quem fará isso será o micro:bit. Usaremos o comando de pressionar um botão para que o micro:bit mostre se a batata queimou ou não.

Agora que já sabe como será o jogo, que tal começar a programar para brincar?



# Batata Quente

## Passo a passo:

1. Na aba **Input**, selecione o bloco de acionamento **no botão A pressionado**, como nas Figuras 16.1 e 16.2;

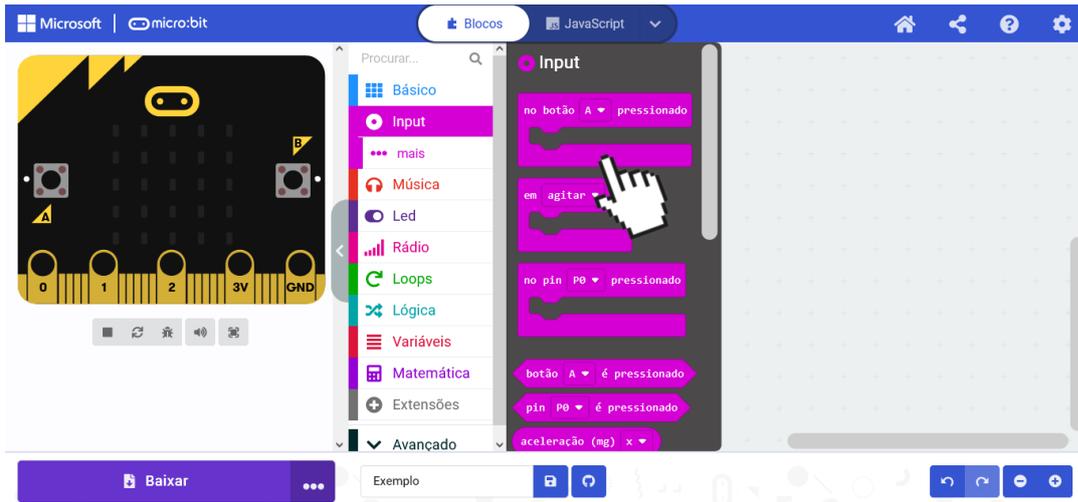


Figura 16.1: Selecionando um bloco de acionamento.



Figura 16.2: Montando um acionamento.

# Batata Quente

- Na aba **Básico**, selecione dois comandos **mostrar ícone** e coloque-os dentro do bloco de acionamento **no botão A pressionado**. Altere o ícone do primeiro para **alvo** e do outro para **diamante pequeno**, Figuras 16.3, 16.4, 16.5, 16.6 e 16.7;

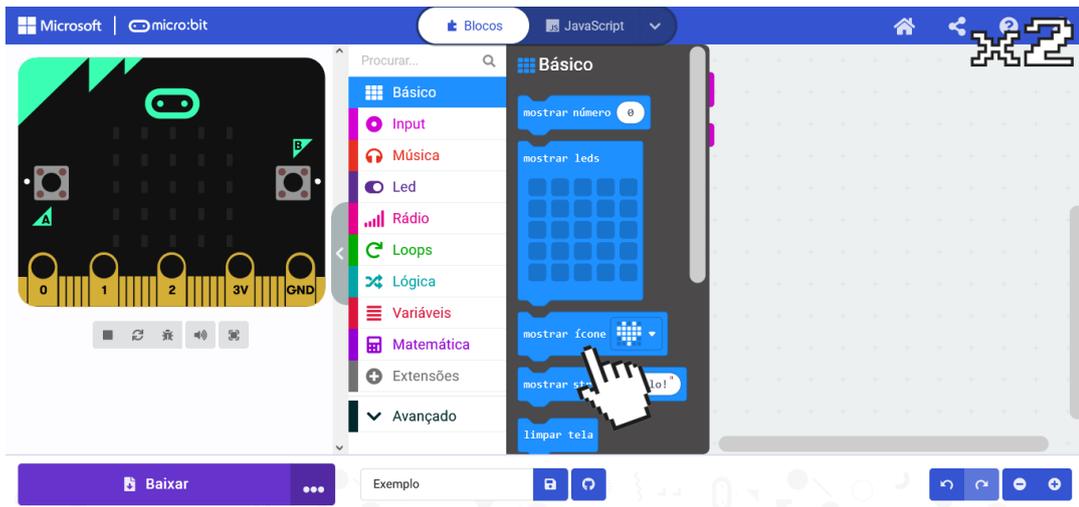


Figura 16.3: Selecionando um comando.



Figura 16.4: Montando comandos.



# Batata Quente

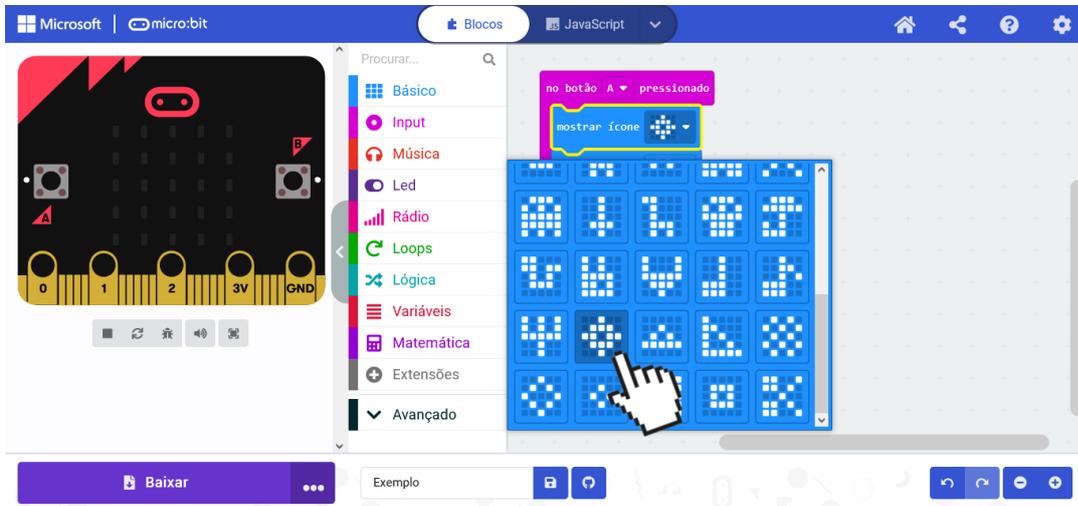


Figura 16.5: Alterando um ícone.

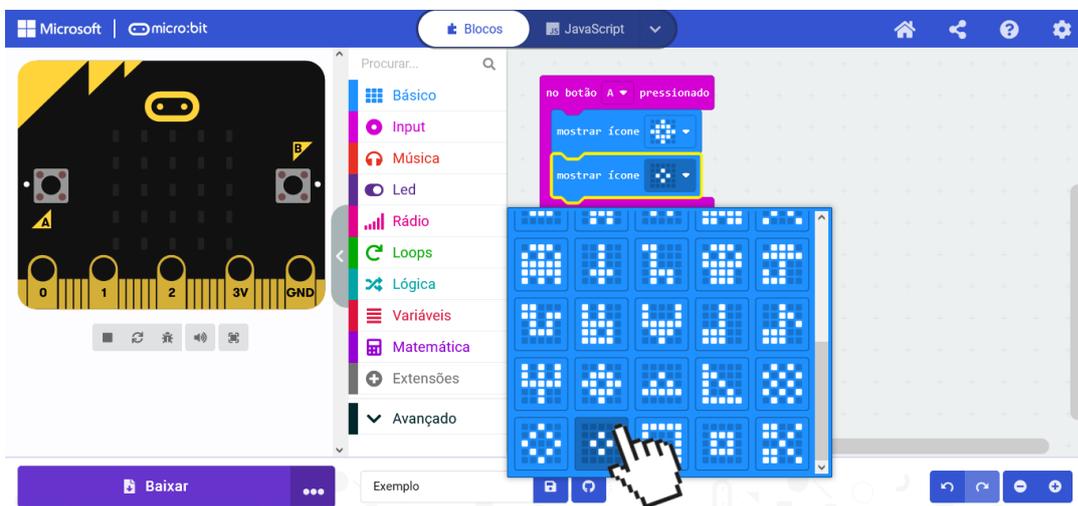


Figura 16.6: Alterando um ícone.

# Batata Quente

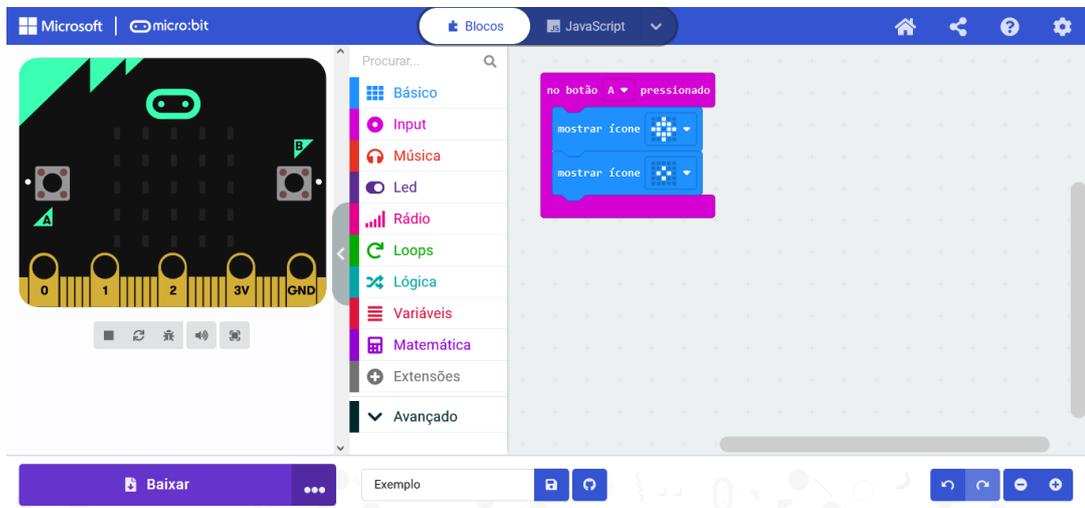


Figura 16.7: Montando comandos.

3. Na aba **Lógica**, selecione o bloco **se verdadeiro então senão** e coloque-o logo abaixo de **mostrar ícone**, também dentro do bloco de acionamento **no botão A pressionado**, Figuras 16.8 e 16.9;

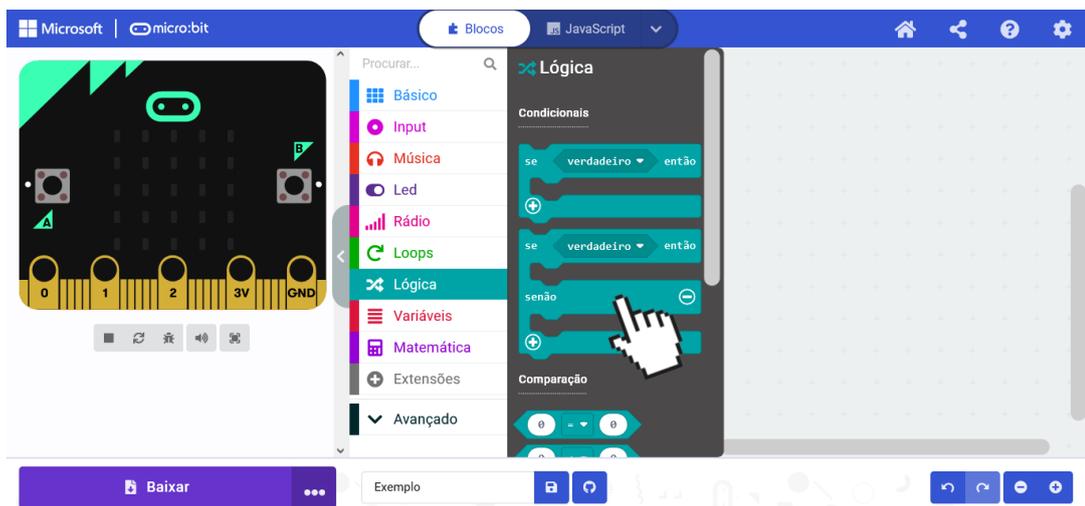


Figura 16.8: Selecionando um bloco de lógica.



# Batata Quente

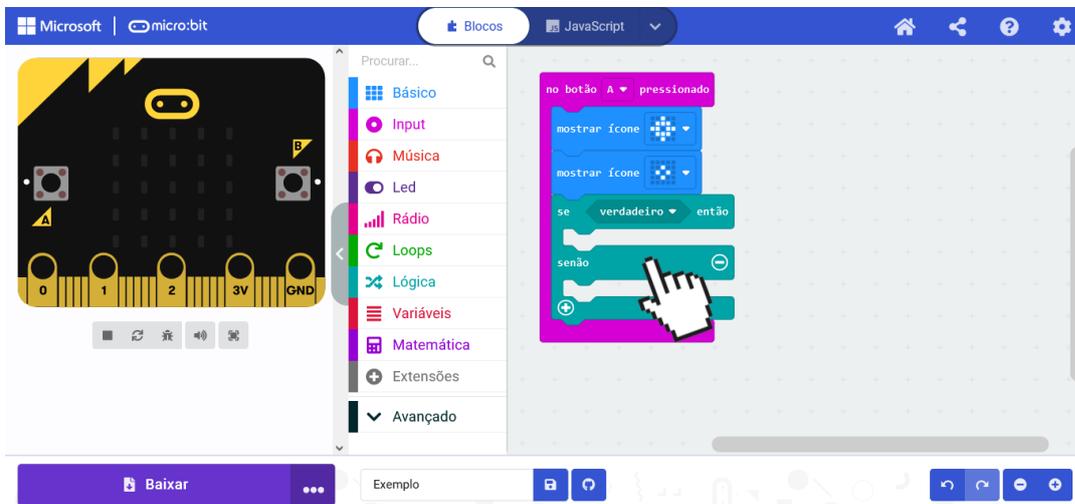


Figura 16.9: Montando um comando.

4. Ainda na aba **Lógica**, selecione a comparação **0 = 0** e coloque-a no lugar de **verdadeiro** em **se verdadeiro então**, Figura 16.10 e 16.11;



Figura 16.10: Selecionando uma comparação.

# Batata Quente

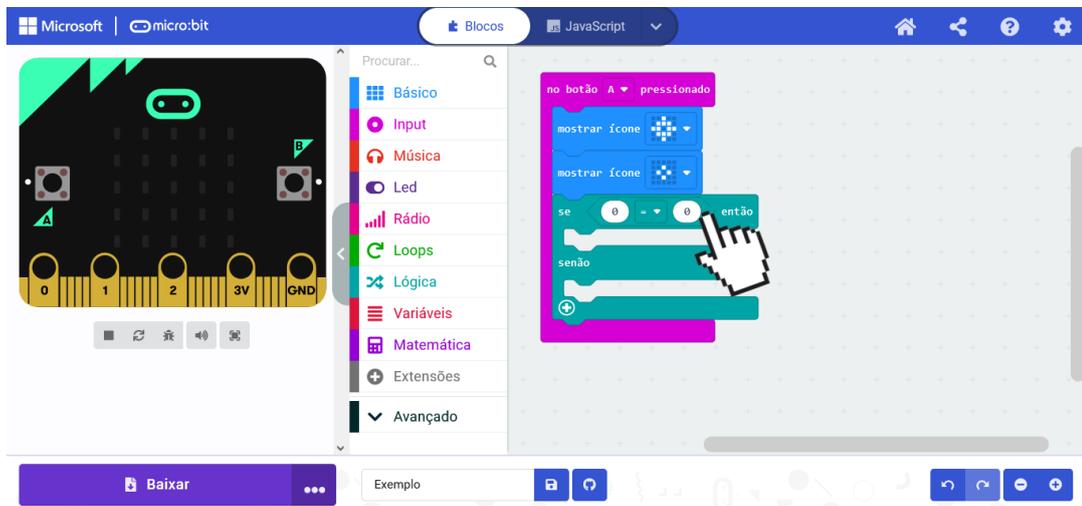


Figura 16.11: Montando uma comparação.

5. Agora, na aba **Matemática**, selecione o comando **escolher aleatório 0 até 10** e coloque-o no lugar do segundo número **0** da comparação **0 = 0**. Depois, altere o número **10** para **9**, Figuras 16.12, 16.13 e 16.14;

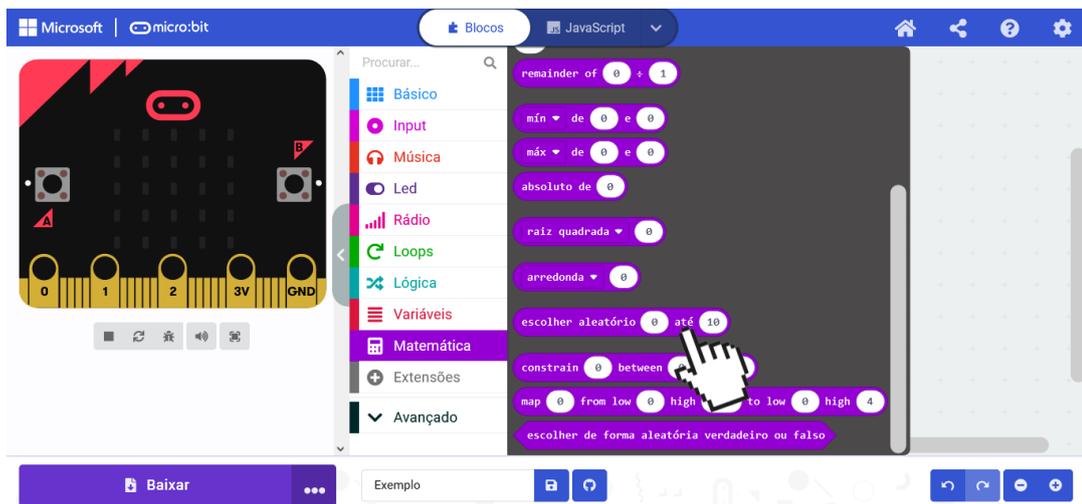


Figura 16.12: Selecionando um comando.



# Batata Quente

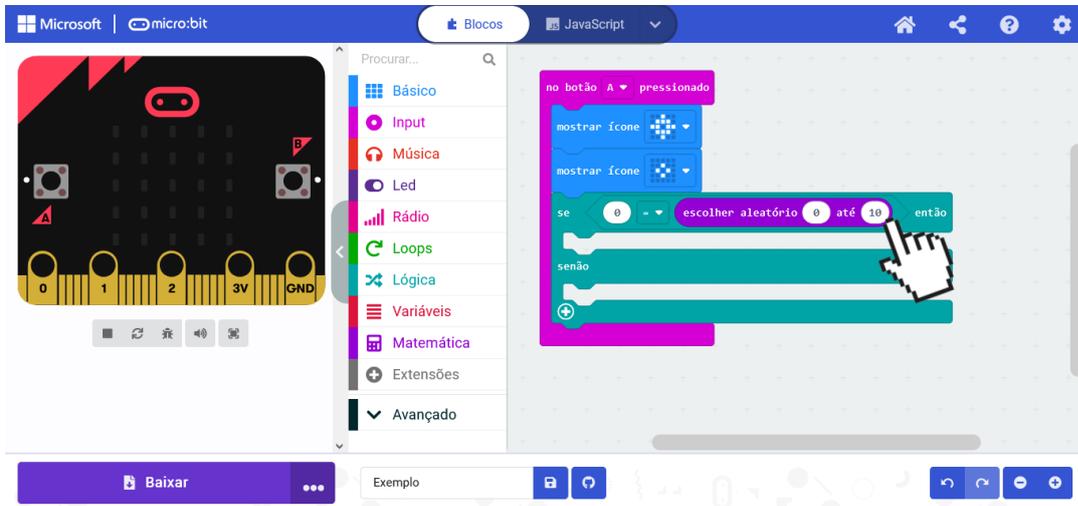


Figura 16.13: Montando um comando.

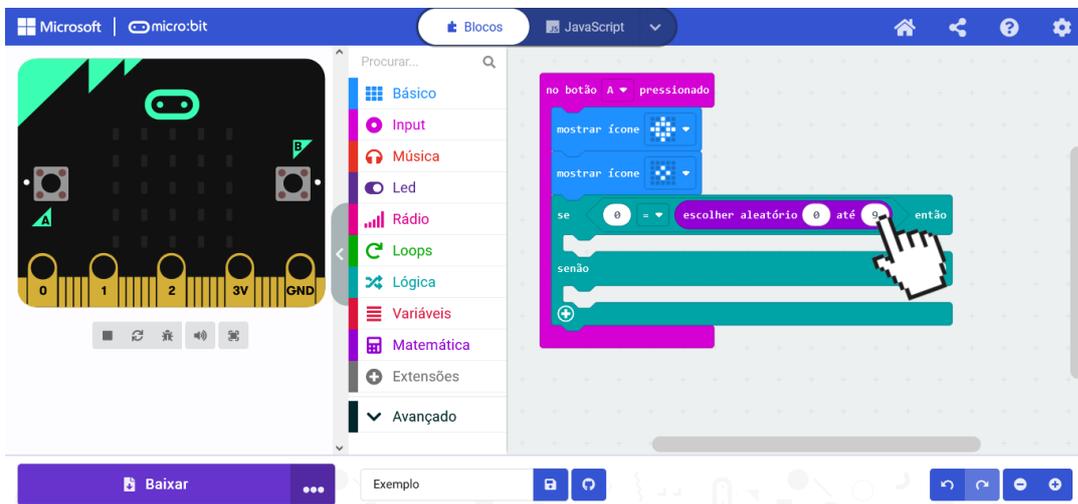


Figura 16.14: Alterando um intervalo de números aleatórios.

# Batata Quente

- Volte na aba **Básico**, selecione o comando **mostrar ícone** e coloque-o dentro do bloco **se 0 = escolher aleatório 0 até 9 então**, Figuras 16.15 e 16.16;

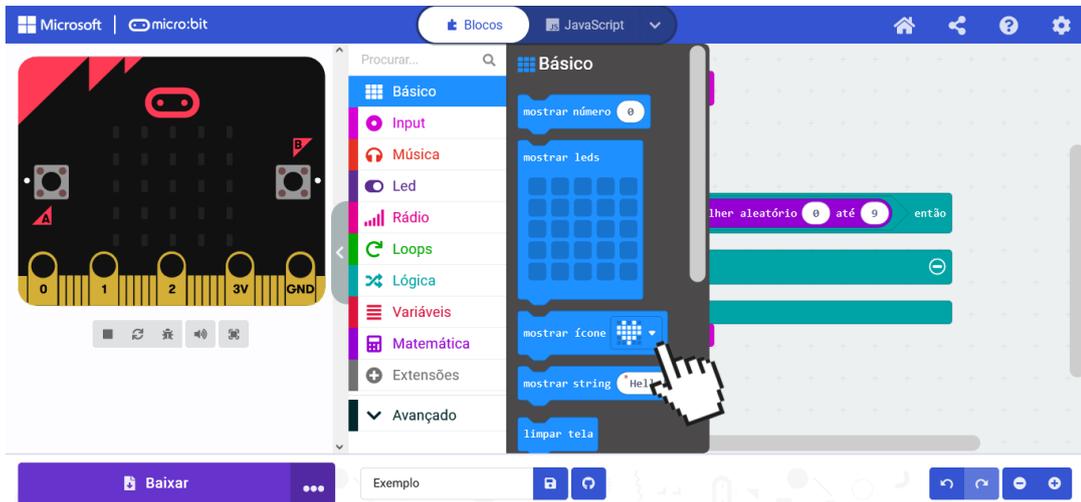


Figura 16.15: Selecionando um ícone.



Figura 16.16: Montando um comando.



# Batata Quente

- Escolha o ícone da **caveira**, para indicar que a batata queimou, Figura 16.17;

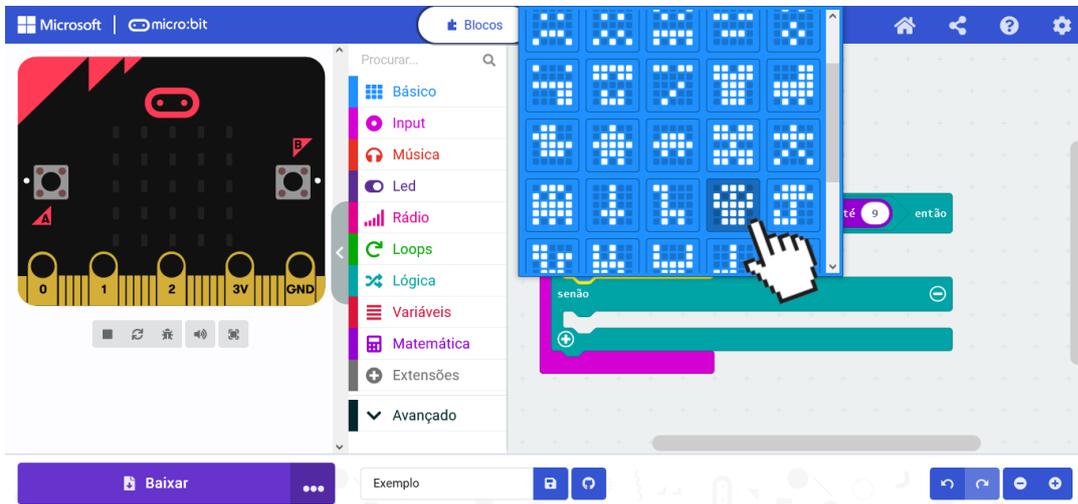


Figura 16.17: Alterando um ícone.

- Na aba **Música**, selecione o comando **iniciar melodia dadadum repetindo uma vez** e coloque-o abaixo do **mostrar ícone** com o desenho da **caveira**, Figuras 16.18 e 16.19;

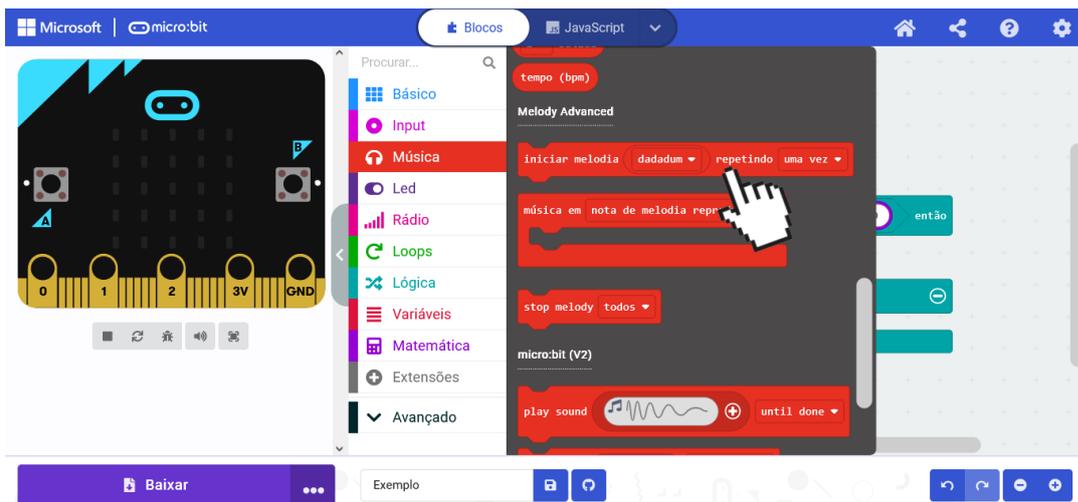


Figura 16.18: Selecionando um comando.

# Batata Quente

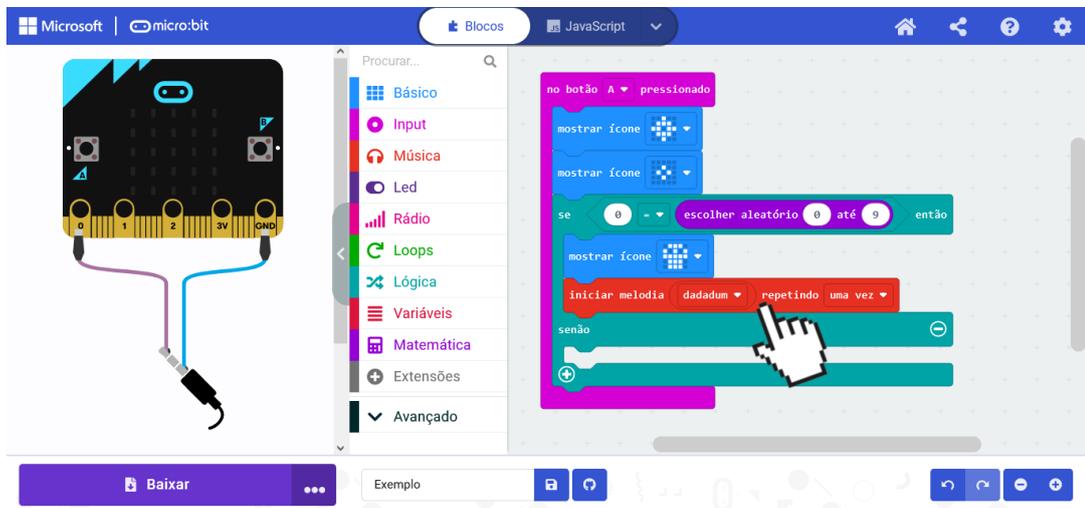


Figura 16.19: Montando um comando.

9. Em **Básico**, selecione novamente um comando **mostrar ícone** e coloque-o dentro do último bloco, abaixo de **senão**, Figuras 16.20 e 16.21;

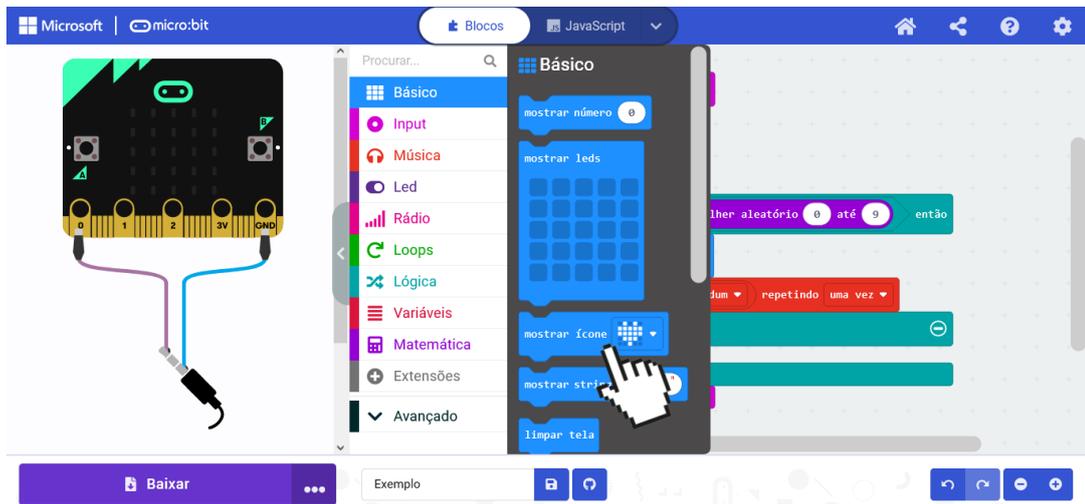


Figura 16.20: Selecionando um comando.



# Batata Quente

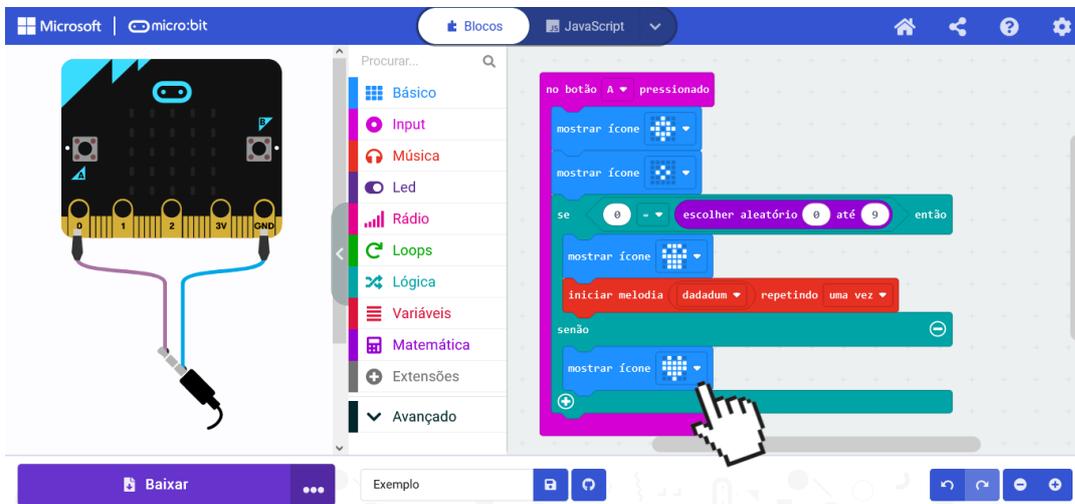


Figura 16.21: Montando um comando.

10. Altere o ícone para o **quadrado** que indicará que a batata ainda não queimou, Figura 16.22;

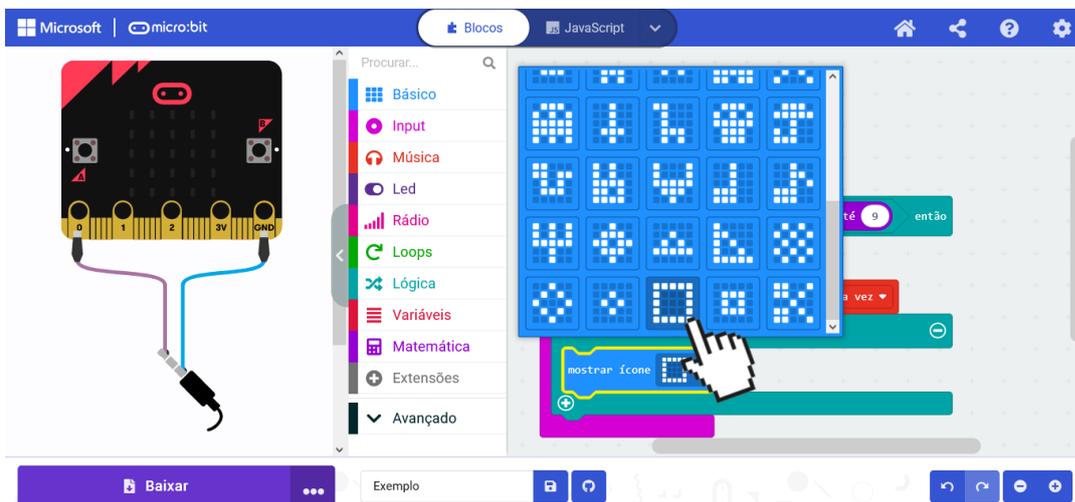


Figura 16.22: Alterando um ícone.

# Batata Quente

1. Na aba **Música**, selecione o comando **reproduzir tom C médio por 1 batida** e coloque-o logo abaixo de **mostrar ícone quadrado**, dentro do bloco **senão**, Figuras 16.23 e 16.24.

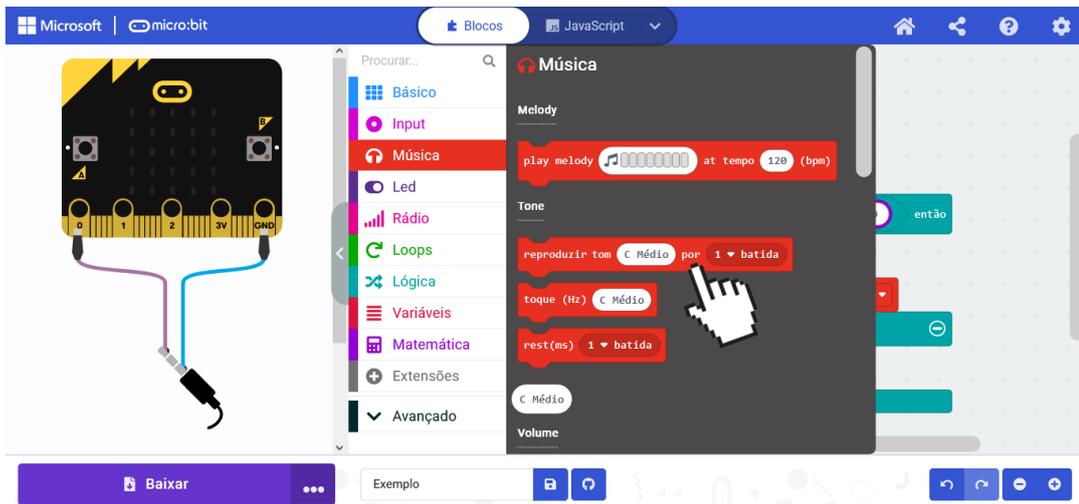
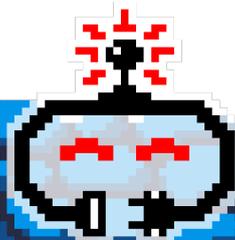


Figura 16.23: Selecionando um comando.



Figura 16.24: Montando um comando.



## Batata Quente

Dê um play na simulação, teste-a e está pronta a programação.

### Avançado

Avançado é uma nova parte da biblioteca, que contém várias abas que aumentam as possibilidades de projetos e facilitam a programação no micro:bit. E é nessa parte que aparece a aba jogos, que nos ajudará nas próximas programações.

### MICRO:DICA

Em nosso código, alteramos a melodia **dadadum** pela melodia **funeral**, clicando na seta para baixo, que fica à direita do nome da melodia. Antes de baixar o código para o micro:bit, teste as melodias no simulador e escolha a que mais gostar para deixar sua brincadeira mais divertida.



[Clique aqui para acessar o código.](#)

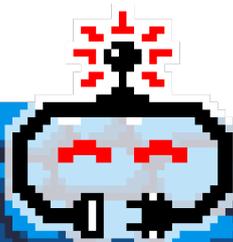


## Capítulo 17: Fantasma



Neste capítulo, a programação será feita para que uma imagem se “teletransporte” de um micro:bit para outro com apenas um movimento. Para que isso seja possível, será necessário escolher um número que o micro:bit receberá, como se fosse a estação de rádio em que ele está sintonizado, para que o teletransporte aconteça.

Caso esta atividade esteja sendo feita em grupos, cada grupo deve ter um número para evitar interferência e mostrar o ícone selecionado corretamente. Então, vamos começar!



# Fantasma

## Passo a passo:

1. Na aba **Básico**, selecione o bloco de acionamento **no iniciar**, como nas Figuras 17.1 e 17.2;

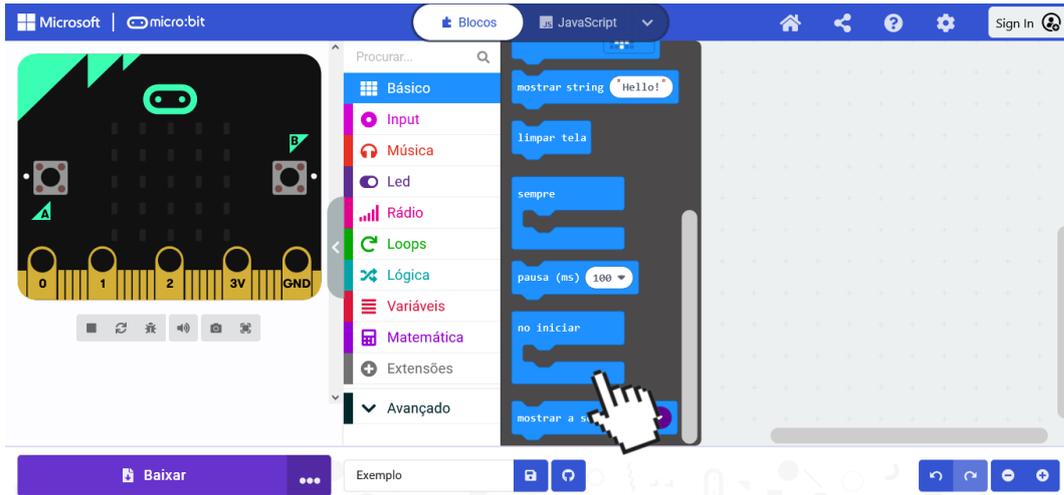


Figura 17.1: Selecionando um bloco de acionamento.

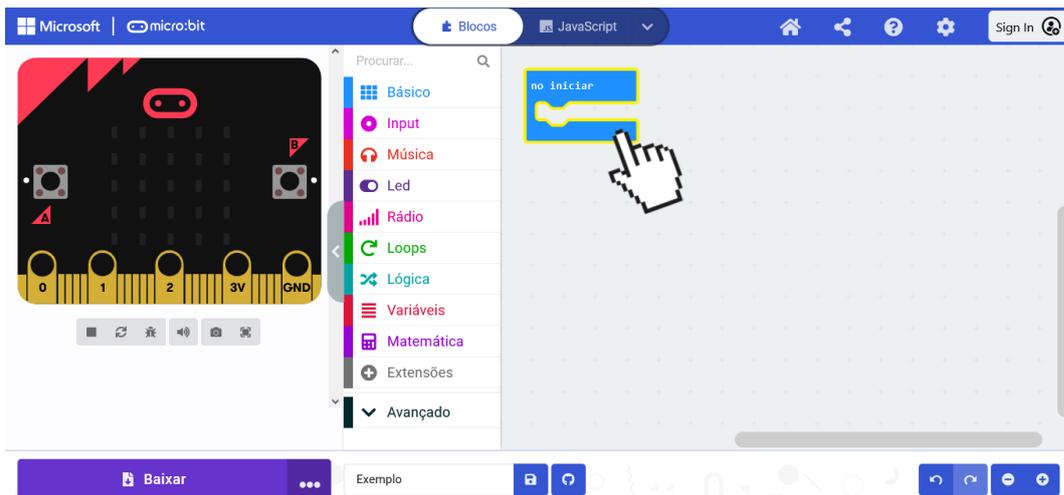


Figura 17.2: Montando um comando.

# Fantasma

2. Na aba **Rádio**, selecione o comando **definir grupo do rádio**. Altere o número 1 pelo número que quiser para representar seu grupo, Figura 17.3;

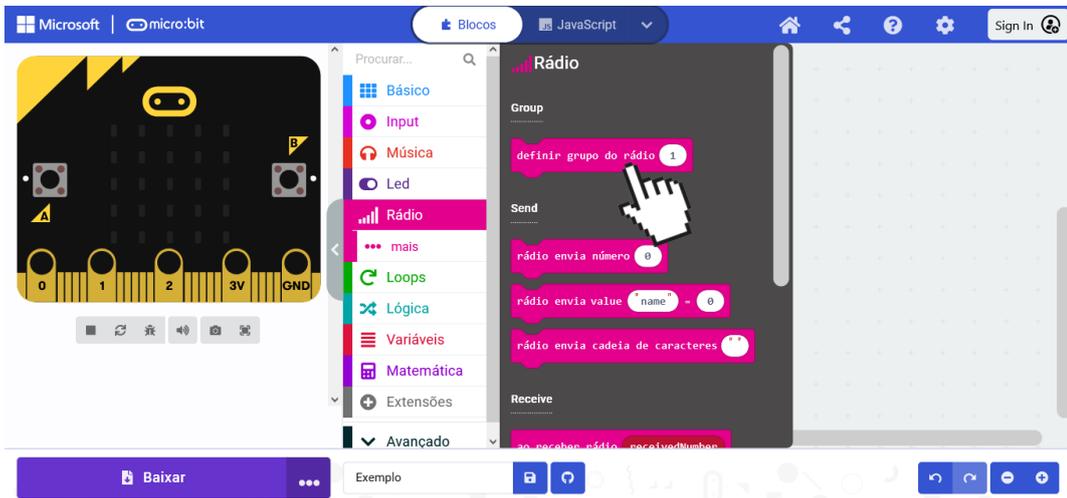


Figura 17.3: Selecionando um comando.

3. Insira o comando selecionado dentro do bloco de acionamento **no iniciar**, Figura 17.4;



Figura 17.4: Montando um comando.



# Fantasma

4. Na aba **Rádio**, selecione o bloco de acionamento **ao receber rádio receivedNumber**, Figuras 17.5 e 17.6;

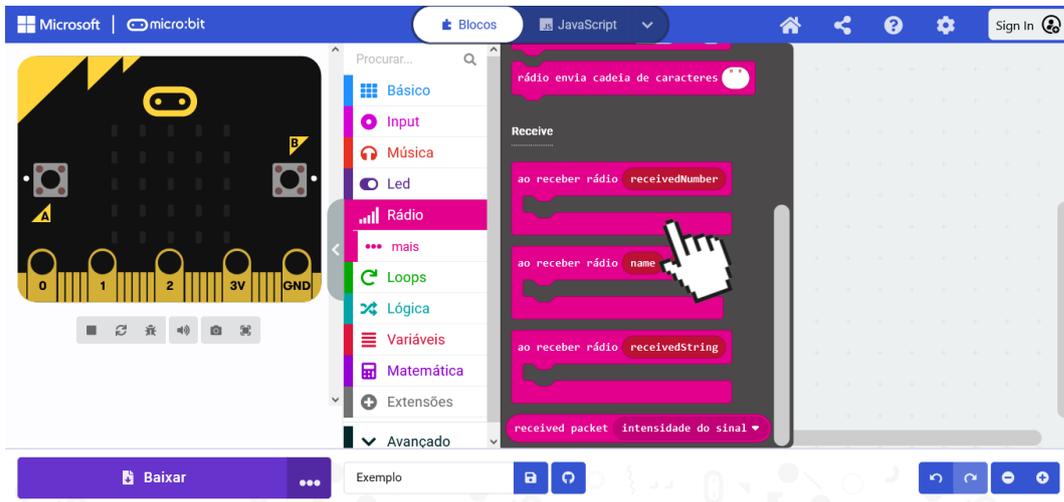


Figura 17.5: Selecionando um bloco de acionamento.

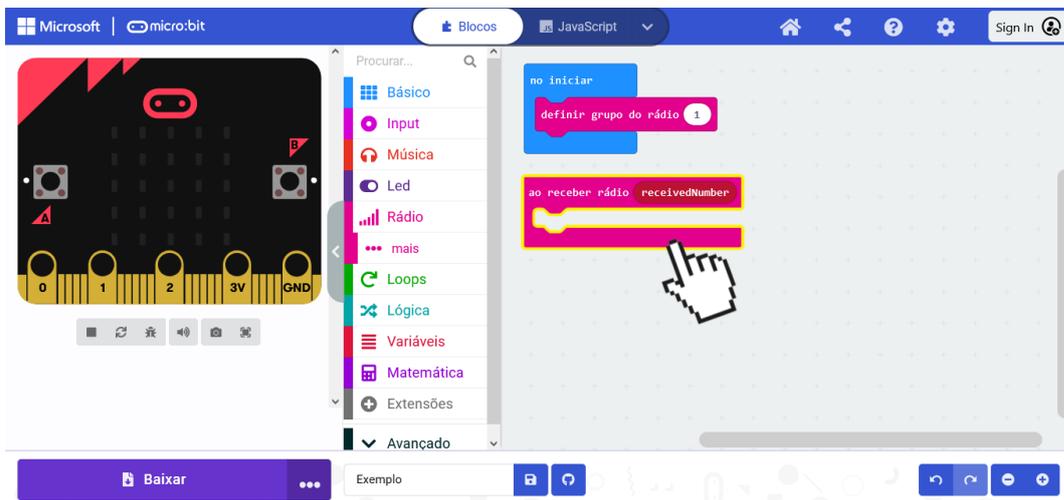


Figura 17.6: Montando um comando.

# Fantasma

5. Na aba **Básico**, selecione o comando **mostrar ícone**, Figura 17.7;



Figura 17.7: Selecionando um comando.

6. Insira o último comando selecionado dentro do bloco **ao receber rádio receivedNumber** e altere o ícone de **coração** para o de **fantasma**, Figuras 17.8 e 17.9;

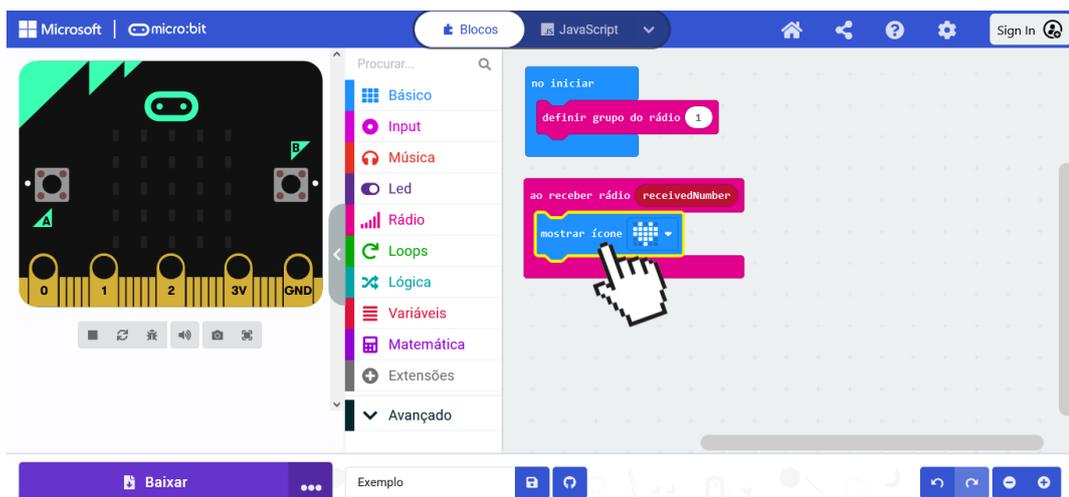


Figura 17.8: Montando um comando.



# Fantasma

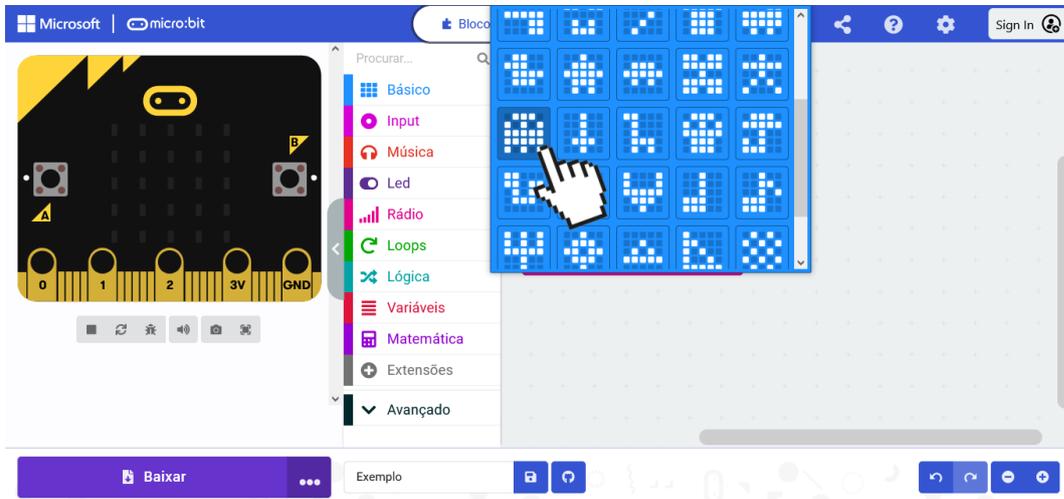


Figura 17.9: Alterando um ícone.

7. Na aba **Input**, selecione o bloco de acionamento **em agitar**, Figuras 17.10 e 17.11;



Figura 17.10: Selecionando um bloco de acionamento.

# Fantasma

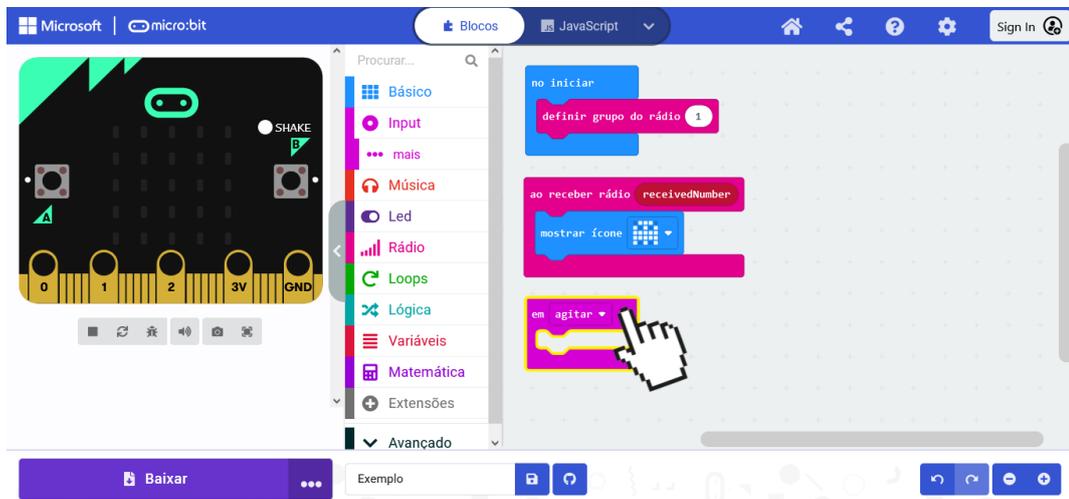


Figura 17.11: Montando um comando.

- Na aba **Básico**, selecione o comando **limpar tela**, Figura 17.12;

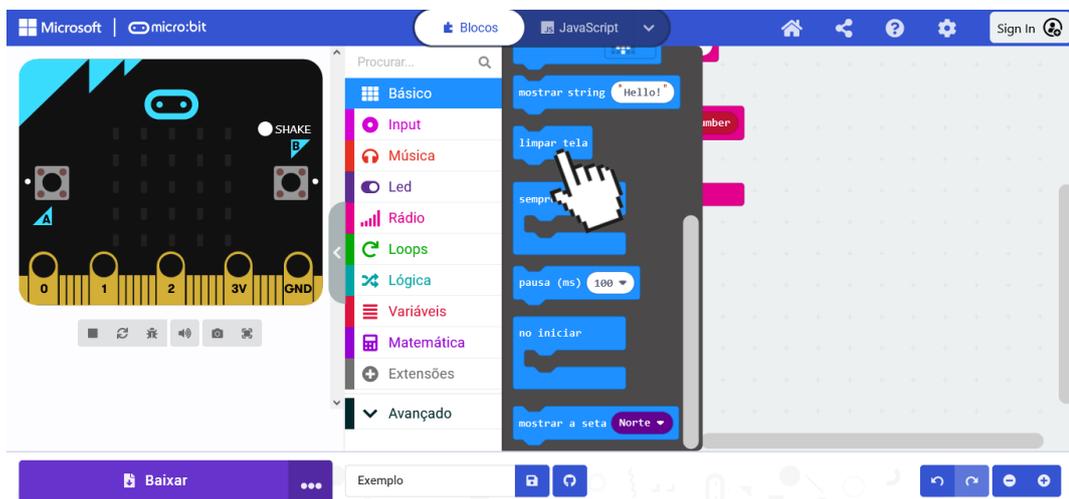
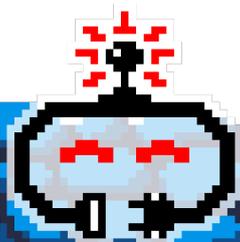


Figura 17.12: Selecionando um comando.



# Fantasma

9. Insira o último comando selecionado dentro do bloco **em agitar**, Figura 17.13;

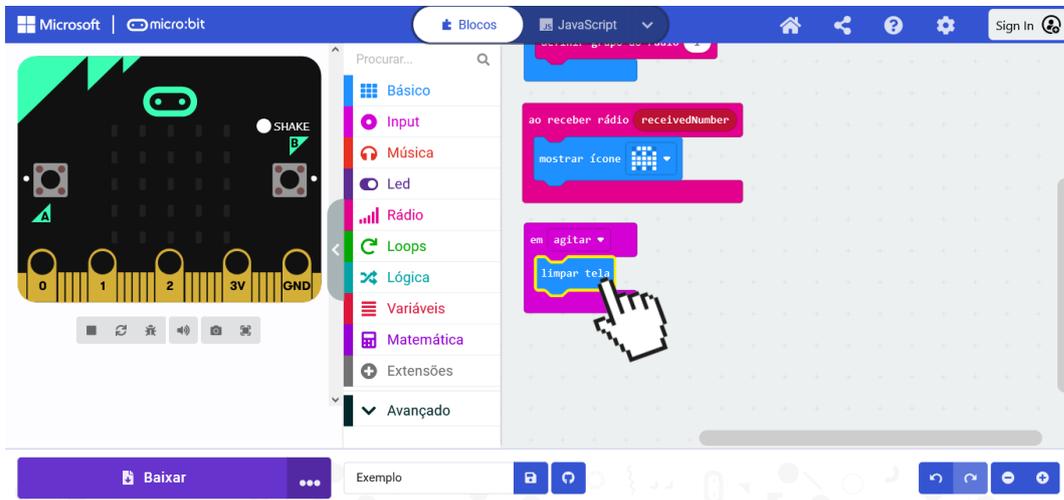


Figura 17.13: Montando um comando.

10. Na aba **Rádio**, selecione o comando **rádio envia número**, Figura 17.14;



Figura 17.14: Selecionando um comando.

# Fantasma

- 11.** Insira o último comando selecionado logo abaixo de **limpar tela**, Figura 17.15;

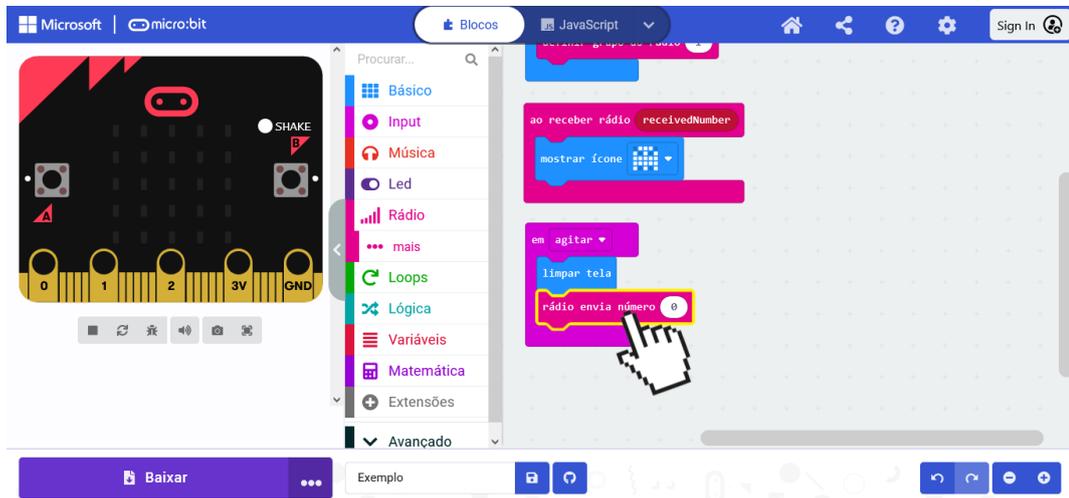


Figura 17.15: Montando um comando.

- 12.** Altere o número **zero** para o número definido no **Passo 2**. Neste exemplo, o número foi alterado para **1**, Figura 17.16.

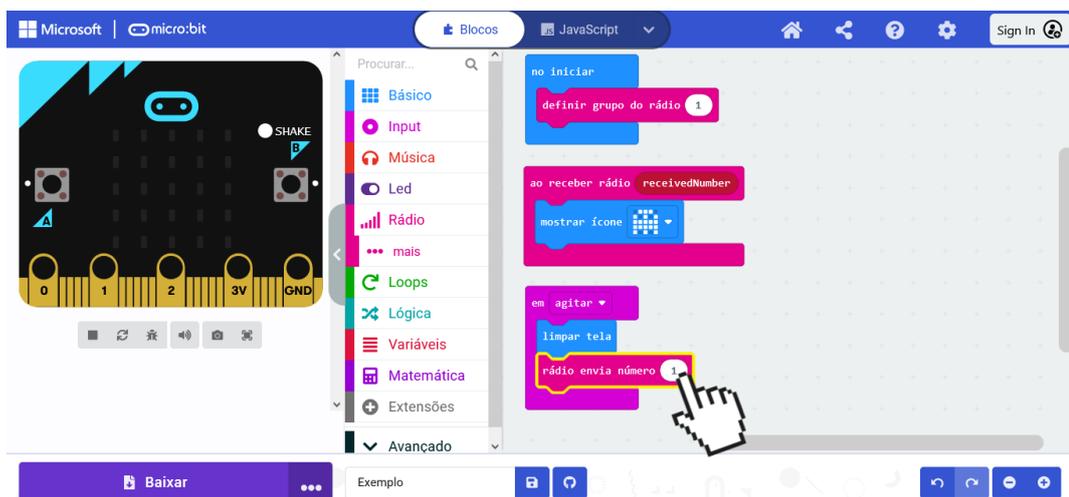


Figura 17.16: Alterando o número do rádio.



## Fantasma

Agora, é só baixar o programa para o micro:bit e ver o teletransporte acontecer quando a placa é agitada.

### MICRO:DICA

É possível enviar frases também. Basta trocar o [mostrar ícone](#) por [mostrar string](#) e escrever a frase que deseja enviar.



[Clique aqui para acessar o código.](#)



# Capítulo 18: Extensões



As extensões são formas de expandir as possibilidades de utilização do micro:bit, levando-o além das bibliotecas de funções básicas e possibilitando que se utilize novos blocos com as mais diversas funcionalidades.

Uma das extensões que mais usaremos é a do nosso carrinho Maqueen.

## Maqueen

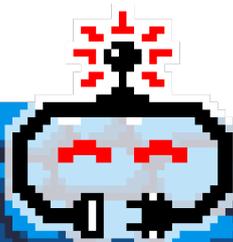
Maqueen é um robô que interage com o micro:bit e foi criado pela empresa DFRO-BOT. Ele é um carro que possui sensor ultrassônico, LEDs, buzzer, sensores de luz, entre outros. O carro é controlado pelo micro:bit e funciona com pilhas.

Vamos aprender como usar as extensões com o exemplo de uma régua.

## Régua

O que é uma régua? Já usou ou sabe usá-la?

A régua é uma ferramenta utilizada para traçar retas e medir distâncias. É possível programar o micro:bit para ser utilizado como uma régua.



# Extensões

## Passo a passo:

1. Clique na aba **Extensões** e busque por **Maqueen** na barra de pesquisa, como nas Figuras 18.1, 18.2, 18.3 e 18.4;

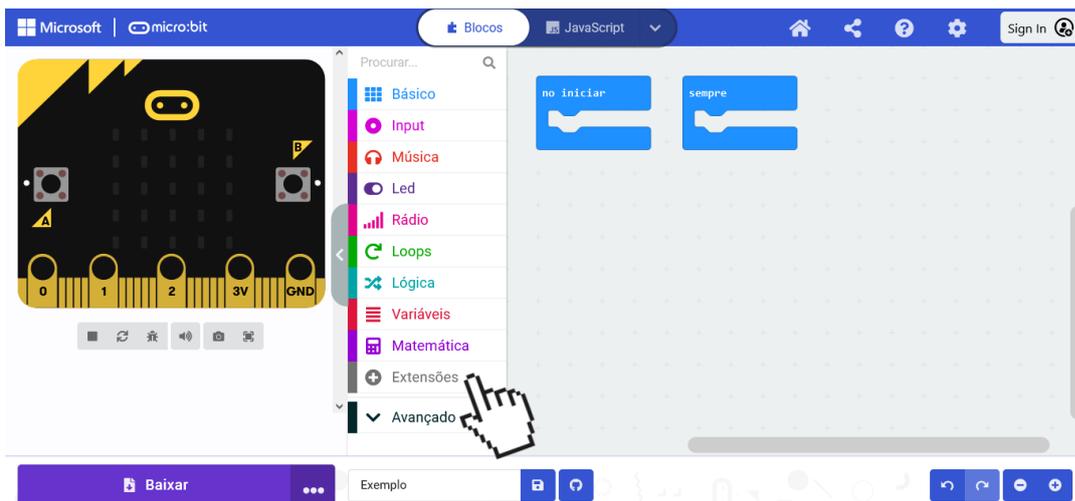


Figura 18.1: Selecionando uma biblioteca.

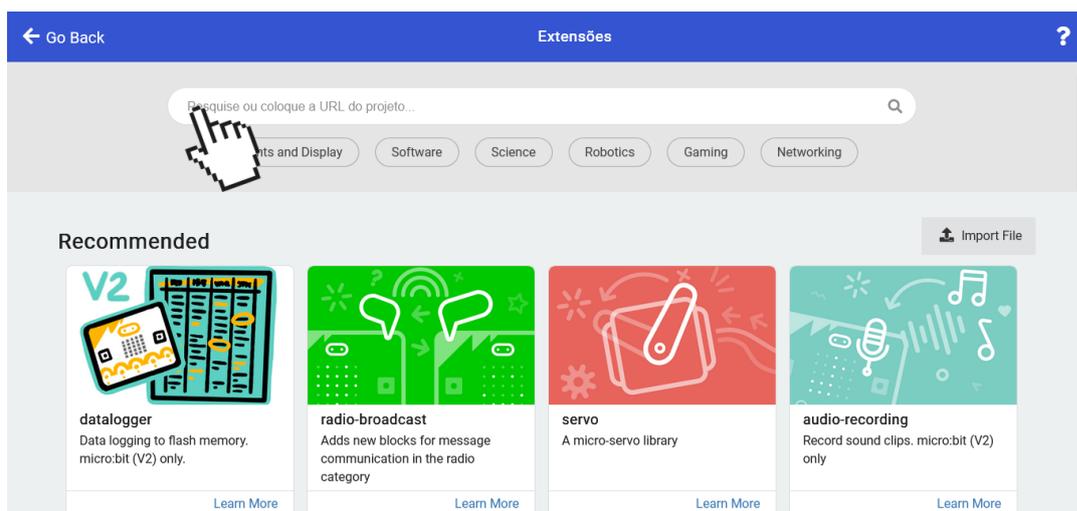


Figura 18.2: Pesquisando uma extensão.

# Extensões

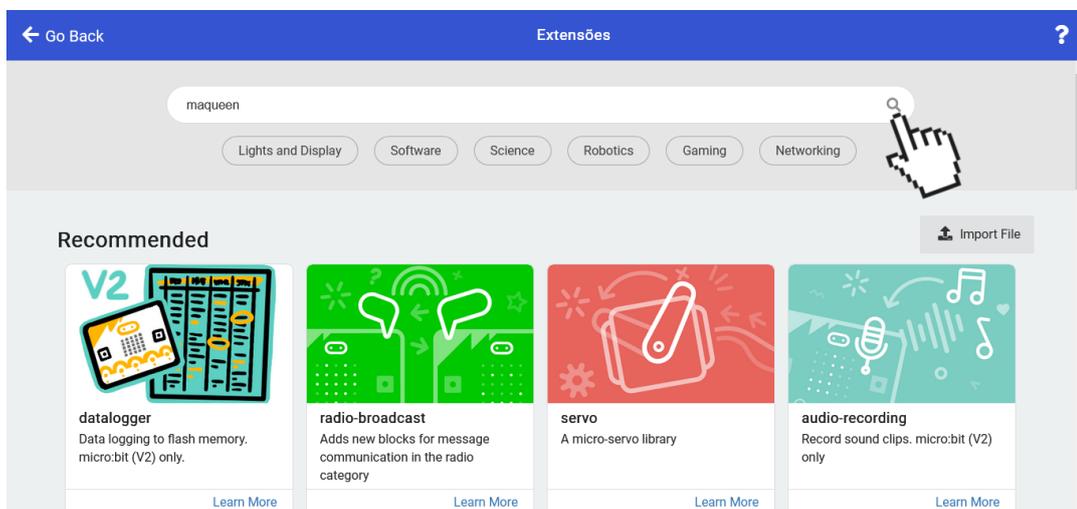


Figura 18.3: Buscando por uma extensão.

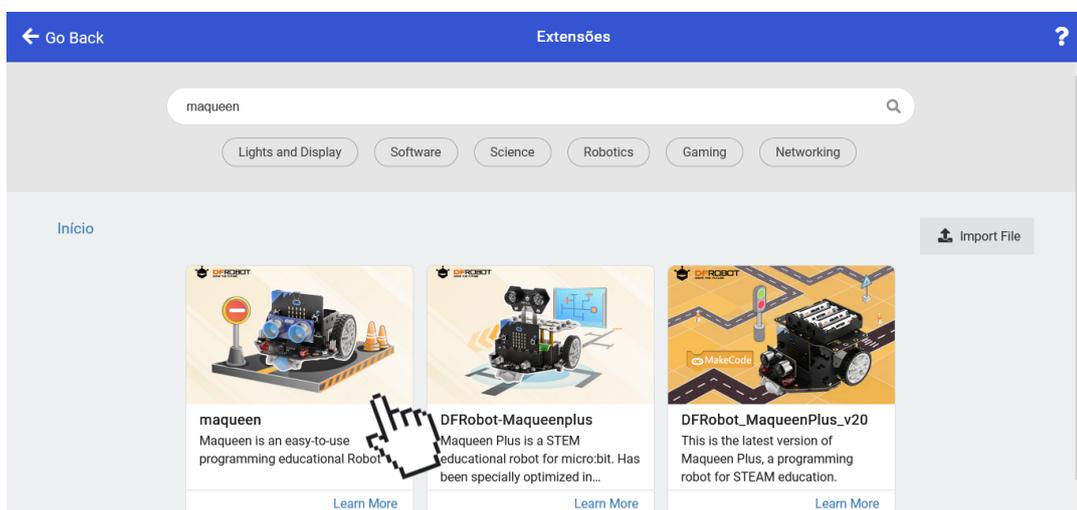
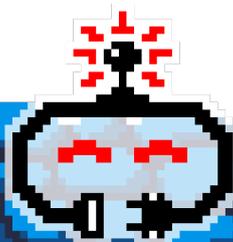


Figura 18.4: Adicionando uma extensão.



# Extensões

2. Perceba que, logo abaixo da aba **Led**, apareceram duas novas abas **Maqueen** e **IR**, Figura 18.5;



Figura 18.5: Novas bibliotecas.

3. Em **Básico**, selecione o comando **mostrar número 0**, e coloque-o dentro do bloco de acionamento **sempre**, Figuras 18.6 e 18.7;



Figura 18.6: selecionando um comando.

# Extensões

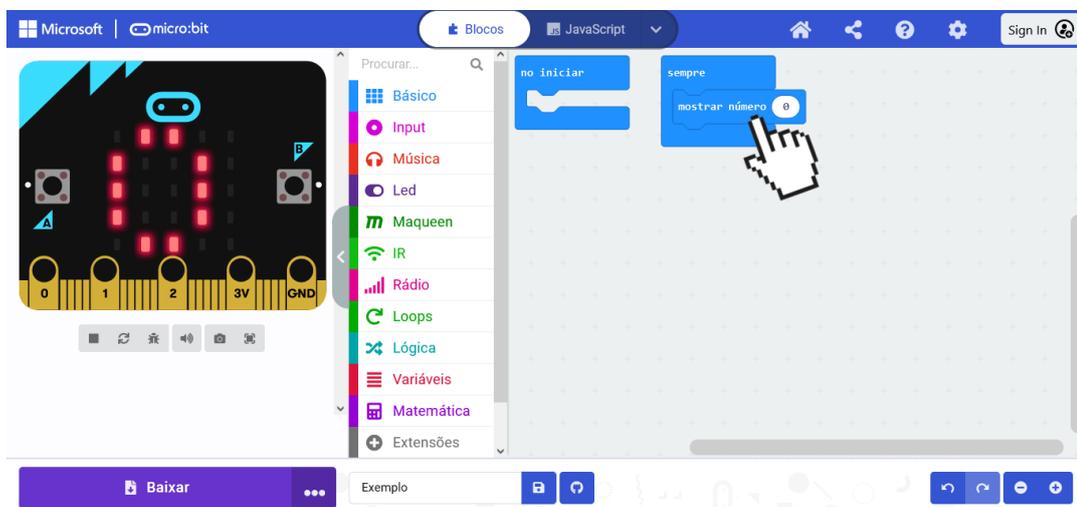
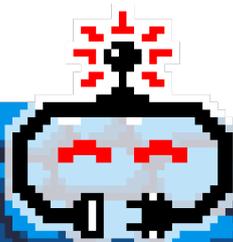


Figura 18.7: Montando um comando.

4. Na nova biblioteca **Maqueen**, selecione o comando **distância em unidades cm** e coloque-o no lugar do **0** em **mostrar número 0**, Figuras 18.8 e 18.9.



Figura 18.8: Selecionando um comando.



# Extensões



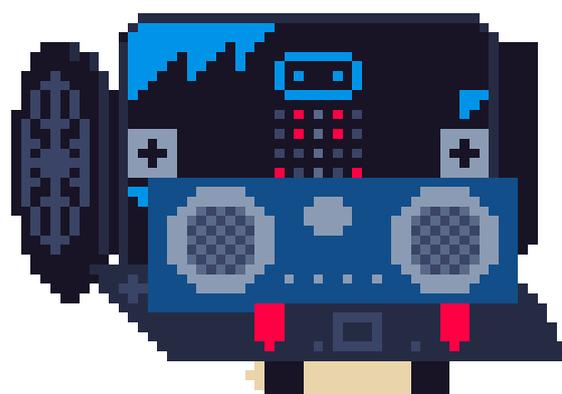
Figura 18.9: Montando um comando.

**Baixe o código para o micro:bit e encaixe a placa no carrinho para medir distâncias.**

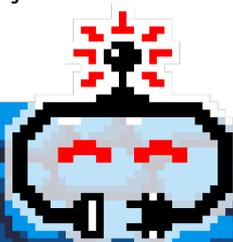
[Clique aqui para acessar o código.](#)



# Capítulo 19: Carro controlado por luz



Imagine como seria brincar com um carro guiado pela luz. A melhor parte é que não precisa só ficar na imaginação. Com o micro:bit isso é possível!



# Carro controlado por luz

## Passo a passo:

1. Para programar esse carro é necessário adicionar uma nova biblioteca de funções. Para isso, clique sobre a aba **Extensões** e busque por **Maqueen** na barra de pesquisa, como nas Figuras 19.1, 19.2 e 19.3;

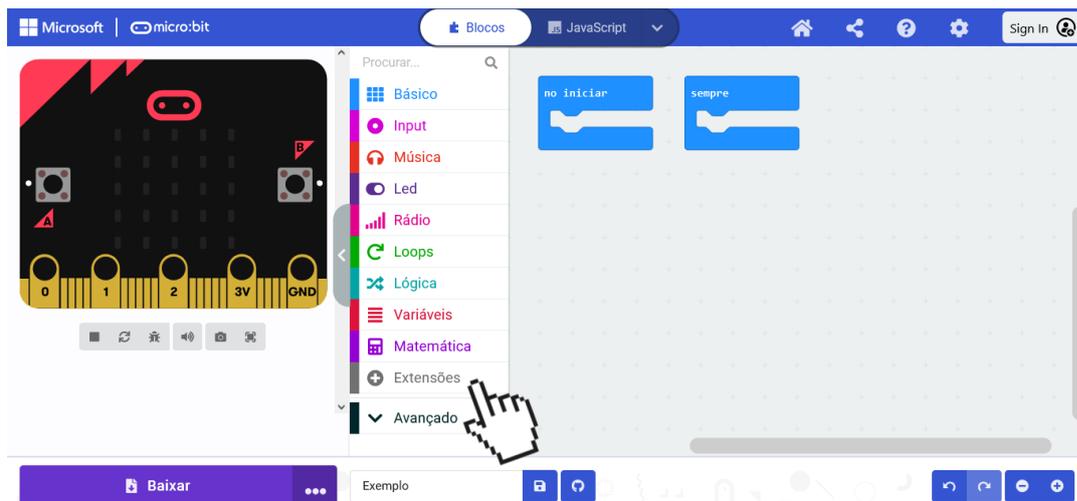


Figura 19.1: Adicionando uma extensão.

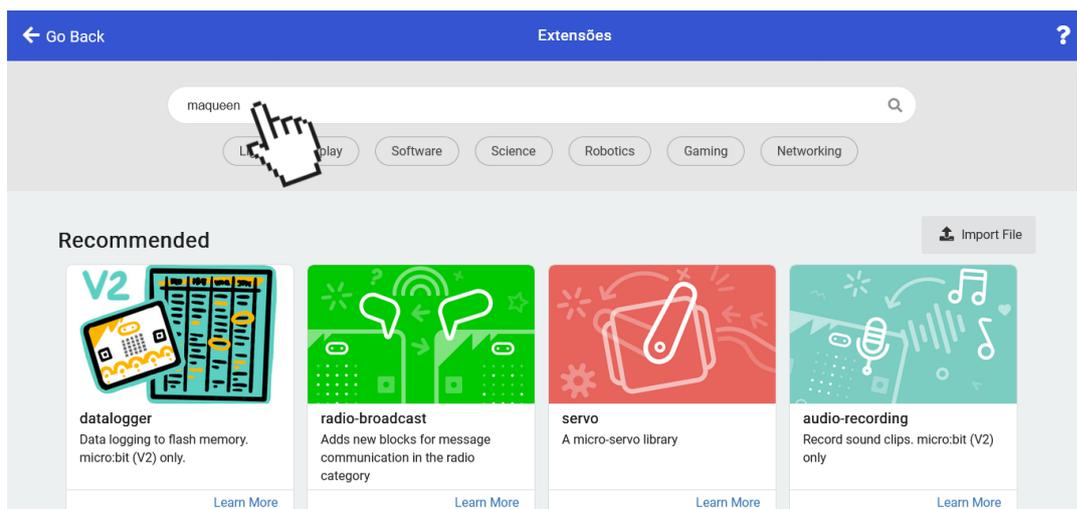


Figura 19.2: Pesquisando uma extensão.

# Carro controlado por luz

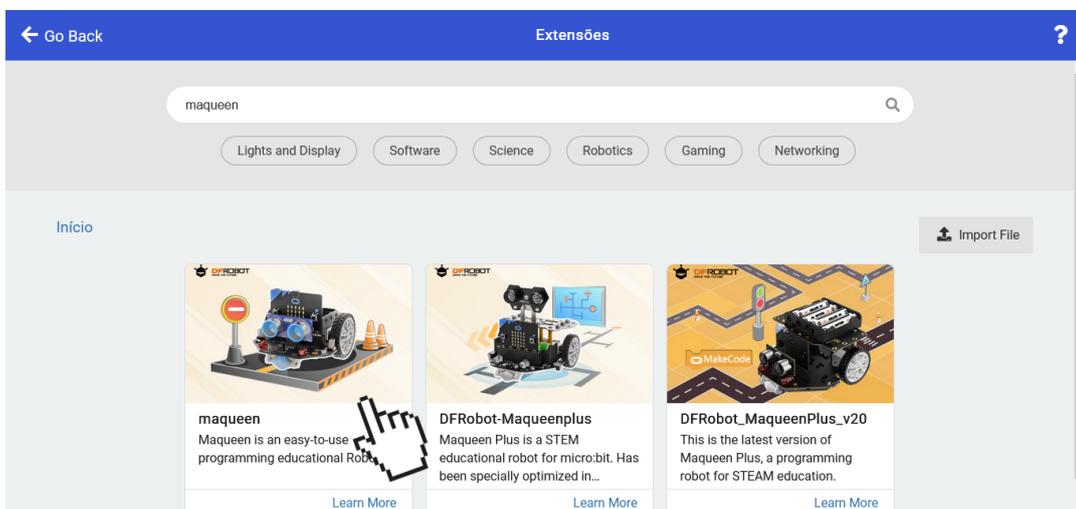


Figura 19.3: Selecionando uma extensão.

2. Em **Lógica**, selecione o bloco de comando **se verdadeiro então, senão** e coloque-o dentro do bloco de acionamento **sempre**, Figuras 19.4 e 19.5;

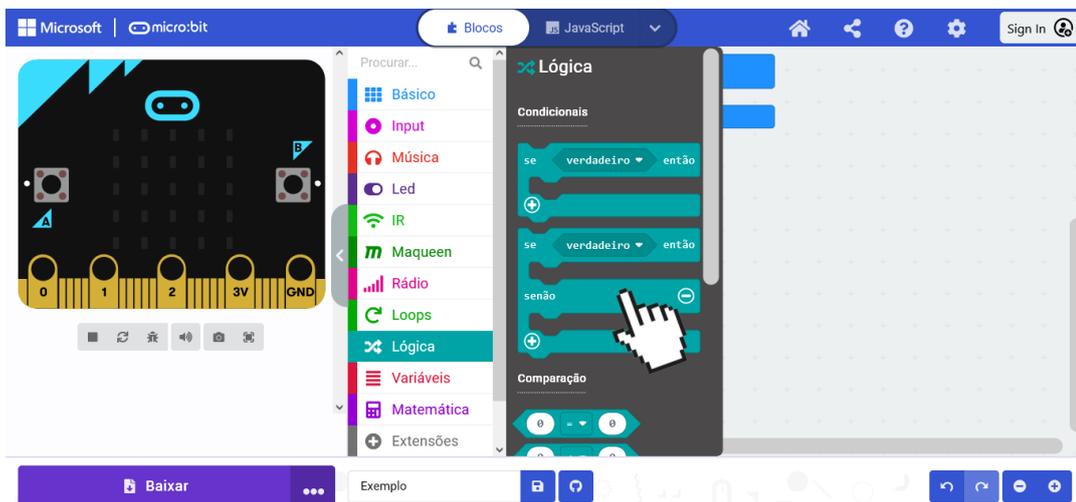
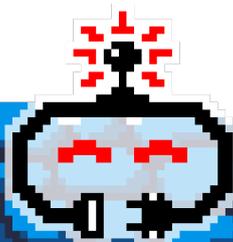


Figura 19.4: Selecionando um comando.



# Carro controlado por luz



Figura 19.5: Montando um comando.

3. Novamente em **Lógica**, selecione uma comparação  $0 < 0$  e coloque-a sobre o **verdadeiro**, Figura 19.6;

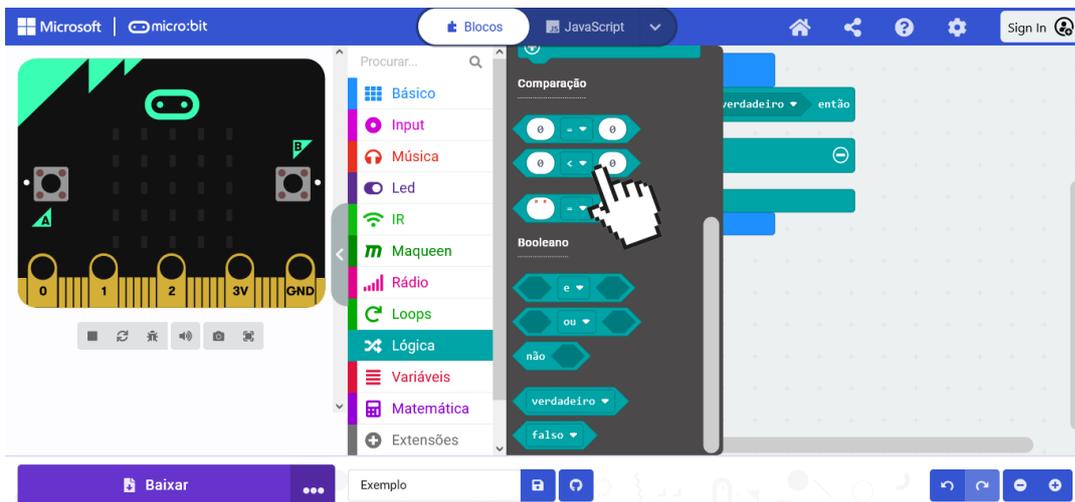


Figura 19.6: Selecionando uma comparação.

# Carro controlado por luz

4. Na comparação, selecione a opção que representa o sinal de maior,  $0 > 0$ , como na Figura 19.7;



Figura 19.7: Alterando um sinal de comparação.

5. Ainda na comparação, altere o segundo 0 por 100, para que fique com  $0 > 100$ , como na Figura 19.8;

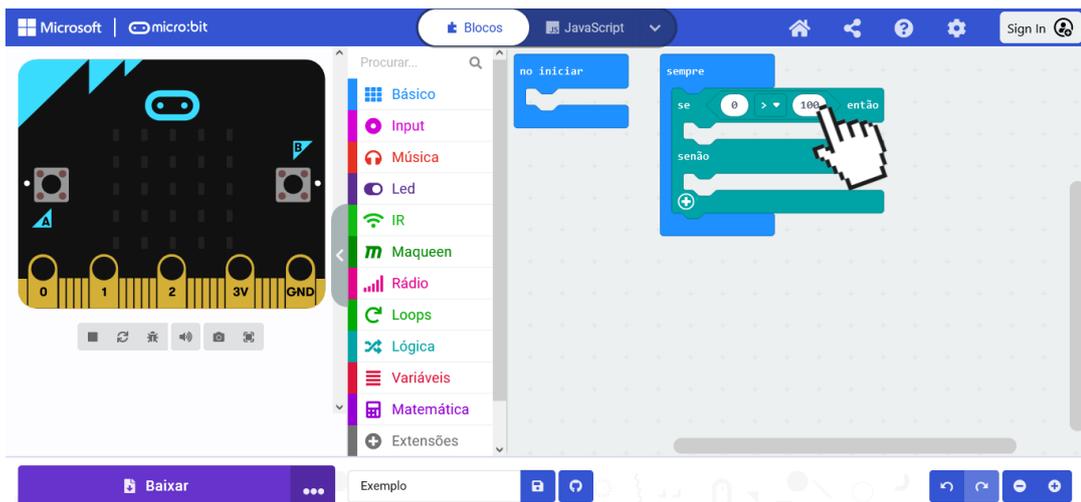
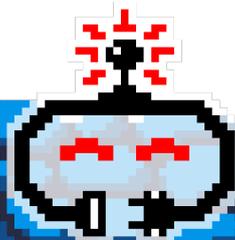


Figura 19.8: Alterando uma comparação.



# Carro controlado por luz

- Em **Input**, selecione o comando **nível de luz** e coloque-o no lugar do **0** na comparação de **0 > 100**, como nas Figuras 19.9 e 19.10;



Figura 19.9: Selecionando um comando.



Figura 19.10: Montando uma comparação.

# Carro controlado por luz

7. Na aba **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0** e insira-o dentro do bloco de acionamento **se nível de luz > 100 então**, Figuras 19.11 e 19.12;

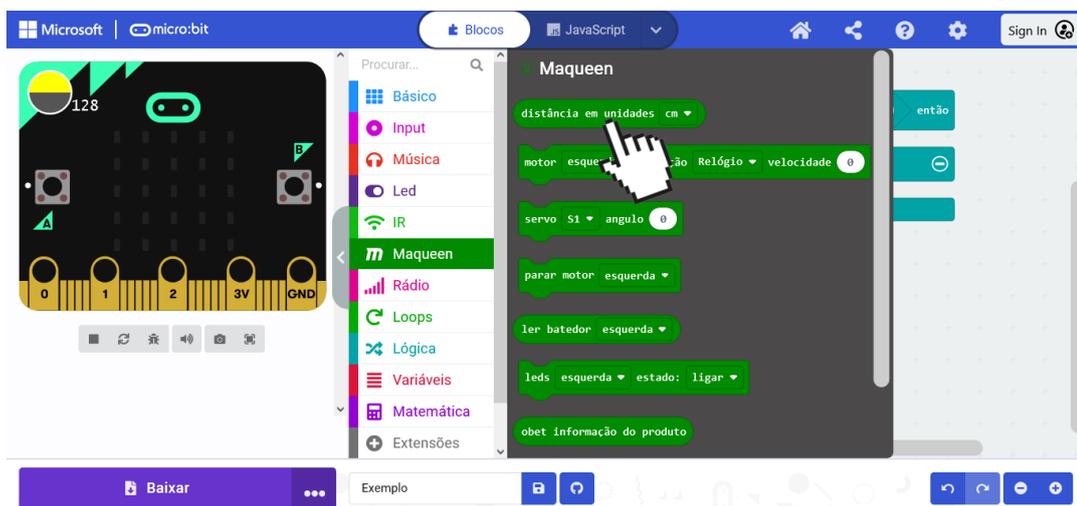
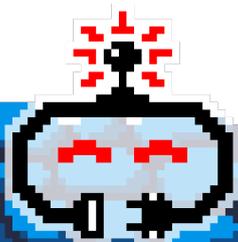


Figura 19.11: Selecionando um comando.



Figura 19.12: Montando um comando.



# Carro controlado por luz

- Altere o número **0** da velocidade do motor para **255**, como na Figura 19.13;

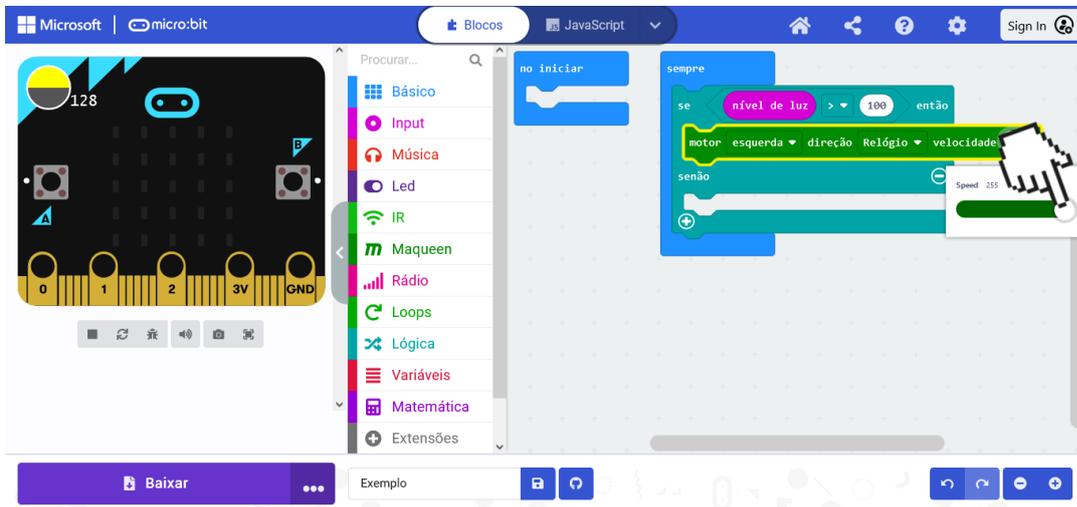


Figura 19.13: Alterando uma velocidade.

- Repita os três últimos passos e altere o motor da **esquerda** para **direita**, Figuras 19.14 e 19.15;



Figura 19.14: Alterando um motor.

# Carro controlado por luz



Figura 19.15: Alterando uma velocidade.

10. Novamente na biblioteca **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0**, Figura 19.16;

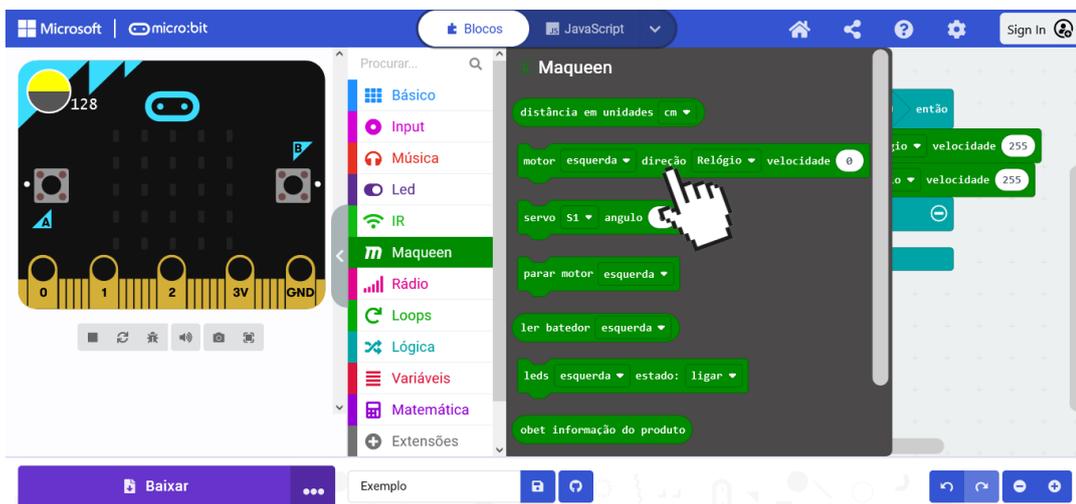
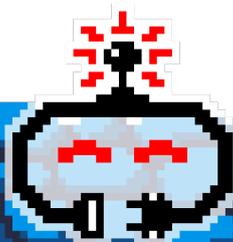


Figura 19.16: Selecionando um comando.



# Carro controlado por luz

- 11.** Coloque o último comando selecionado dentro do bloco **senão**, Figura 19.17;



Figura 19.17: Montando um comando.

- 12.** Selecione outra vez, na aba **Maqueen**, o comando **motor esquerda direção Relógio velocidade 0**, Figura 19.18, e coloque-o abaixo do último comando montando;

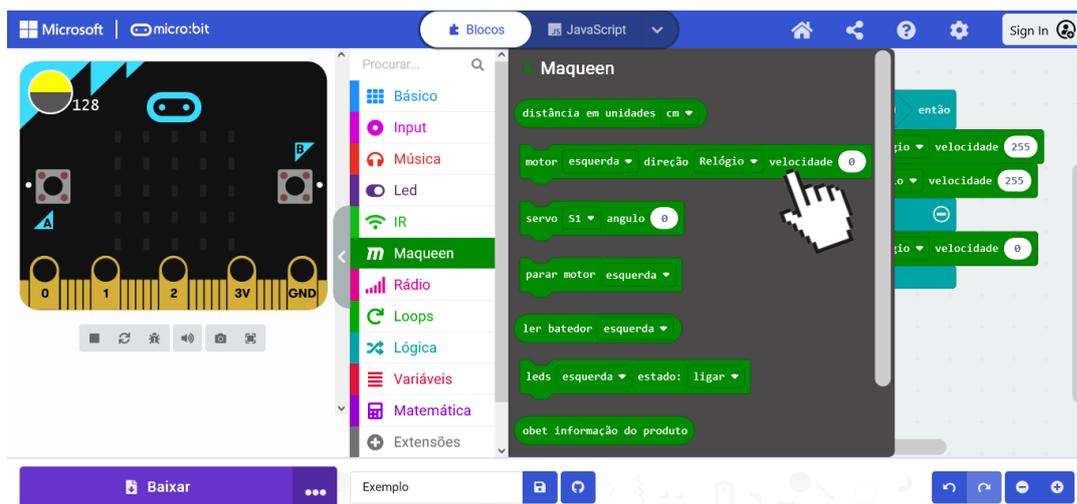


Figura 19.18: Selecionando um comando.

# Carro controlado por luz

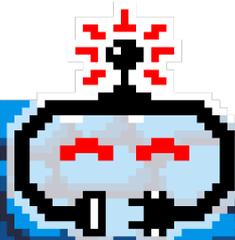
**13.** Altere o motor **esquerda** para **direita** e o código está pronto.



Figura 19.19: Alterando um motor.

**Baixe o programa para o micro:bit e encaixe a placa no Maqueen para vê-lo andar quando há luz no ambiente.**

[Click aqui para acessar o código.](#)





Carro controlado por  
luz



# Capítulo 20: Carro que desvia



Nesta atividade, aprenda a fazer um carro que desvia sozinho dos obstáculos à sua frente, sem qualquer interferência de controle ou luzes.



# Carro que desvia

## Passo a passo:

1. Em **Extensões**, clique na barra de pesquisa e busque por **Maqueen**, Figuras 20.1 e 20.2;

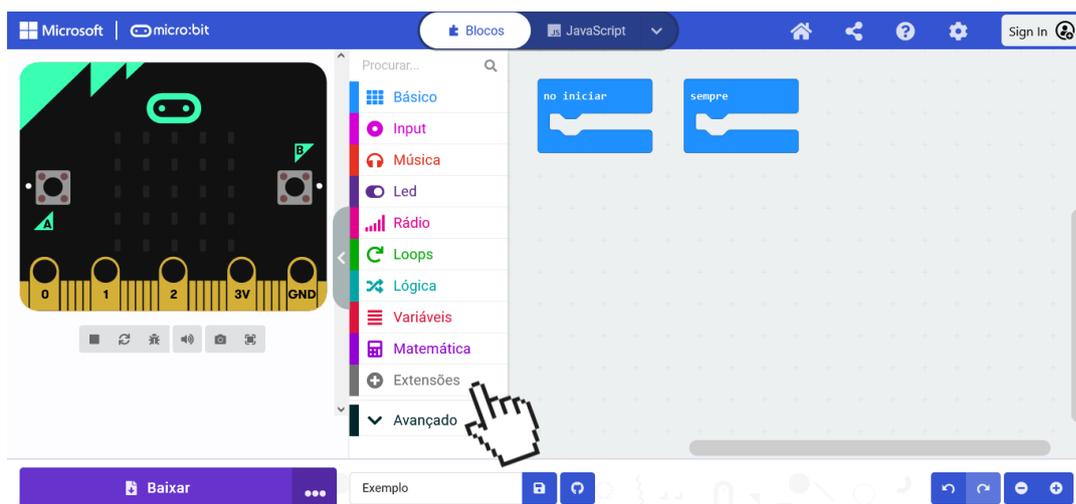


Figura 20.1: Selecionando uma extensão.

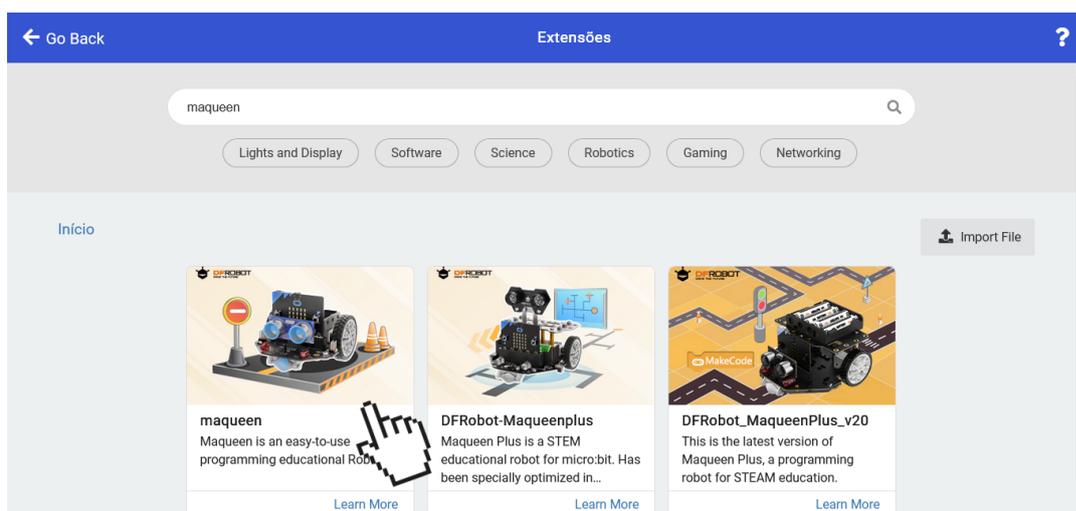


Figura 20.2: Adicionando uma extensão.

# Carro que desvia

2. Em **Lógica**, selecione o bloco **se verdadeiro então, senão** e coloque-o dentro do bloco de acionamento **sempre**, Figura 20.3;

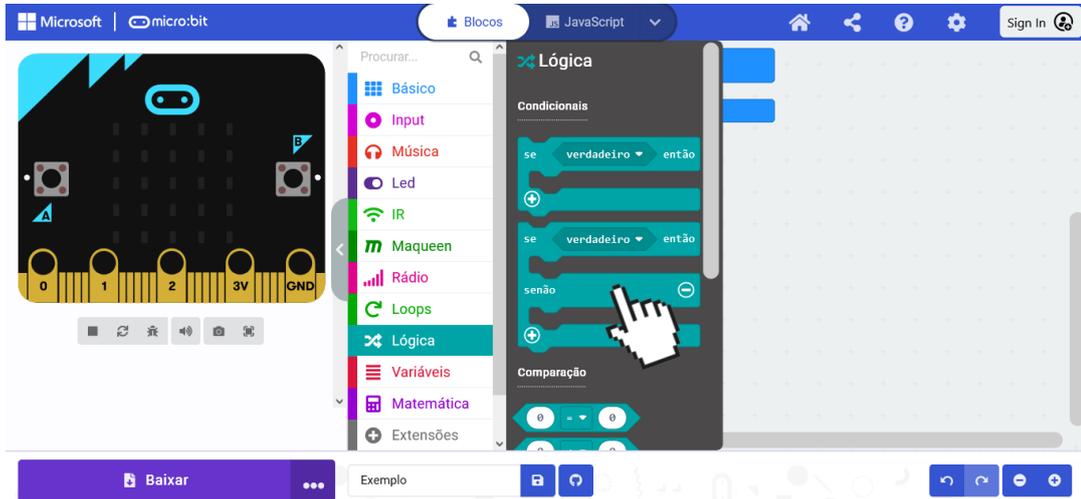


Figura 20.3: Selecionando um bloco.

3. Na aba **Lógica**, selecione o condicional Booleano **e**, Figura 20.4;

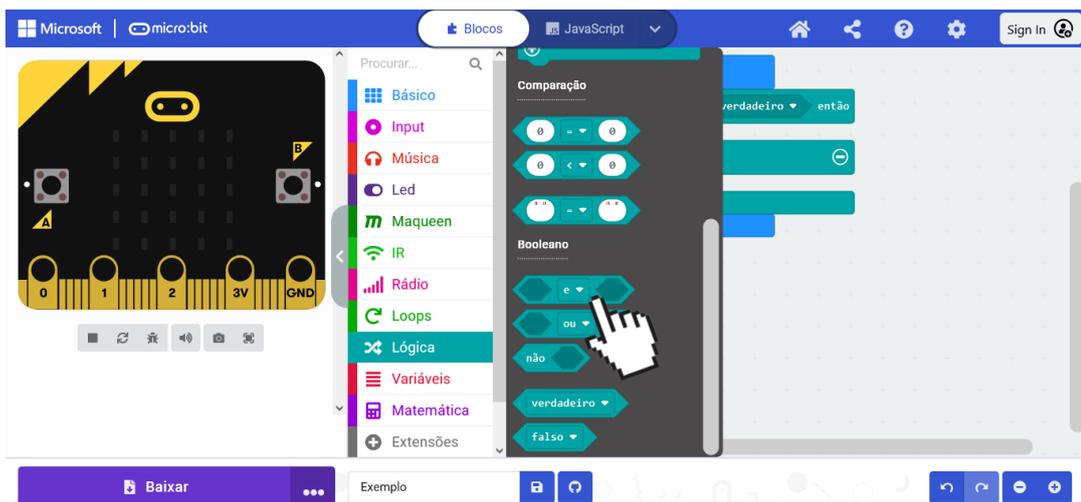
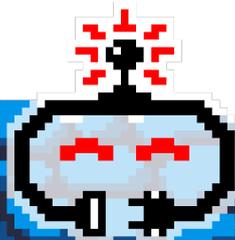


Figura 20.4: Selecionando um condicional booleano.



# Carro que desvia

4. Insira o condicional booleano **e** no lugar do verdadeiro no bloco **se verdadeiro então**, Figura 20.5;



Figura 20.5: Montando uma condição.

5. Na aba **Lógica**, selecione a comparação **0 = 0** duas vezes e coloque-as dentro dos espaços vazios no condicional booleano **e**, Figuras 20.6 e 20.7;

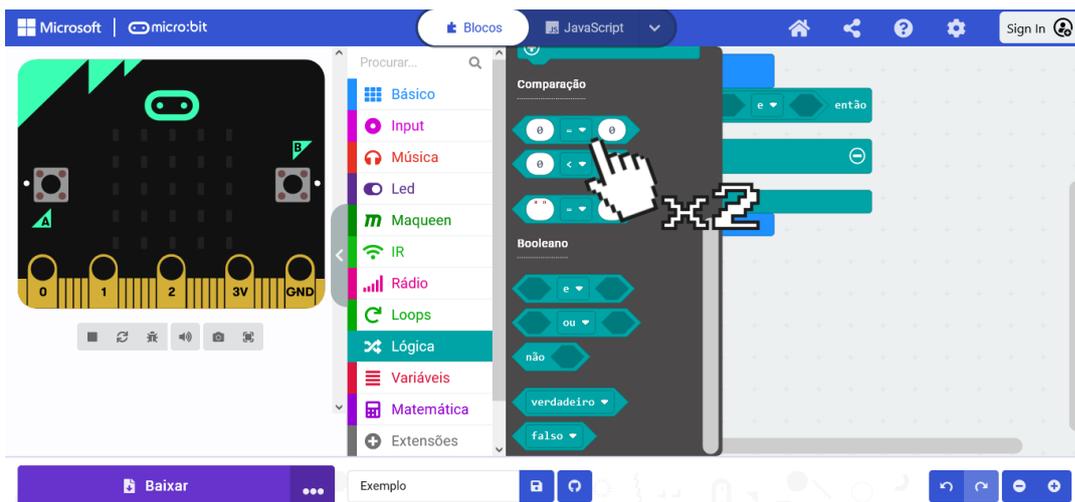


Figura 20.6: Selecionando comparações.

# Carro que desvia

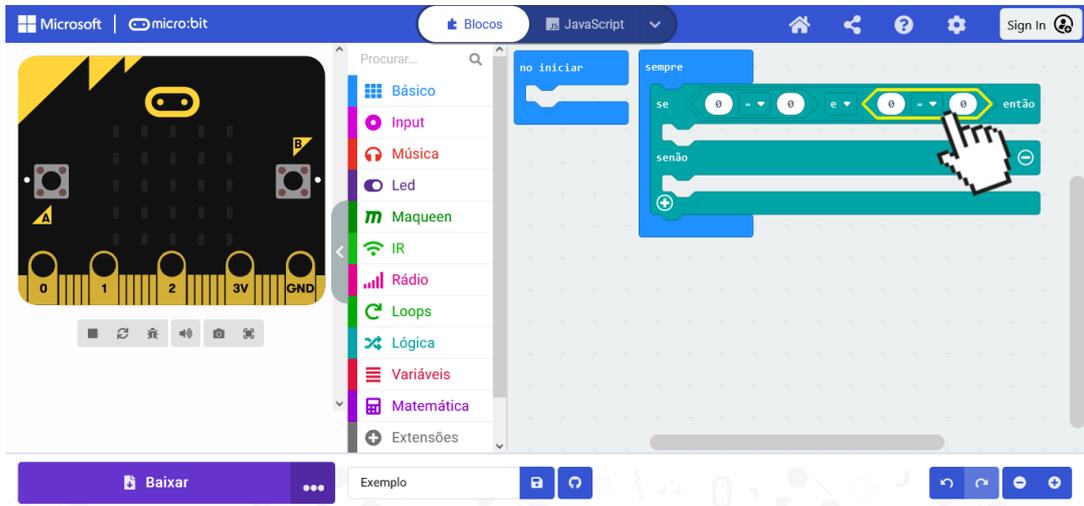


Figura 20.7: Montando comparações.

6. Vá na aba **Maqueen** e pegue o comando **distância em unidades cm**, como na Figura 20.8;

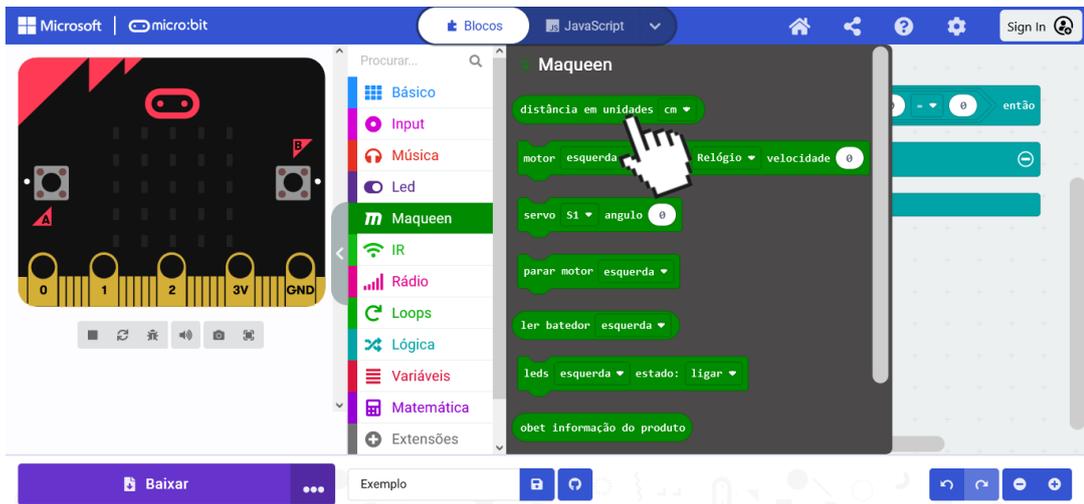


Figura 20.8: Selecionando um comando.



# Carro que desvia

7. Coloque o comando selecionado dentro do primeiro **zero** da comparação **0 = 0**, como na Figura 20.9;

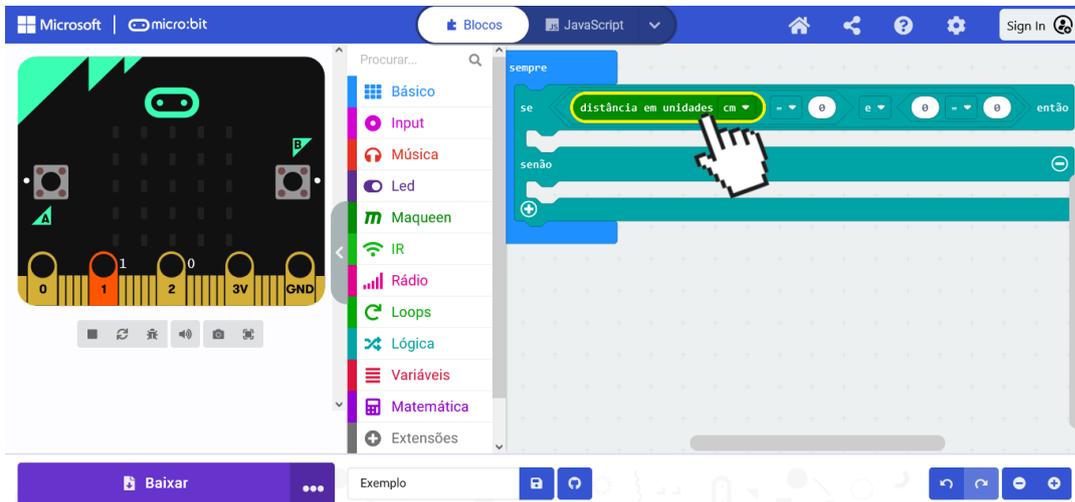


Figura 20.9: Montando uma comparação.

8. No comando **distância em unidades cm = 0**, troque o sinal de igualdade, **=**, para o sinal de menor ou igual, **≤**. Depois, altere o número **0** por **20**, como nas Figuras 20.10 e 20.11;



Figura 20.10: Mudando uma comparação.

# Carro que desvia

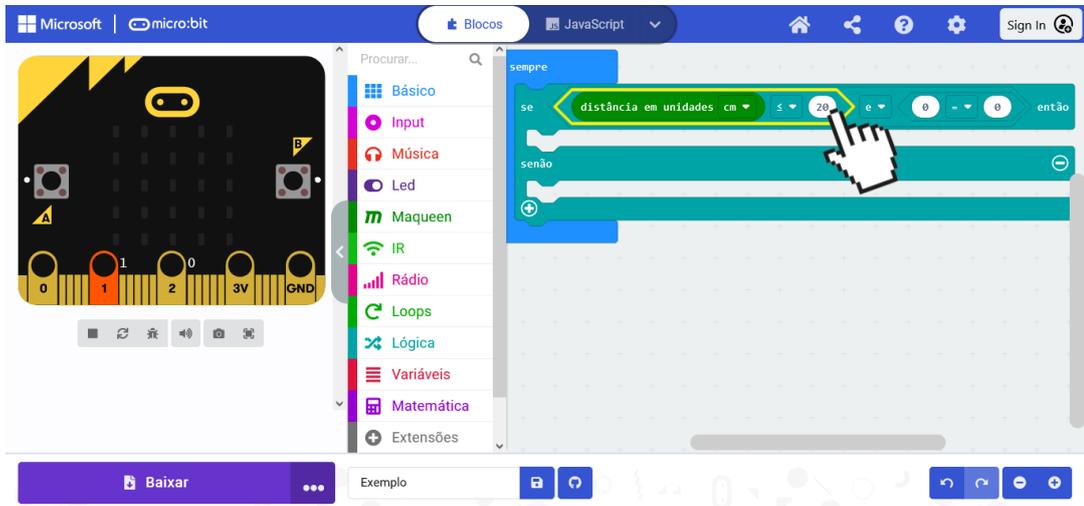


Figura 20.11: Alterando um valor comparado.

9. Novamente na aba **Maqueen**, pegue o comando **distância em unidades cm**, Figura 20.12;

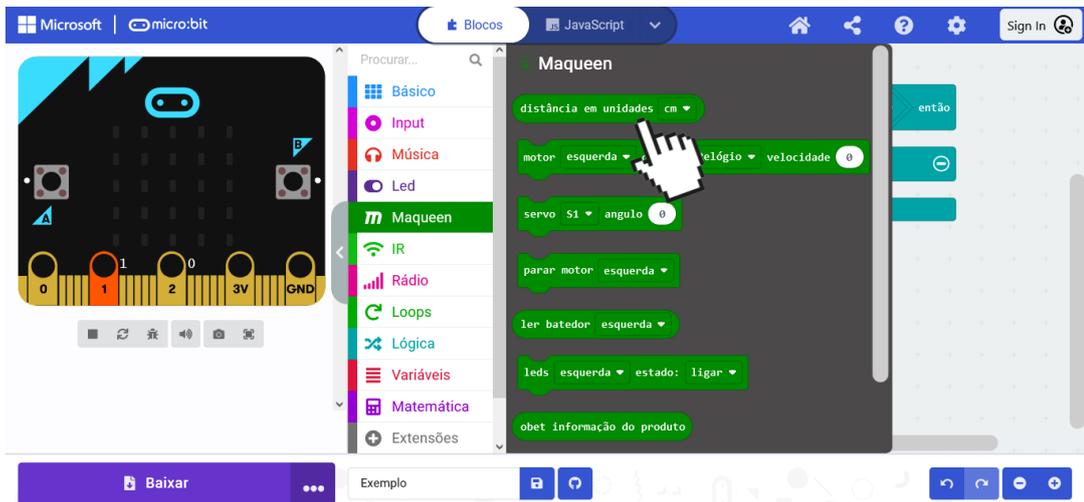


Figura 20.12: Selecionando um comando.



# Carro que desvia

10. Coloque o comando selecionado dentro do primeiro **zero** da segunda comparação **0 = 0**, como na Figura 20.13;

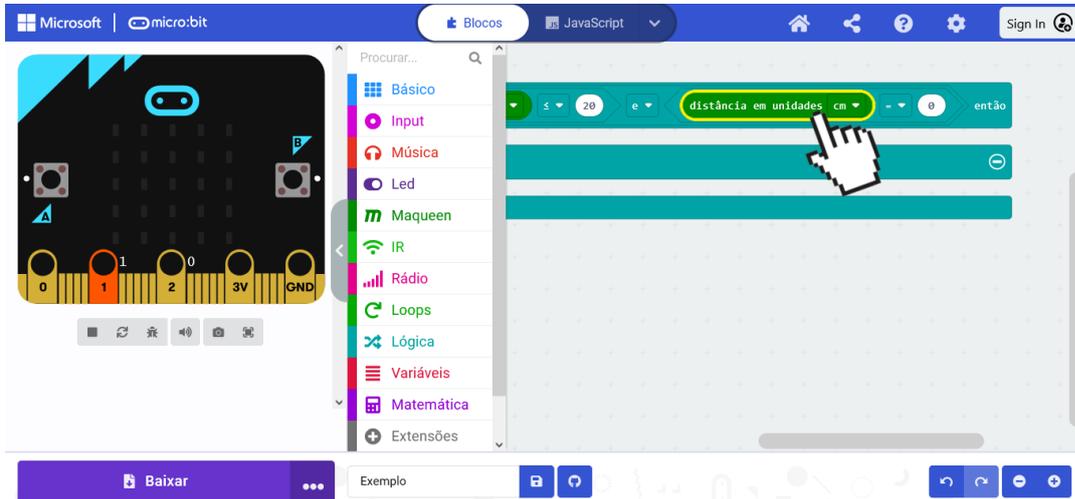


Figura 20.13: Montando uma comparação.

11. Em **distância em unidades cm = 0**, troque o sinal de igual, =, para o de diferente,  $\neq$ , como na Figura 20.14;

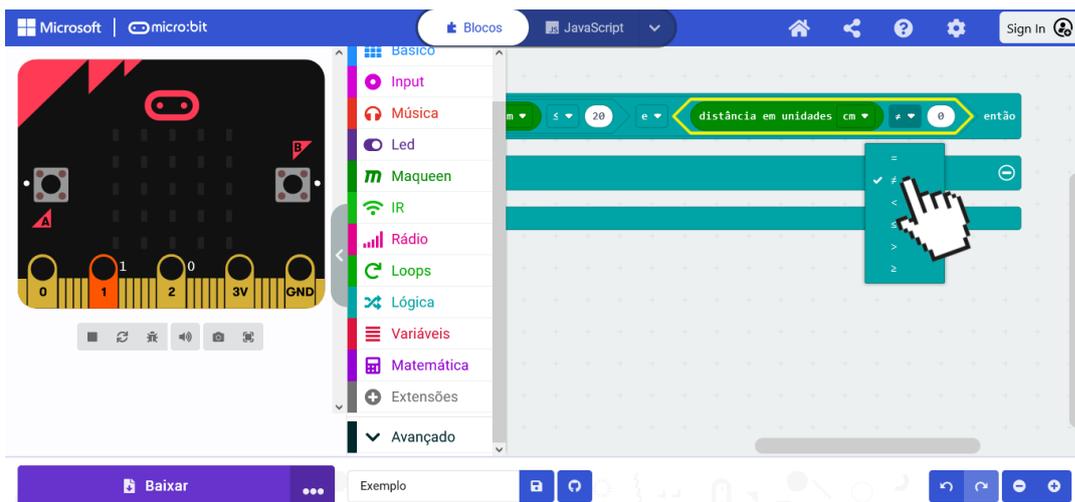


Figura 20.14: Mudando uma comparação.

# Carro que desvia

**12.** Em **Lógica**, selecione outro bloco **se verdadeiro então, senão**, Figura 20.15;

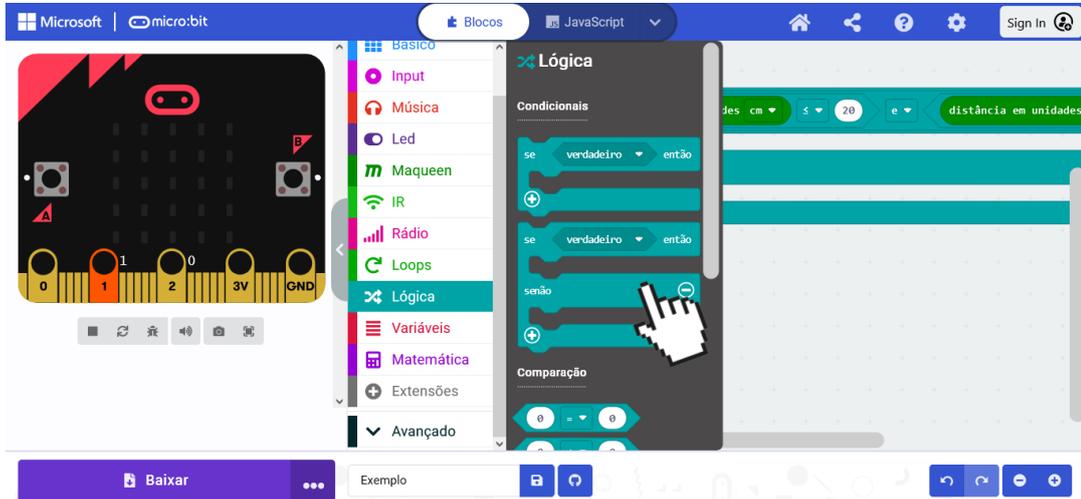


Figura 20.15: Selecionando um bloco condicional.

**13.** Insira o bloco selecionado dentro do bloco **se distância em unidades cm  $\leq$  20**, Figura 20.16;



Figura 20.16: Montando um bloco.



# Carro que desvia

14. Na aba **Matemática**, selecione o comando **escolher de forma aleatória verdadeiro ou falso**, Figura 20.17;

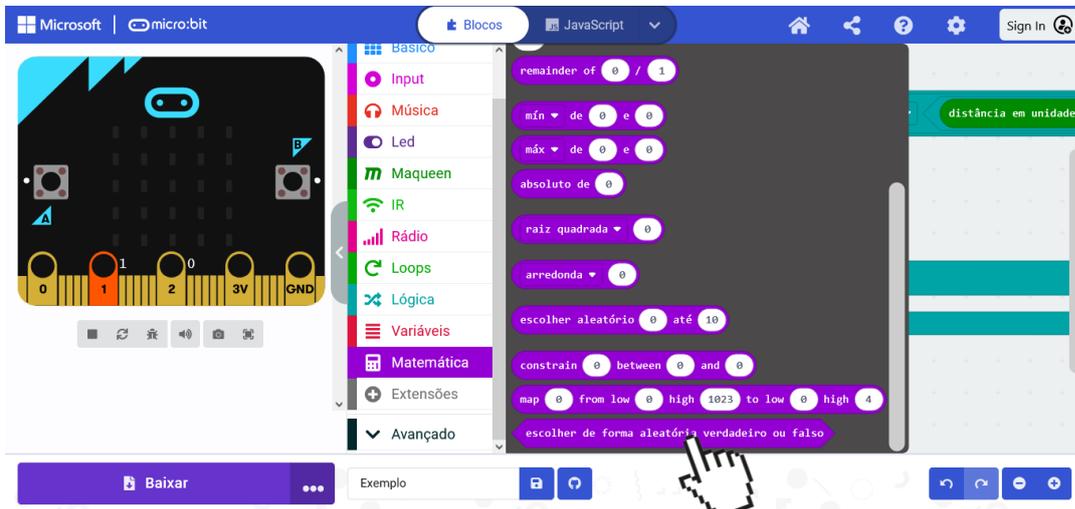


Figura 20.17: Selecionando um comando.

15. Insira o comando selecionado no lugar de **verdadeiro** do bloco **se verdadeiro então**, como na Figura 20.18;

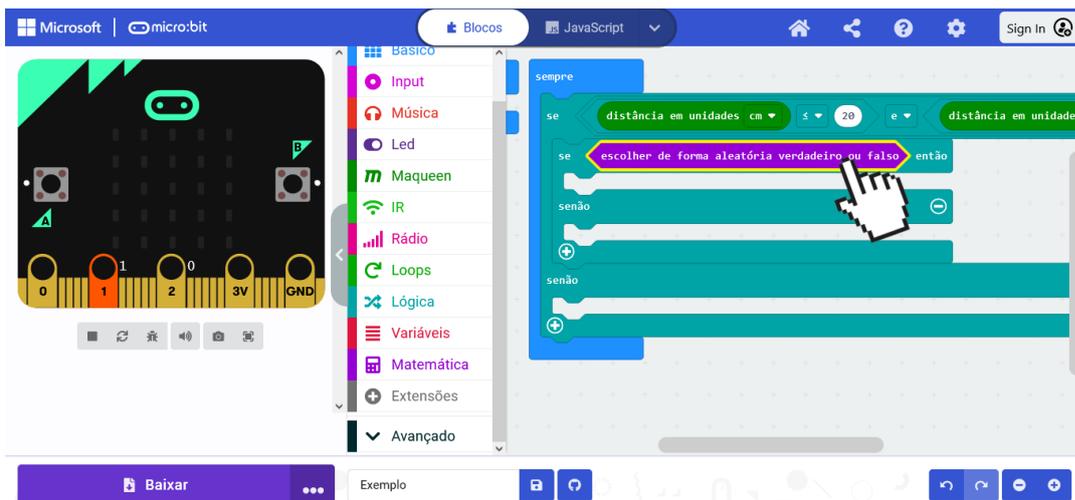


Figura 20.18: Montando uma condição.

# Carro que desvia

- 16.** Em **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0**, Figura 20.19;

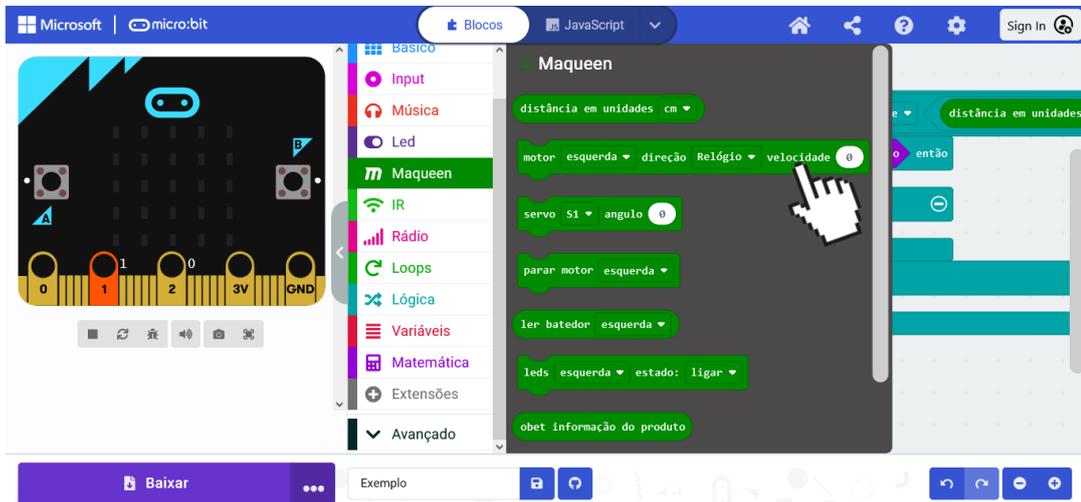


Figura 20.19: Selecionando um comando.

- 17.** Insira o comando selecionado dentro do bloco **se escolher de forma aleatória verdadeiro ou falso então**, Figura 20.20;

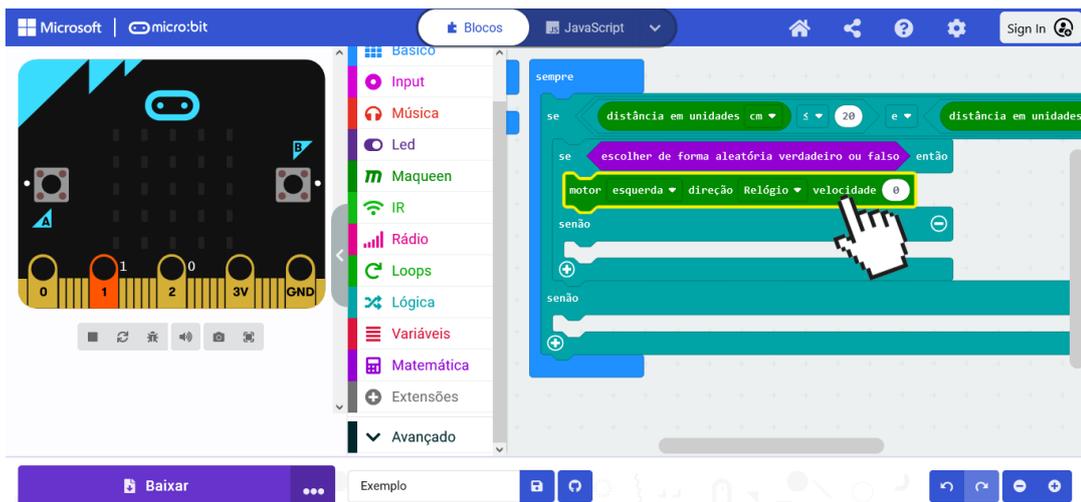


Figura 20.20: Montando um comando.



# Carro que desvia

18. No comando **motor esquerda direção Relógio velocidade 0**, altere o número **0** para **255**, como na Figura 20.21;



Figura 20.21: Alterando uma velocidade.

19. Em **Maqueen**, selecione outro comando **motor esquerda direção Relógio velocidade 0** e insira-o logo abaixo de **motor esquerda direção Relógio velocidade 255**, Figuras 20.22 e 20.23;

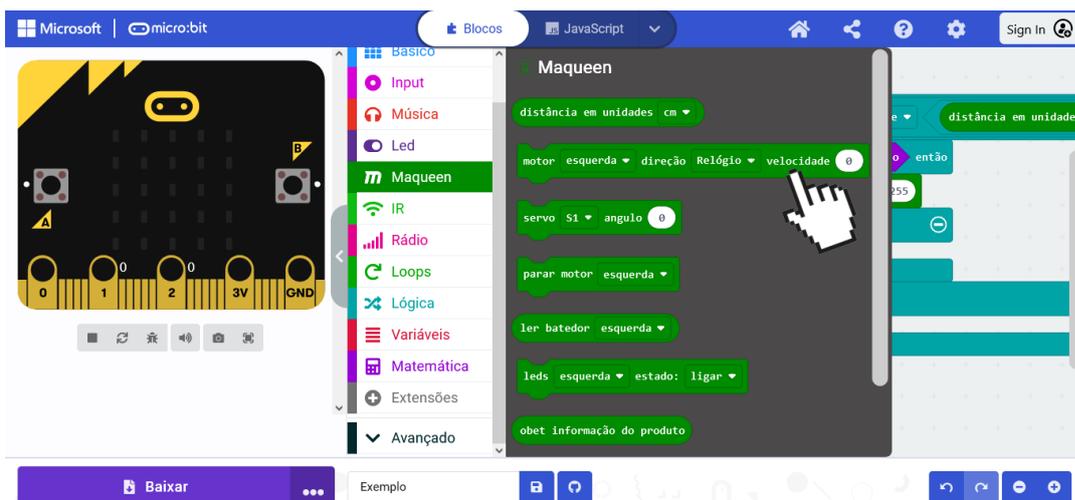


Figura 20.22: Selecionando um comando.

# Carro que desvia

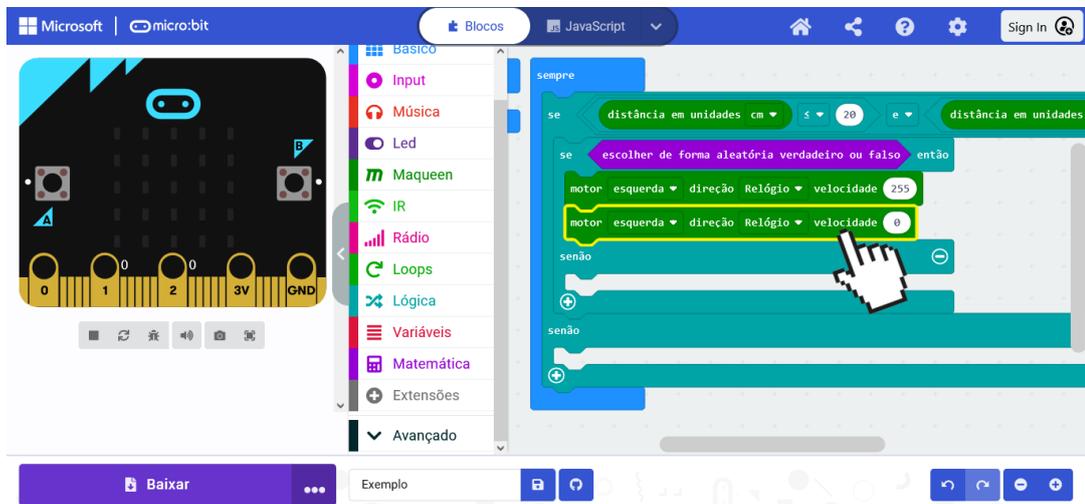


Figura 20.23: Montando um comando.

- 20.** No segundo comando **motor esquerda direção Relógio velocidade 0**, altere motor **esquerda** para **direita**. Depois, altere a direção **Relógio** para **Contra-relógio** e, por fim, mude o número **0** para **255**, como nas Figuras 20.24, 20.25 e 20.26;



Figura 20.24: Alterando um motor.



# Carro que desvia

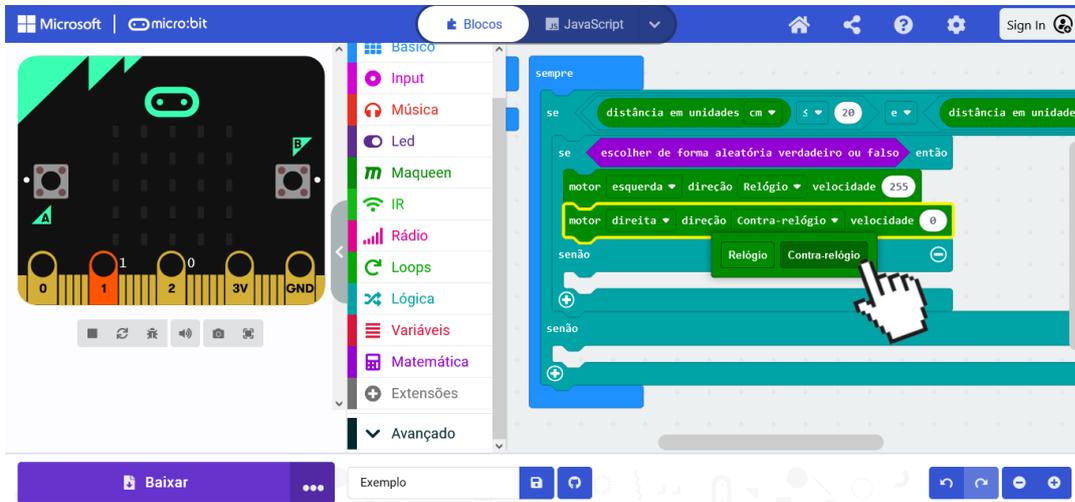


Figura 20.25: Alterando uma direção.

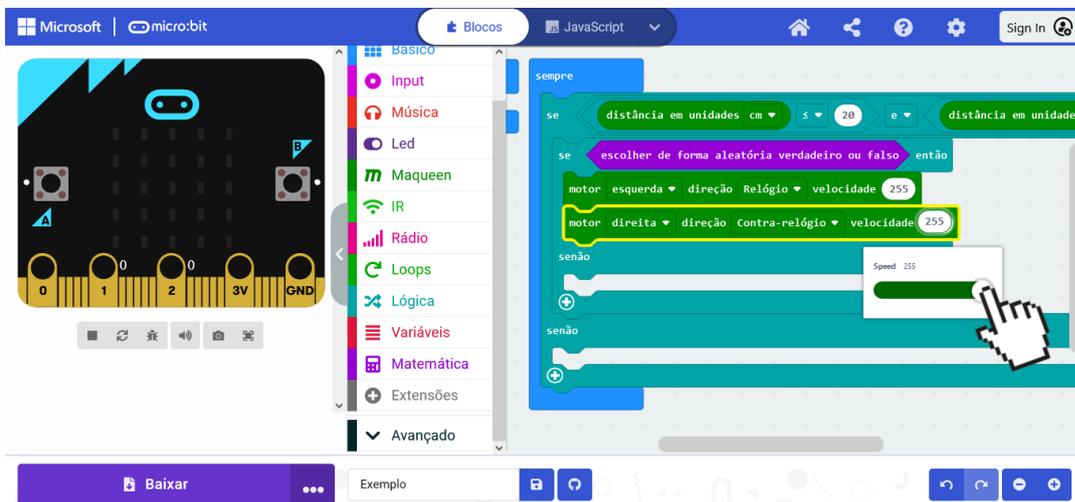


Figura 20.26: Alterando uma velocidade.

# Carro que desvia

- 21.** Em **Maqueen**, selecione o comando **leds esquerda estado: ligar**, Figura 20.27;

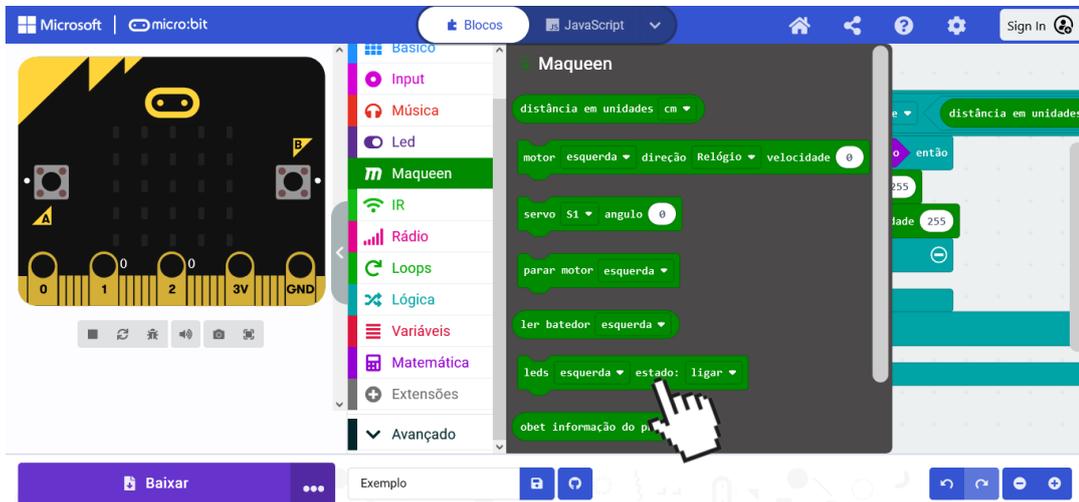


Figura 20.27: Selecionando um comando.

- 22.** Insira o comando selecionado abaixo de **motor direita Contra-relógio velocidade 255**, como na Figura 20.28;

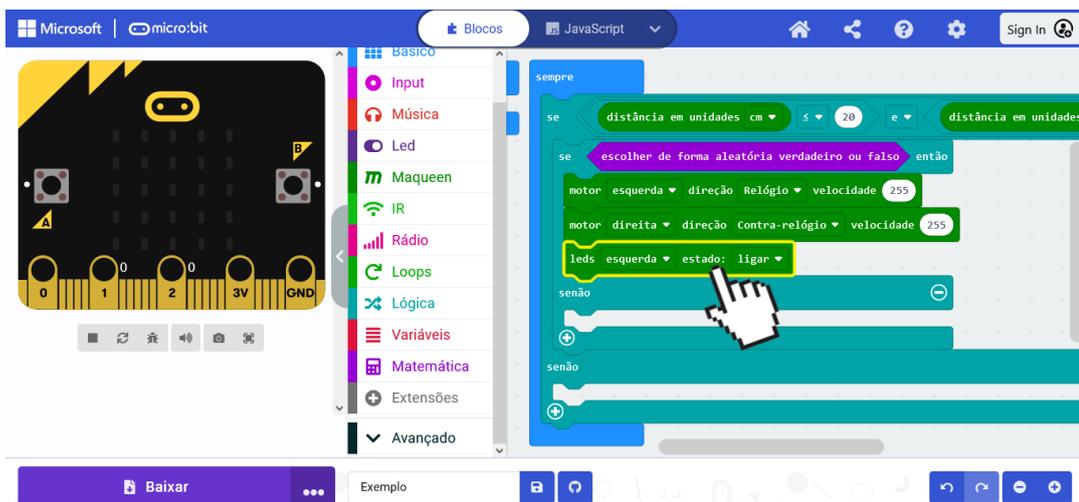


Figura 20.28: Montando um comando.



# Carro que desvia

- 23.** No comando **leds esquerda estado: ligar**, altere **leds esquerda** para **direita**, como na Figura 20.29;

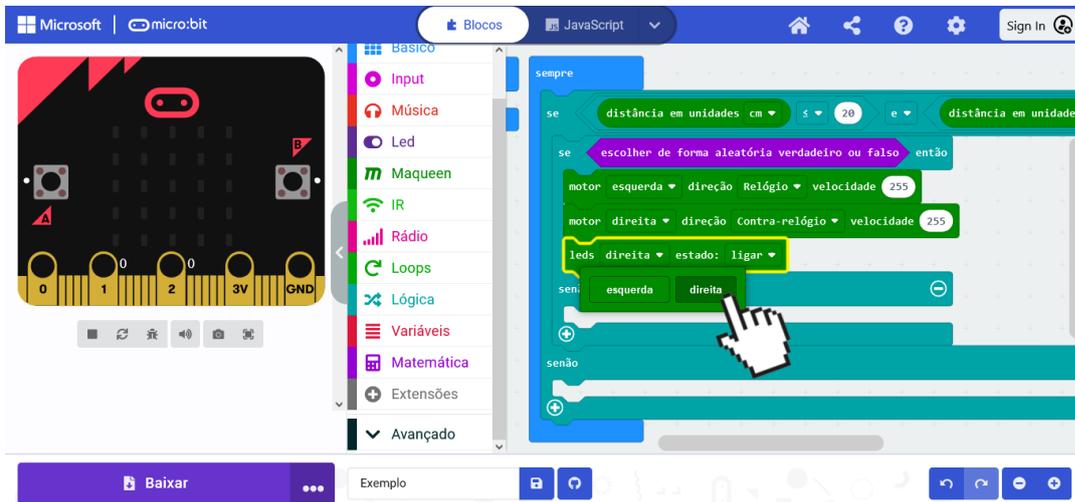


Figura 20.29: Escolhendo um LED.

- 24.** Na aba **Básico**, selecione um comando de **pausa (ms) 100**, Figura 20.30;



Figura 20.30: Selecionando um comando.

# Carro que desvia

**25.** Insira a **pausa** selecionada logo abaixo de **leds direita estado: ligar**, Figura 20.31;

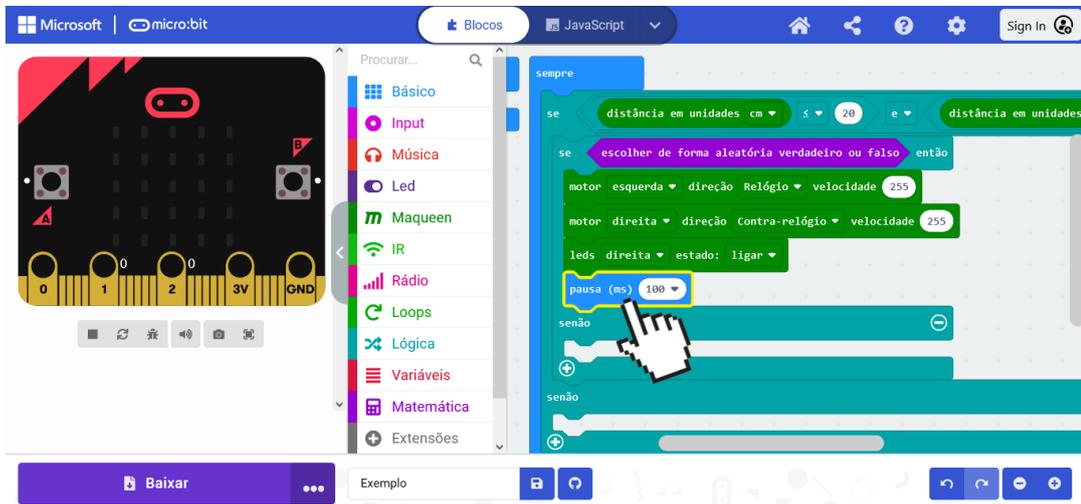


Figura 20.31: Montando uma pausa.

**26.** Altere o tempo da **pausa** de **100 ms** para **1 segundo**, Figura 20.32;

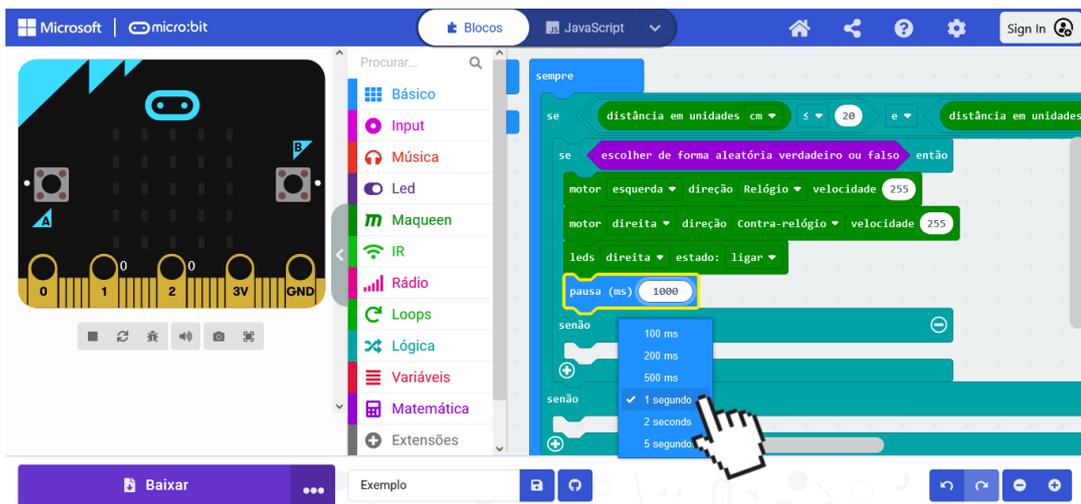


Figura 20.32: Alterando o tempo de uma pausa.



# Carro que desvia

- 27.** Em **Maqueen**, selecione novamente o comando **leds esquerda estado: ligar**, Figura 20.33;

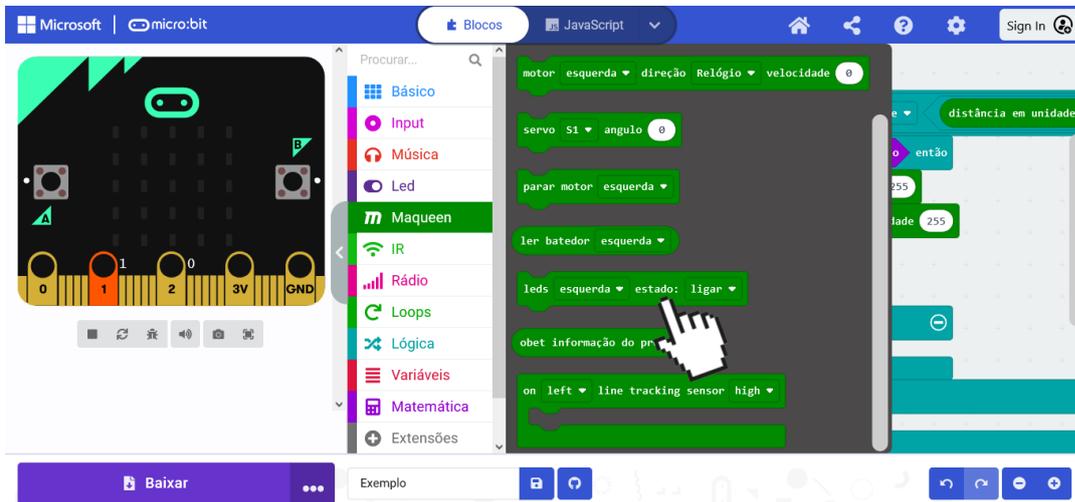


Figura 20.33: Selecionando um comando.

- 28.** Insira o último comando selecionado logo abaixo de **pausa (ms) 1000**, Figura 20.34;

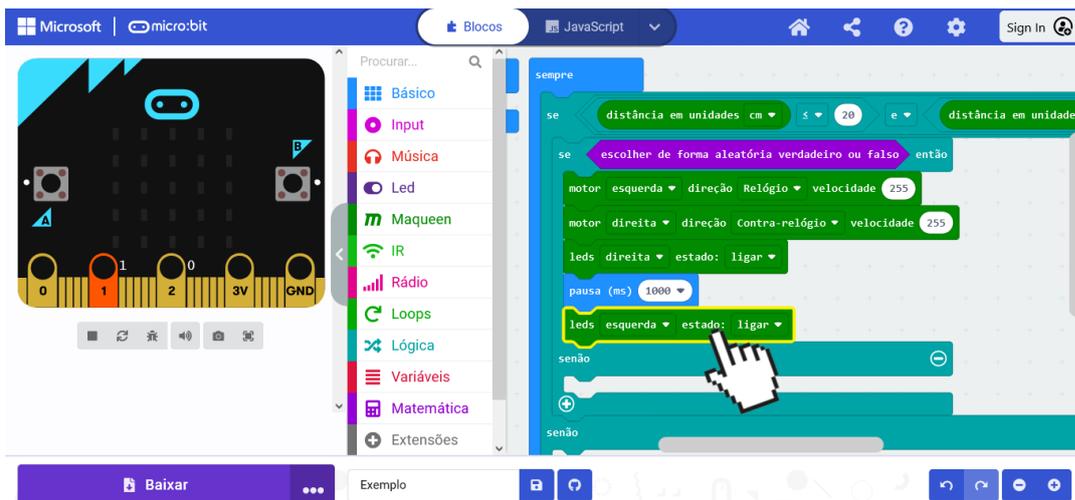


Figura 20.34: Montando um comando.

# Carro que desvia

29. No comando **leds esquerda estado: ligar**, altere o estado **ligar** para **desligar** e **esquerda** para **direita**, Figuras 20.35 e 20.36;

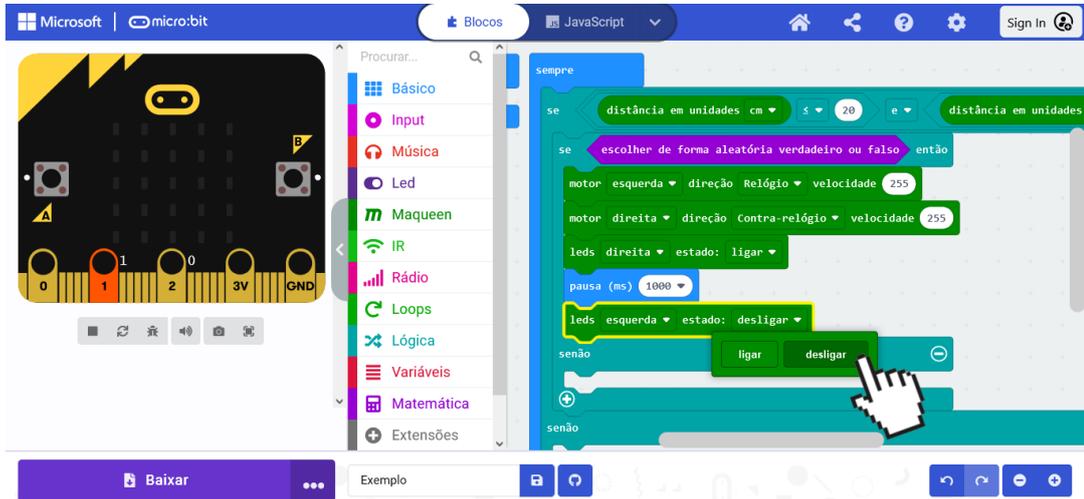


Figura 20.35: Apagando um LED.

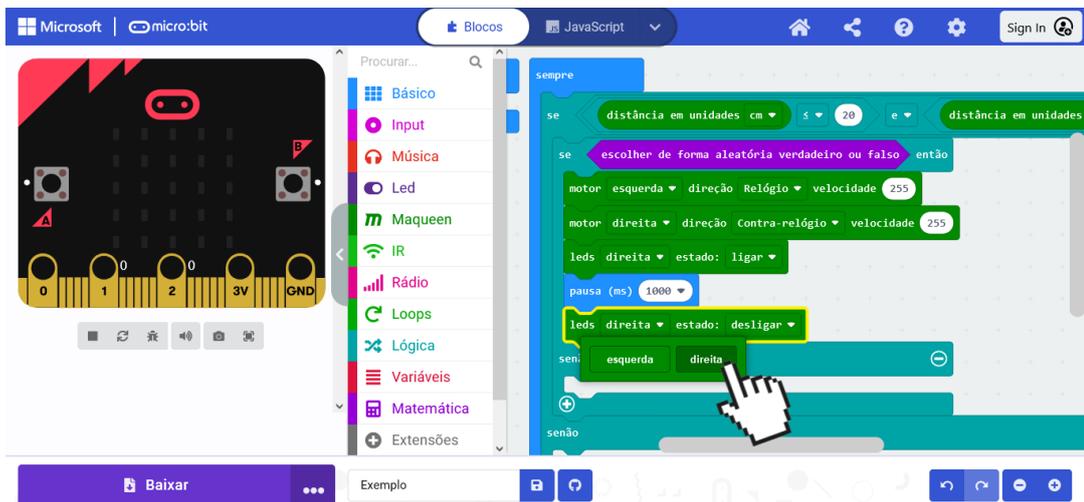


Figura 20.36: Escolhendo um LED.



# Carro que desvia

**30.** Em **Maqueen**, selecione **motor esquerda direção Relógio velocidade 0**, Figura 20.37;

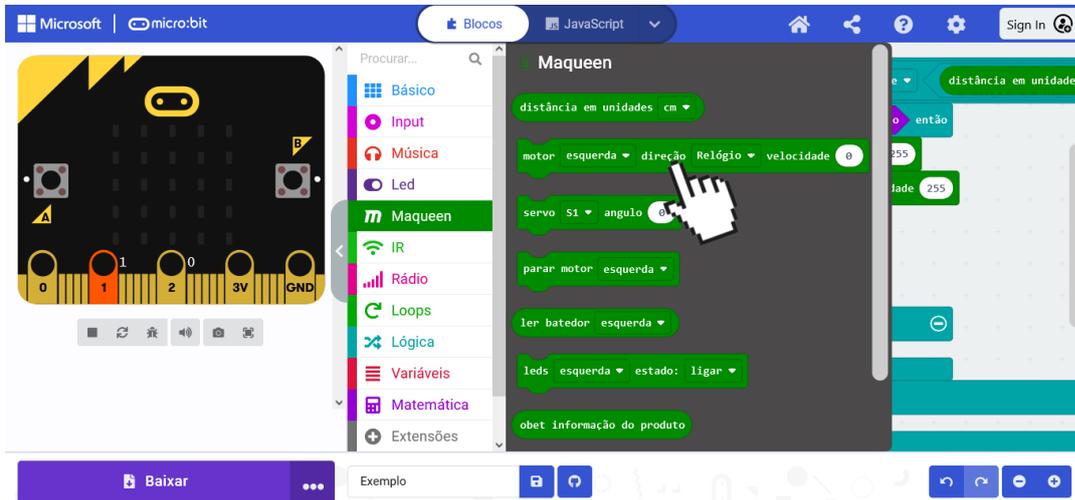


Figura 20.37: Selecionando um comando.

**31.** Insira o comando selecionado dentro do primeiro **senão**, Figura 20.38;



Figura 20.38: Montando um comando.

# Carro que desvia

**32.** Em **motor esquerda direção Relógio velocidade 0**, altere motor **esquerda** para **direita** e a velocidade **0** para **255**, como nas Figuras 20.39 e Figura 20.40;

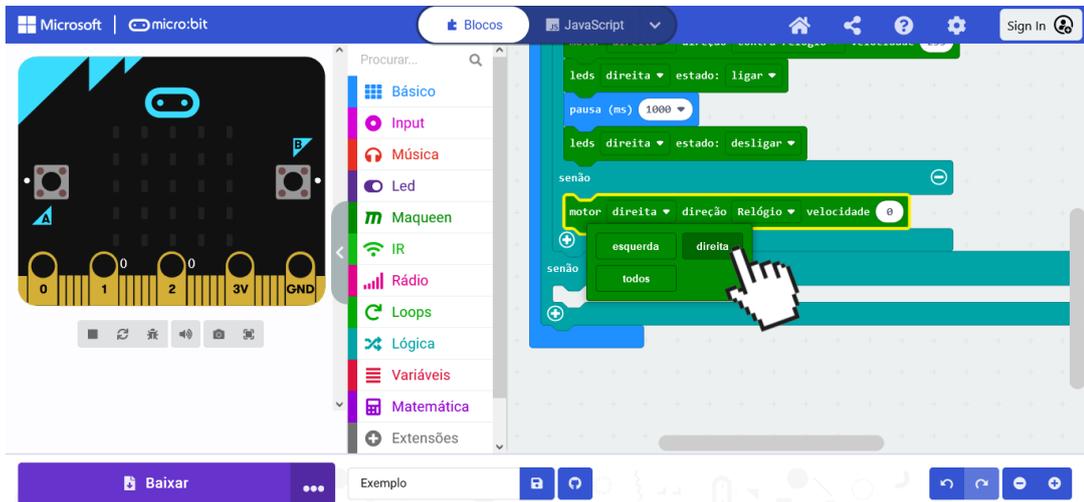


Figura 20.39: Alterando um motor.

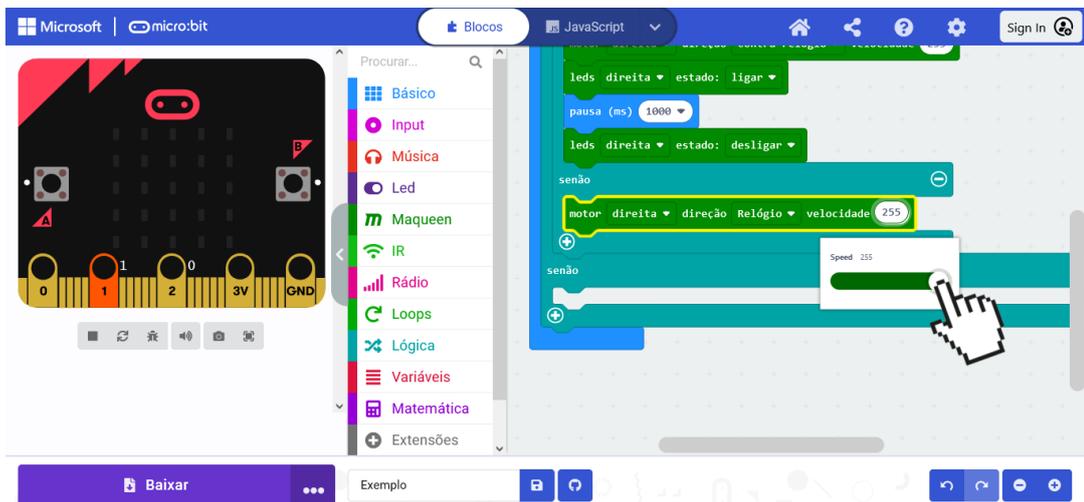


Figura 20.40: Alterando uma velocidade.



# Carro que desvia

- 33.** Novamente em **Maqueen**, selecione **motor esquerda direção Relógio velocidade 0**, Figura 20.41;

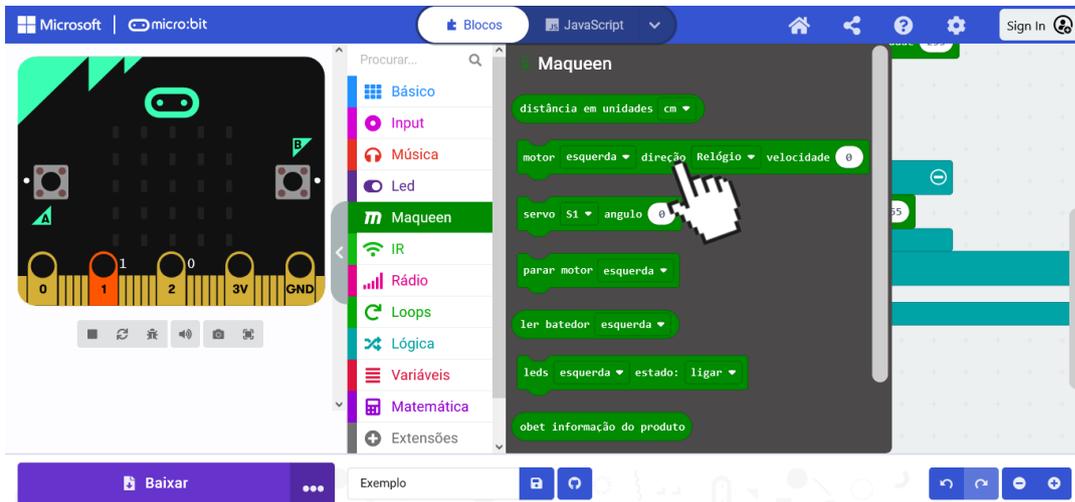


Figura 20.41: Selecionando um comando.

- 34.** Insira o comando selecionado logo abaixo de **motor direita direção Relógio velocidade 255**, Figura 20.42;

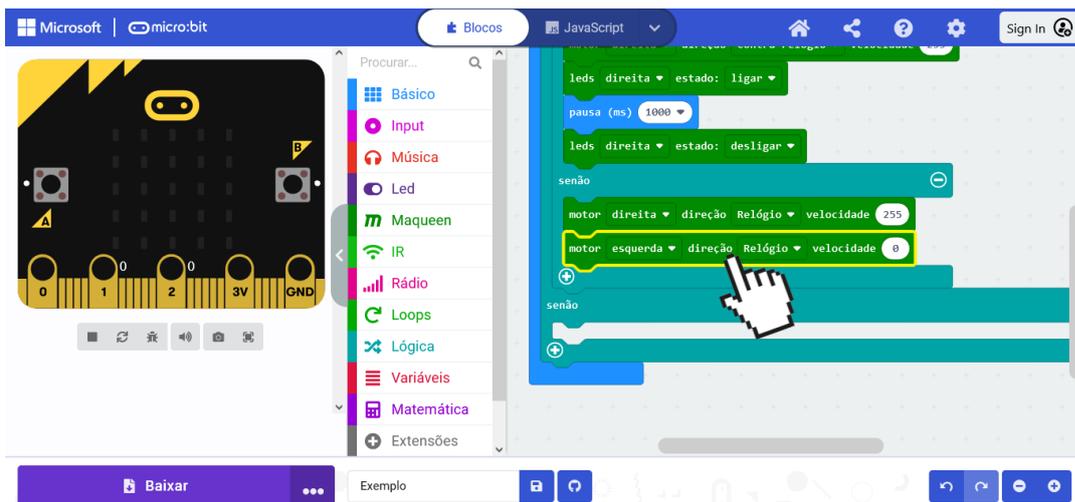


Figura 20.42: Montando um comando.

# Carro que desvia

35. Em **motor esquerda direção Relógio velocidade 0**, altere a direção **Relógio** para **Contra-relógio** e a velocidade **0** para **255**, como nas Figuras 20.43 e 20.44;

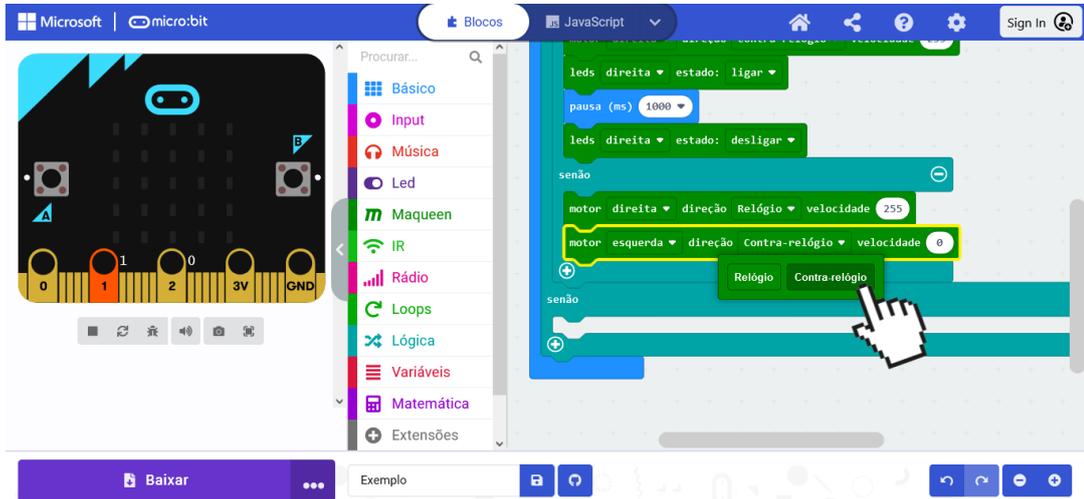


Figura 20.43: Alterando a direção de um motor.

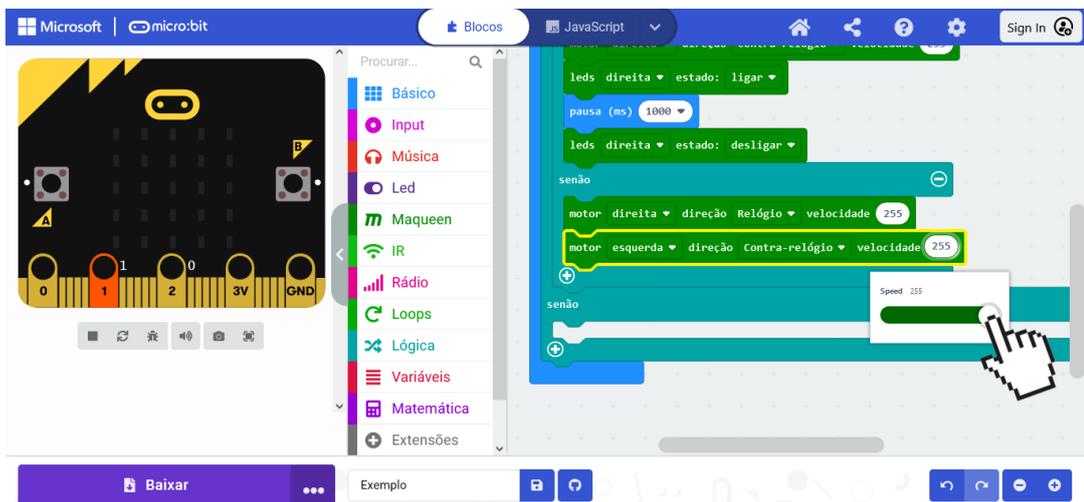
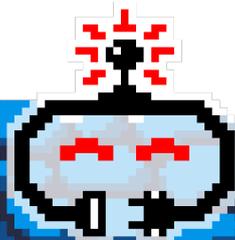


Figura 20.44: Alterando uma velocidade.



# Carro que desvia

**36.** Em **Maqueen**, selecione **leds esquerda estado: ligar**, Figura 20.45;

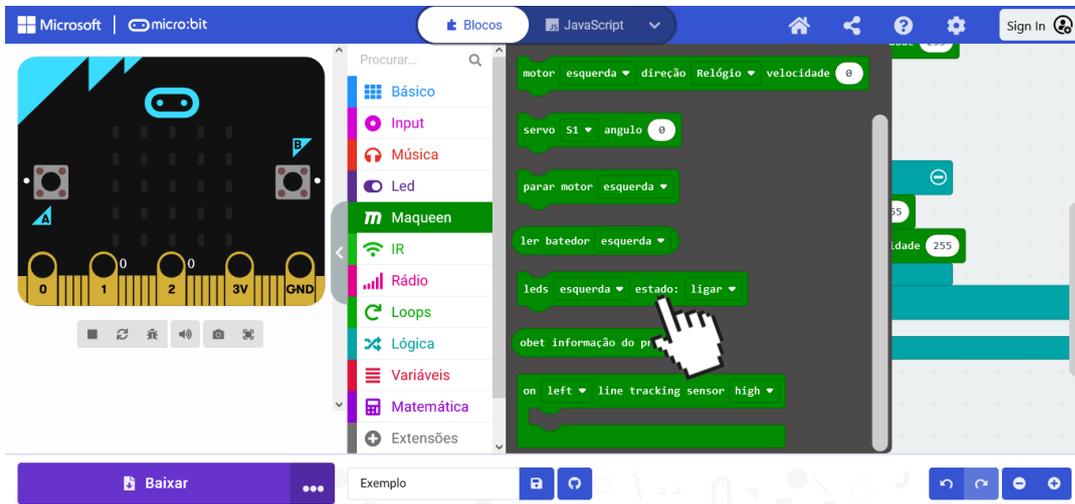


Figura 20.45: Selecionando um comando.

**37.** Insira o comando selecionado logo abaixo de **motor esquerda direção Contra-relógio velocidade 255**, Figura 20.46;

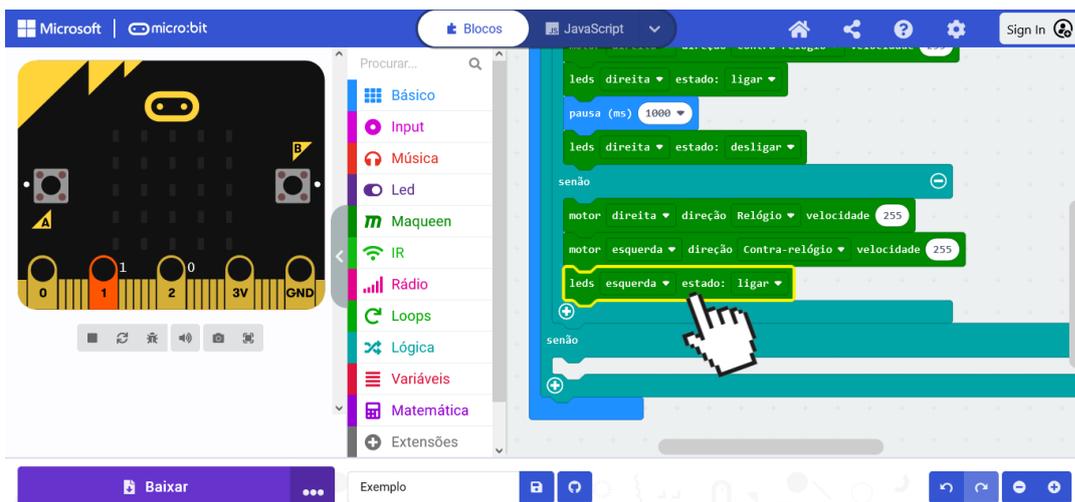


Figura 20.46: Montando um comando.

# Carro que desvia

**38.** Em **Básico**, selecione **pausa (ms) 100**, Figura 20.47;

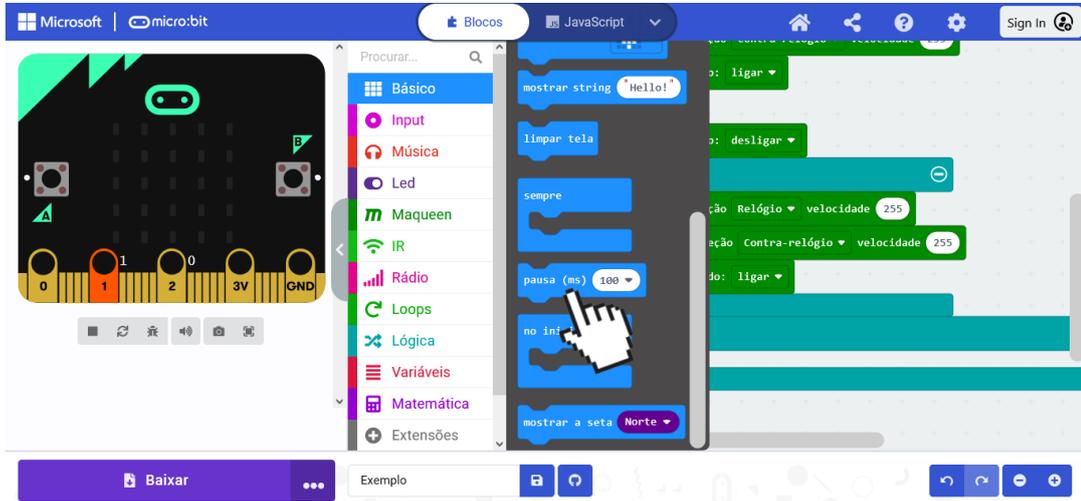


Figura 20.47: Selecionando um comando.

**39.** Adicione a **pausa** logo abaixo de **leds esquerda estado: ligar**, Figura 20.48;

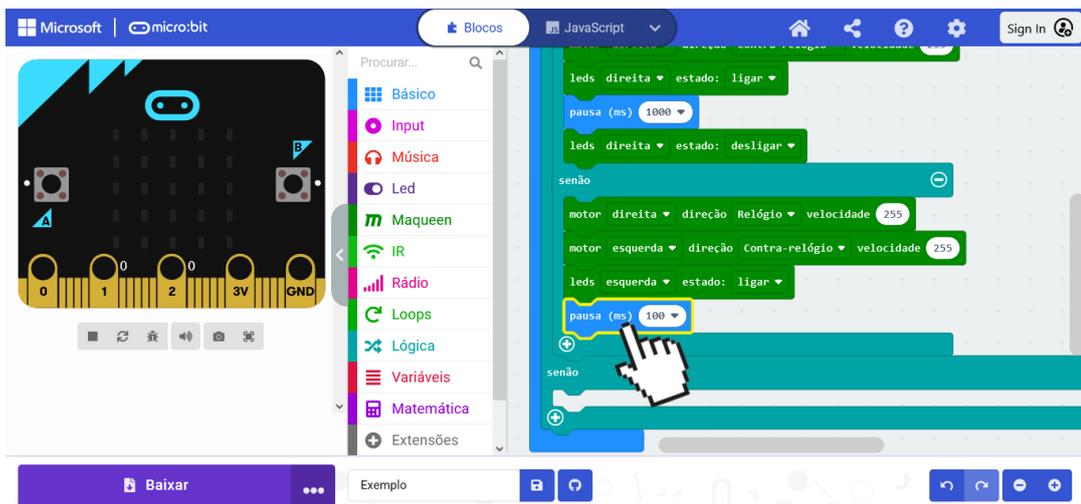


Figura 20.48: Montando um comando.



# Carro que desvia

- 40.** Em **pausa (ms) 100**, altere o tempo de **100 ms** para **1 segundo**, como na Figura 20.49;

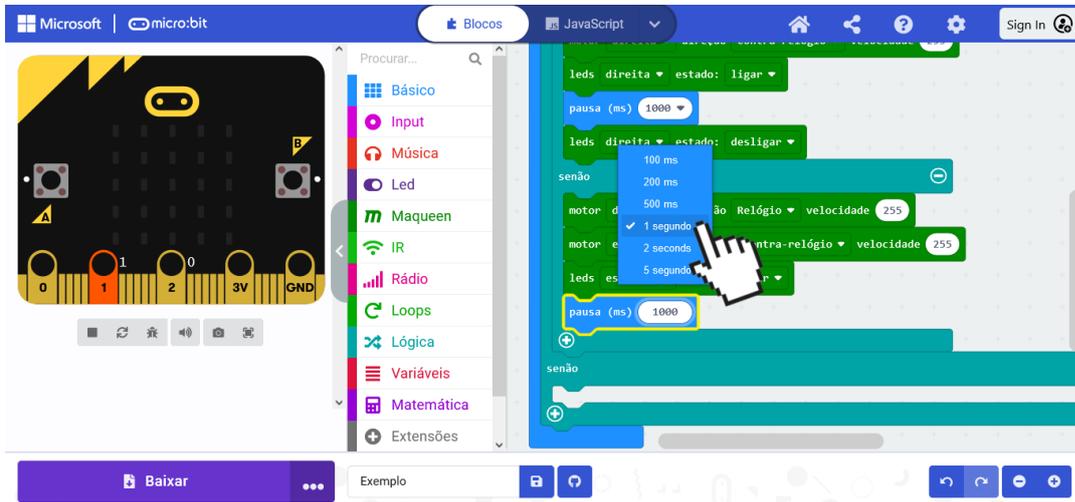


Figura 20.49: Alterando o tempo de uma pausa.

- 41.** Na aba **Maqueen**, selecione o comando **leds esquerda estado: ligar**, Figura 20.50;



Figura 20.50: Selecionando um comando.

# Carro que desvia

- 42.** Adicione o último comando selecionado logo abaixo de **pausa (ms) 1000**, Figura 20.51;

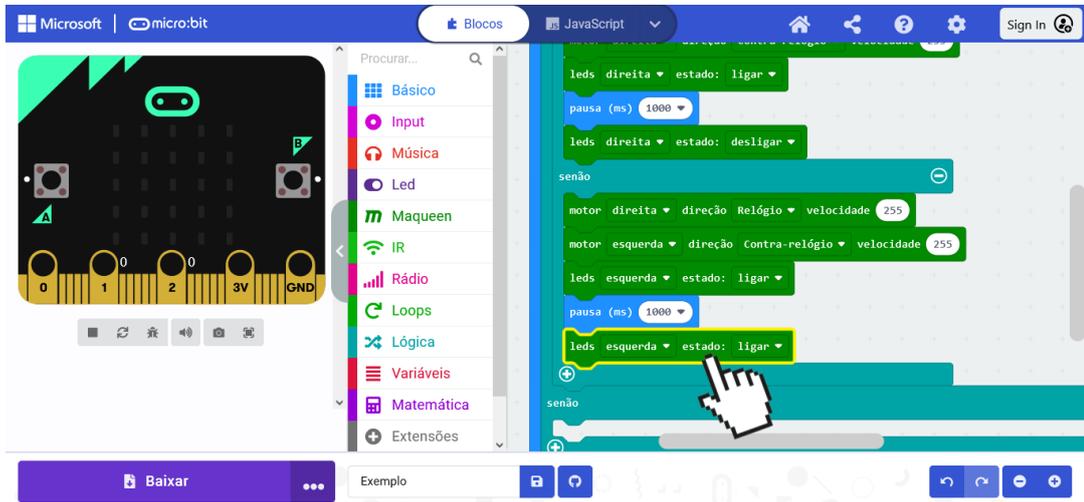


Figura 20.51: Montando um comando.

- 43.** Altere o estado **ligar** do bloco  **leds esquerda estado: ligar**  para **desligar**, como na Figura 20.52;

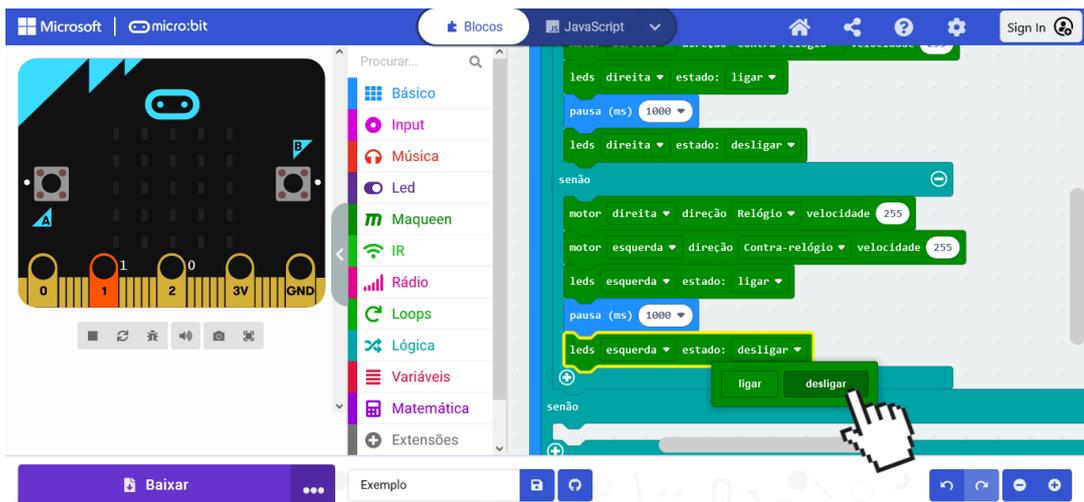


Figura 20.52: Desligando um LED.



# Carro que desvia

44. Em **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0**, Figura 20.53;

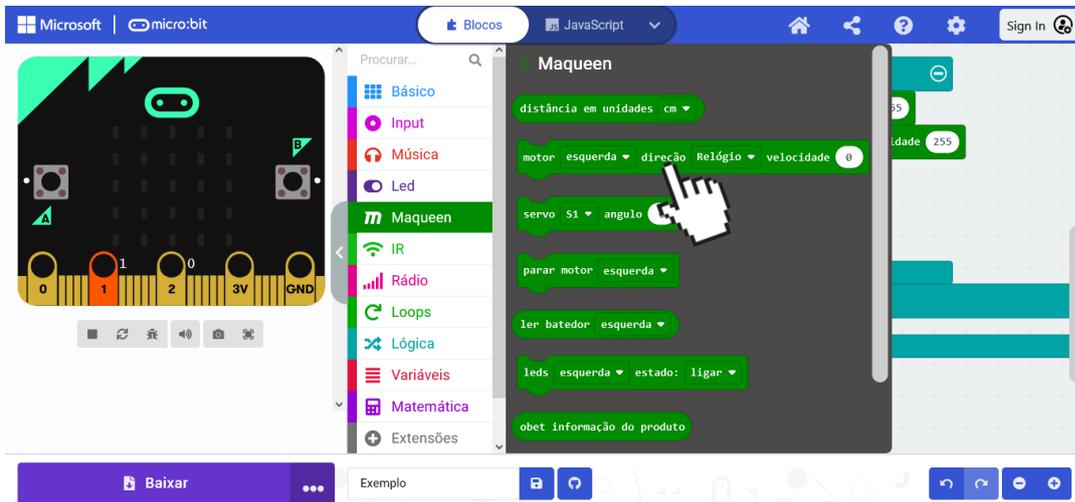


Figura 20.53: Selecionando um comando.

45. Insira o comando selecionado dentro do último **senão** e altere a velocidade **0** de **motor esquerda direção Relógio velocidade 0**, para **255**, como nas Figuras 20.54 e Figura 20.55;



Figura 20.54: Montando um comando.

# Carro que desvia

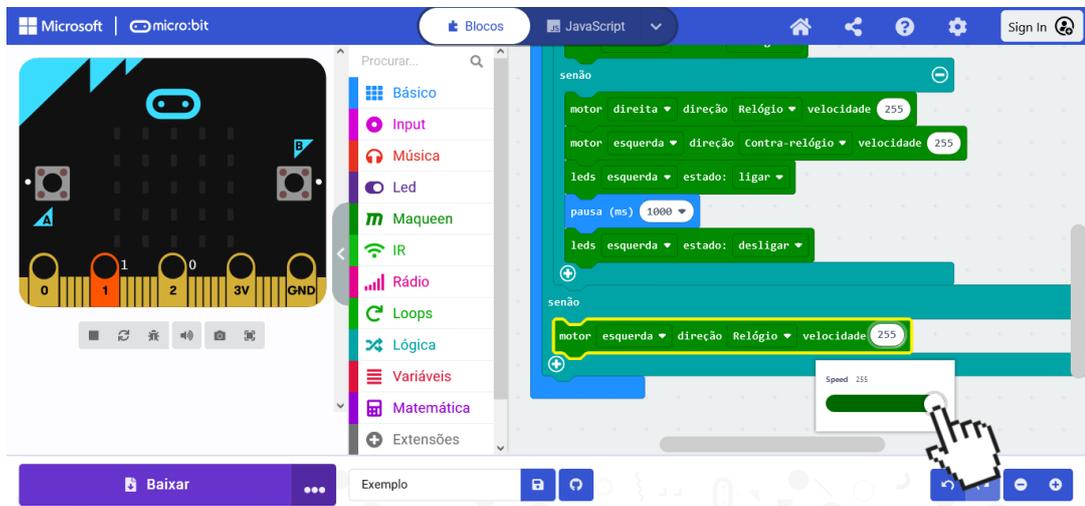


Figura 20.55: Alterando uma velocidade.

- 46.** Novamente na aba **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0** e insira-o logo abaixo de **motor esquerda direção Relógio velocidade 255**, Figuras 20.56 e 20.57;

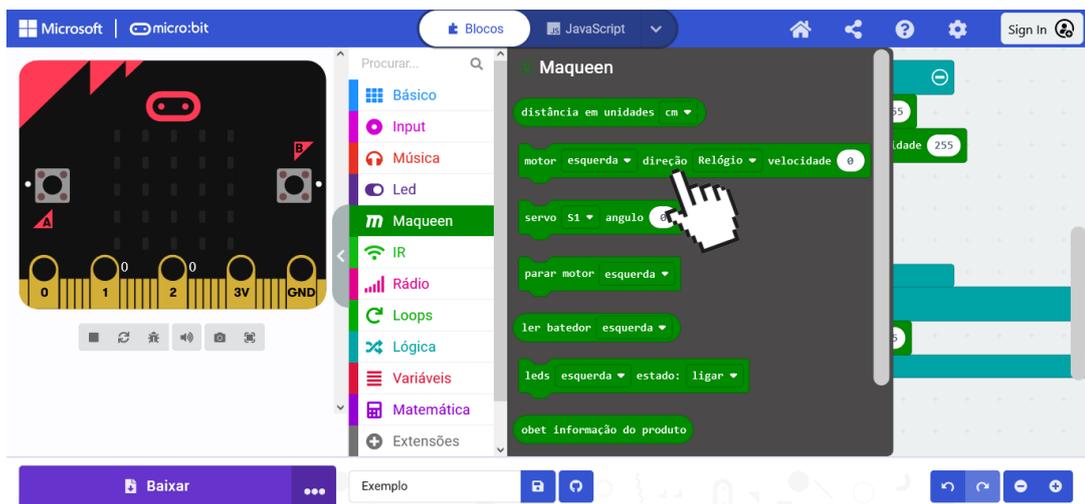


Figura 20.56: Selecionando um comando.



# Carro que desvia

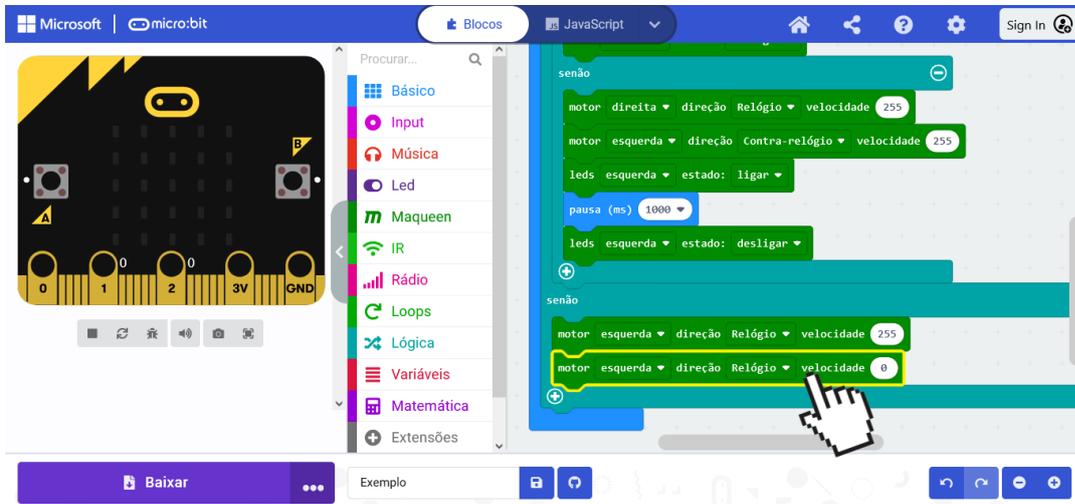


Figura 20.57: Montando um comando.

- 47.** Em **motor esquerda direção Relógio velocidade 0**, altere motor **esquerda** para **direita** e a velocidade **0** para **255**, como nas Figuras 20.58 e 20.59;

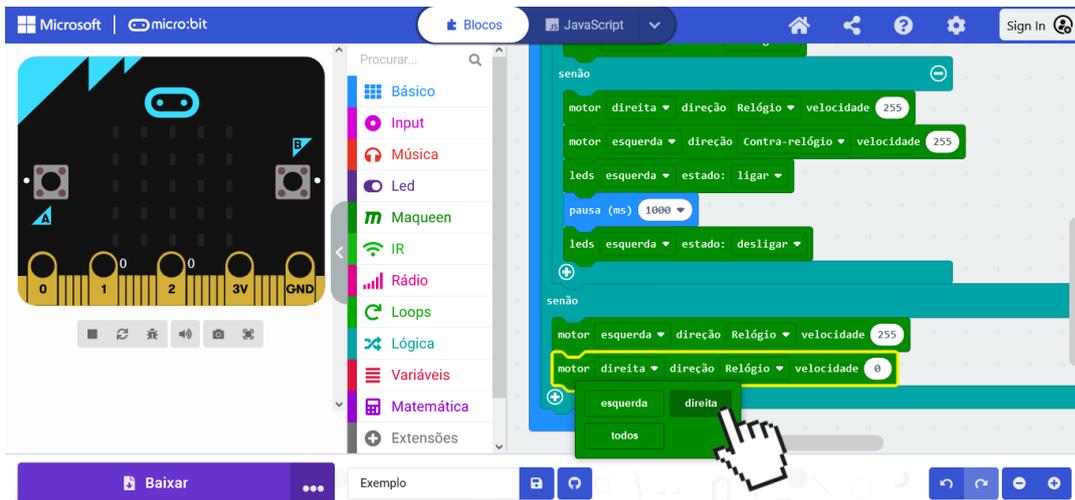


Figura 20.58: Alterando um motor.

# Carro que desvia

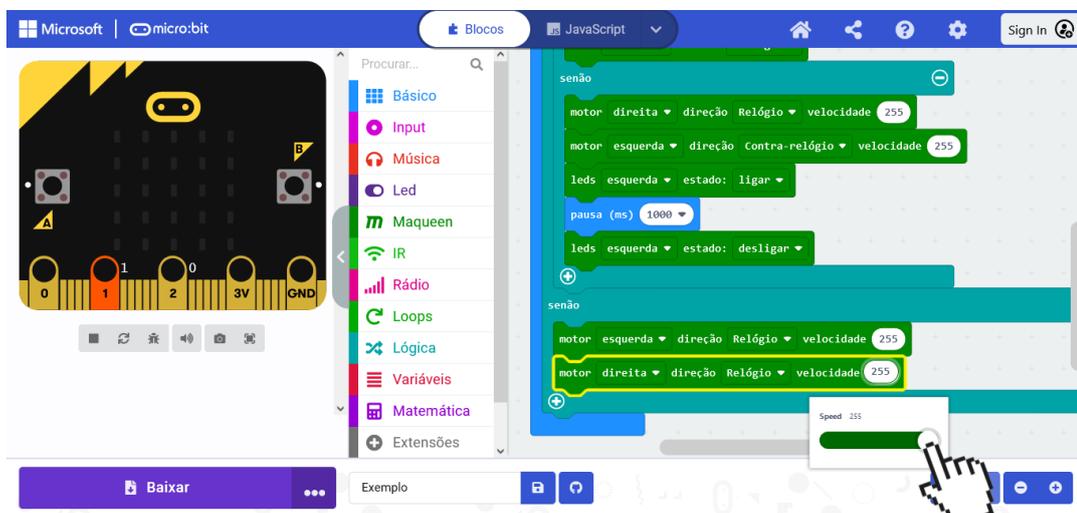


Figura 20.59: Alterando uma velocidade.

**Baixe o programa para o micro:bit e encaixe a placa no Maqueen para vê-lo andar sozinho sem bater em nenhum obstáculo.**

[Clique aqui para acessar o código.](#)





Carro que desvia



# Capítulo 21: Carro segue faixa



Para que serve um robô seguidor de faixa? Nas indústrias, por exemplo, esses robôs manejam as prateleiras de produtos dentro do estoque por meio de orientação de faixas no chão. Seguindo essas faixas, eles transportam os materiais de um lado para outro dentro do armazém.

Nesta atividade, o carrinho segue exatamente uma linha de coloração preta, sem qualquer controle remoto, somente executando a programação. Então, aprenda essa programação agora.



# Carro segue faixa

## Passo a passo:

1. Clique na aba **Extensões**, e busque por **Maqueen** na barra de pesquisa, como nas Figuras 21.1 e 21.2;

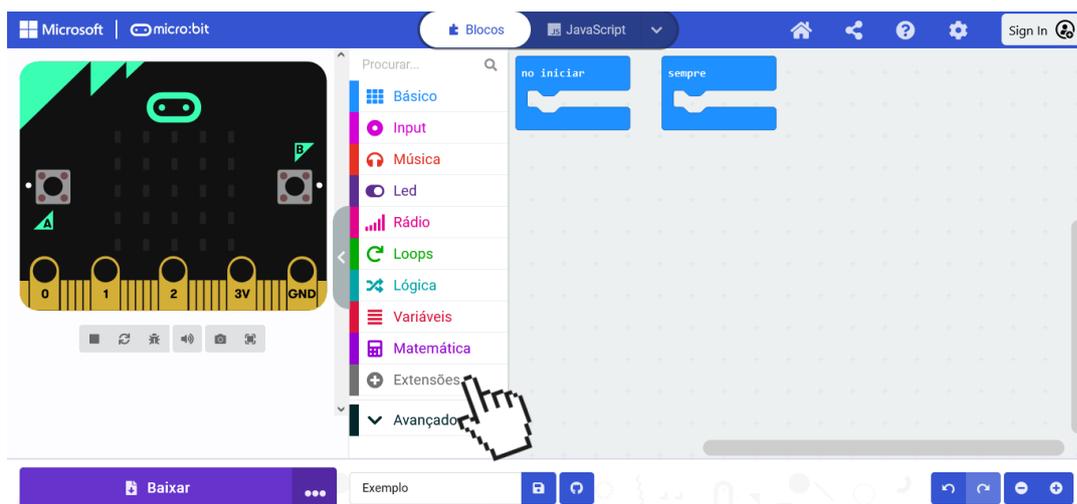


Figura 21.1: Selecionando uma extensão.

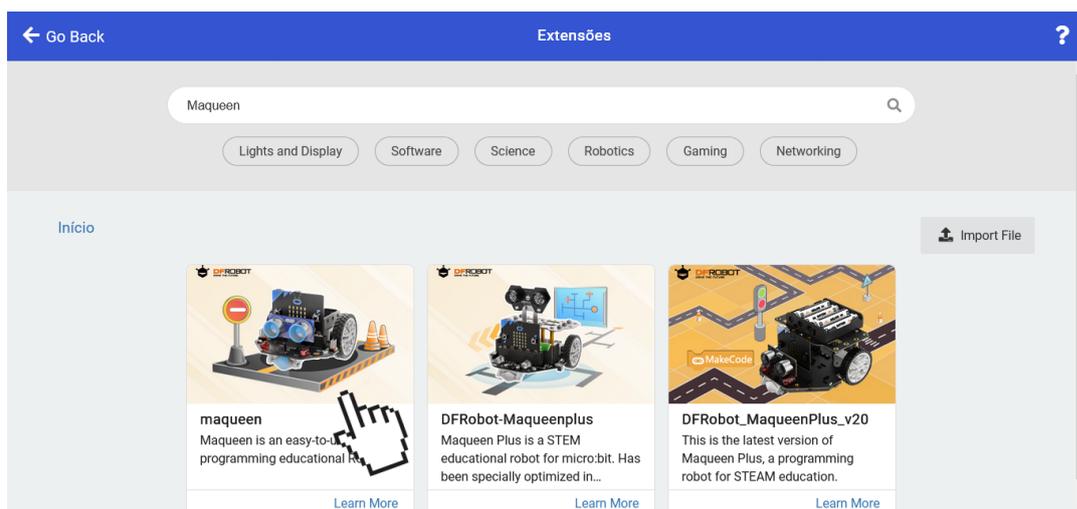


Figura 21.2: Adicionando uma extensão.

# Carro segue faixa

2. Em **Lógica**, selecione o bloco **se verdadeiro então, senão**, Figura 21.3;

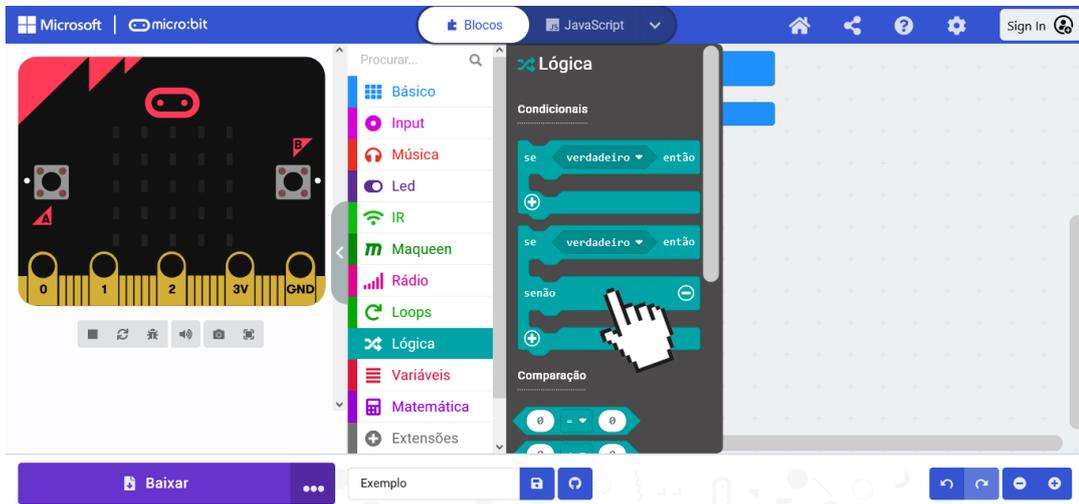


Figura 21.3: Selecionado um bloco.

3. Insira o bloco selecionado dentro do bloco de acionamento **sempre**, Figura 21.4;

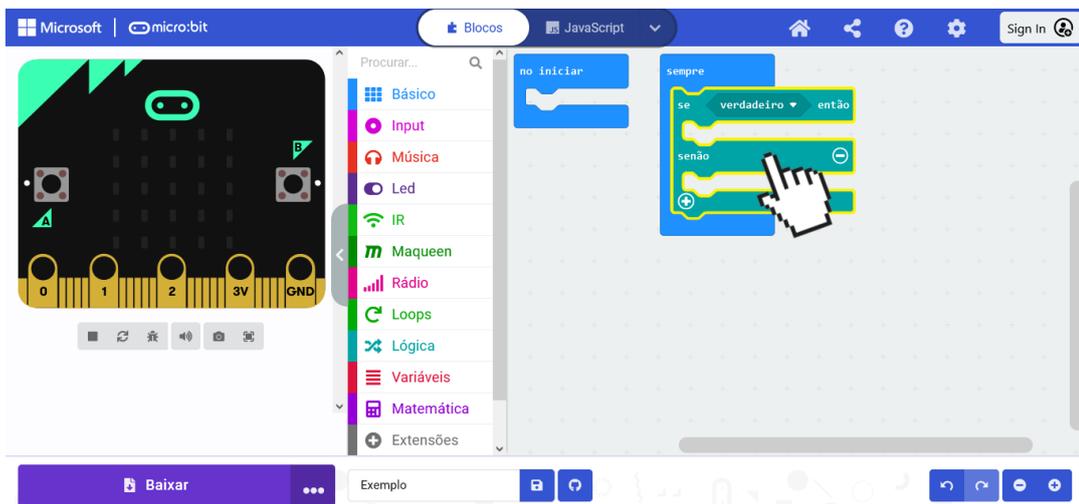


Figura 21.4: Montando um comando.



# Carro segue faixa

4. Em **Lógica**, selecione um condicional Booleano **<> e <>**, Figura 21.5;



Figura 21.5: Selecionando um comando.

5. Insira o comando selecionado no lugar de **verdadeiro** em **se verdadeiro então**, como na Figura 21.6;



Figura 21.6: Montando um comando.

# Carro segue faixa

6. Em **Lógica**, selecione duas vezes a comparação **0 = 0**, Figura 21.7;

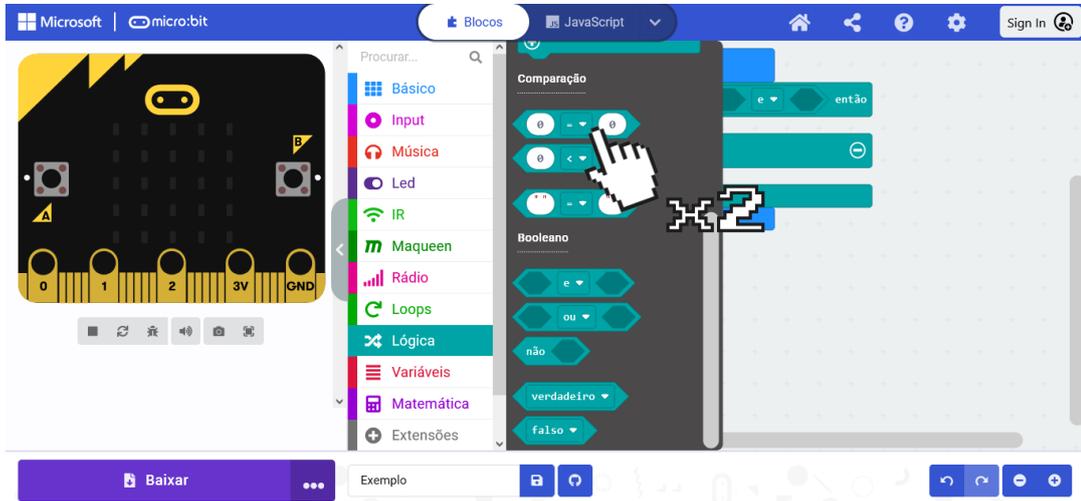


Figura 21.7: Selecionando comandos.

7. Insira uma das comparações no primeiro espaço ao lado de **e** e a outra no segundo espaço do bloco **se <> e <> então**, como na Figura 21.8;



Figura 21.8: Montando uma comparação.



# Carro segue faixa

8. Em **Maqueen**, selecione o comando **ler batedor esquerda**, Figura 21.9;

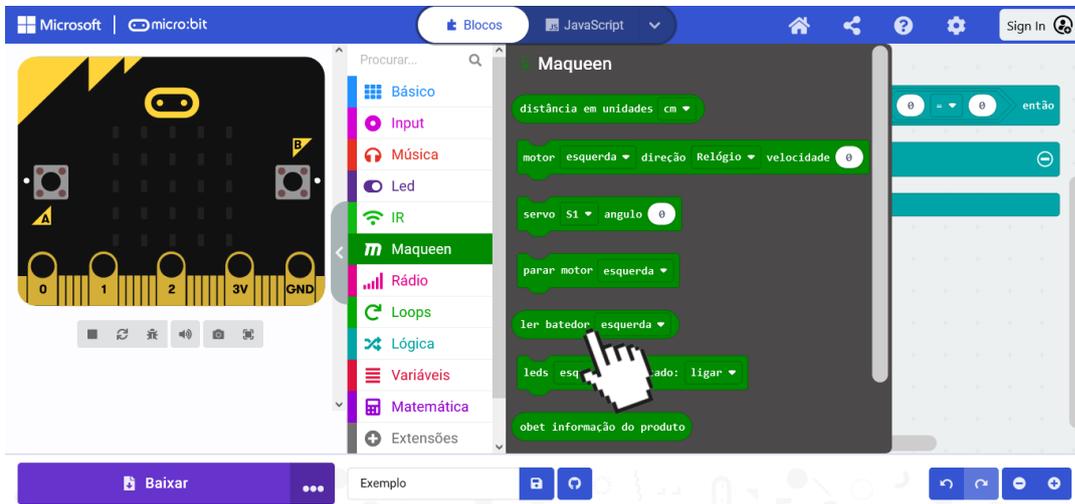


Figura 21.9: Selecionando um comando.

9. Insira o comando selecionado no lugar do primeiro número **0**, da comparação **se 0 = 0 e 0 = 0 então**, como na Figura 21.10;



Figura 21.10: Montando um comando.

# Carro segue faixa

10. Novamente em **Maqueen**, selecione outra vez o comando **ler batedor esquerda** e coloque-o no lugar do primeiro número **0** da segunda comparação, logo após a letra **e**, Figuras 21.11 e 21.12;

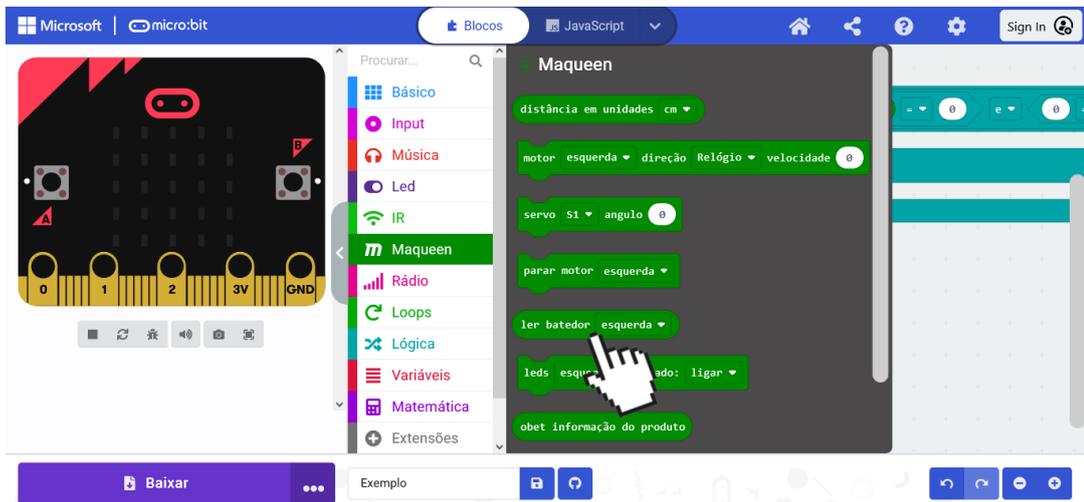


Figura 21.11: Selecionando um comando.

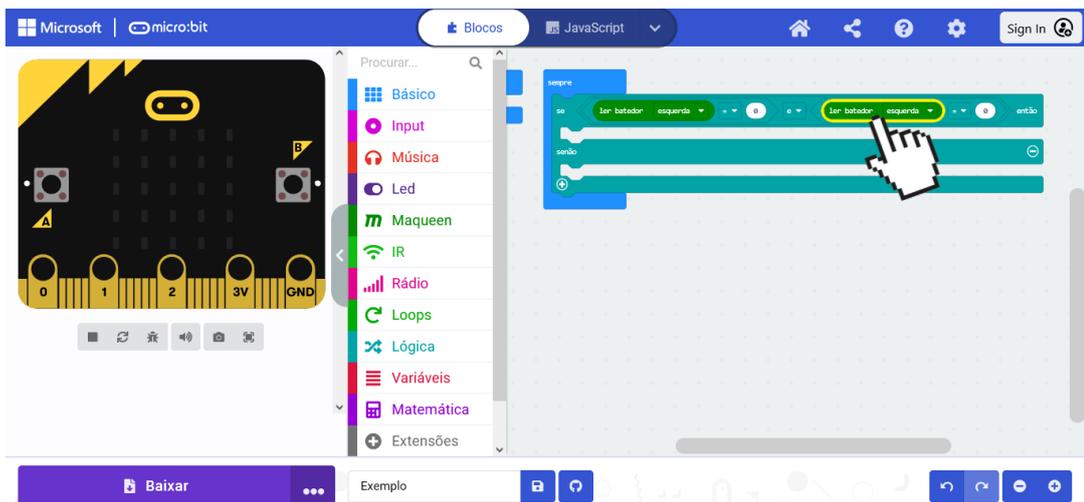


Figura 21.12: Montando um comando.



# Carro segue faixa

11. Altere **esquerda** por **direita** no último comando montado, como na Figura 21.13;

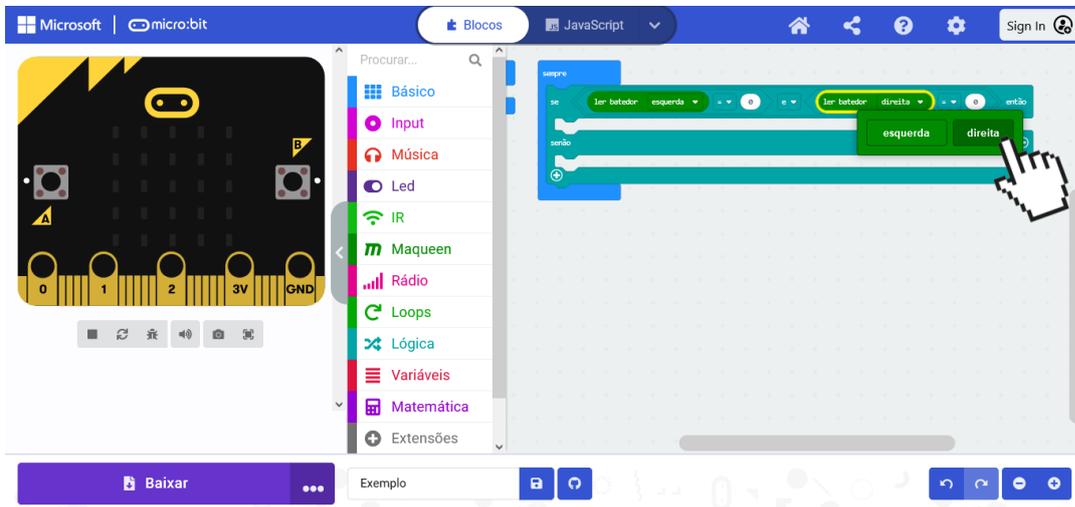


Figura 21.13: Alterando um comando.

12. Em **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0** e insira-o abaixo de **se ler batedor esquerda = 0** Figuras 21.14 e 21.15;

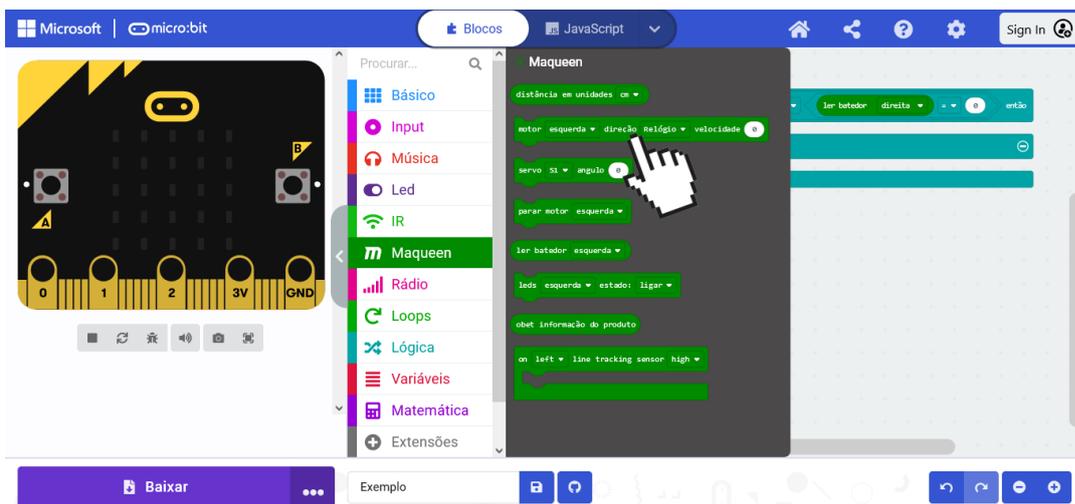


Figura 21.14: Selecionando um comando.

# Carro segue faixa

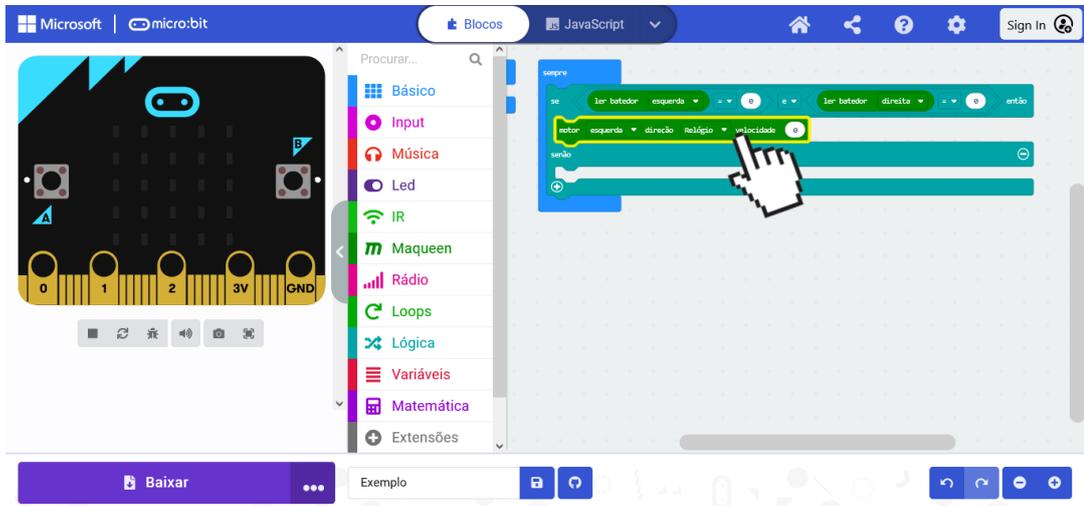


Figura 21.15: Montando um comando.

13. Em **motor esquerda direção Relógio velocidade 0**, altere a velocidade **0** para **150**, como na Figura 21.16;

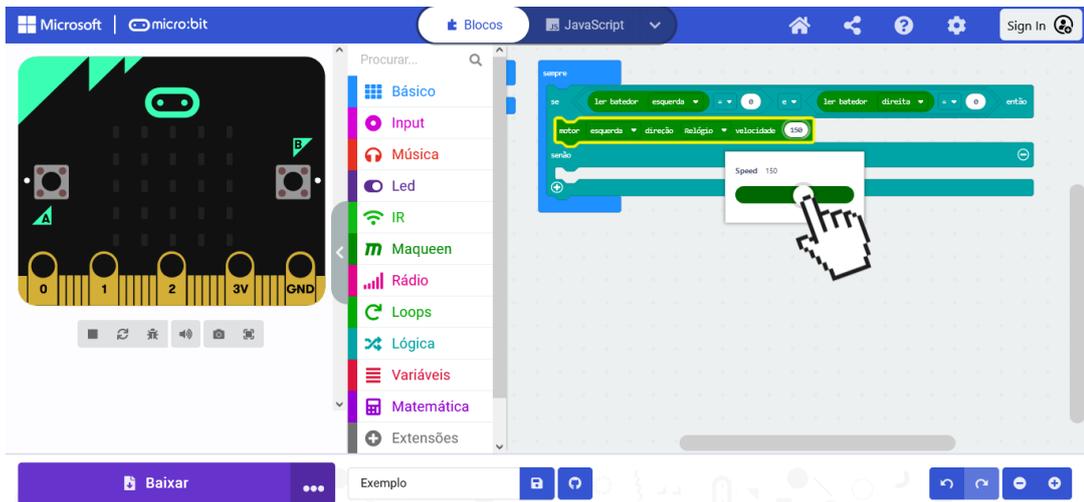


Figura 21.16: Alterando uma velocidade.



# Carro segue faixa

14. Na aba **Maqueen**, selecione novamente o comando **motor esquerda direção Relógio velocidade 0**, Figura 21.17;

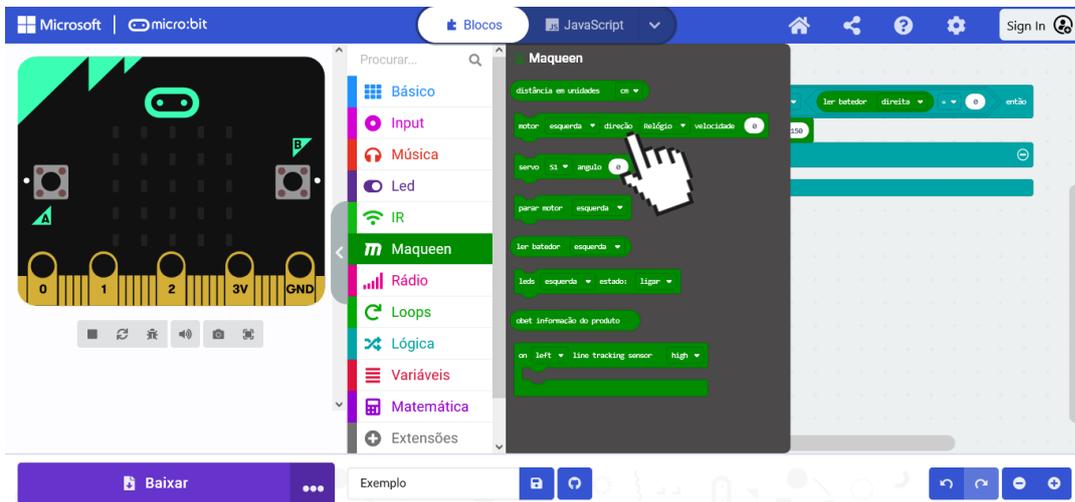


Figura 21.17: Selecionando um comando.

15. Insira o comando selecionado logo abaixo de **motor esquerda direção Relógio velocidade 150**, Figura 21.18;

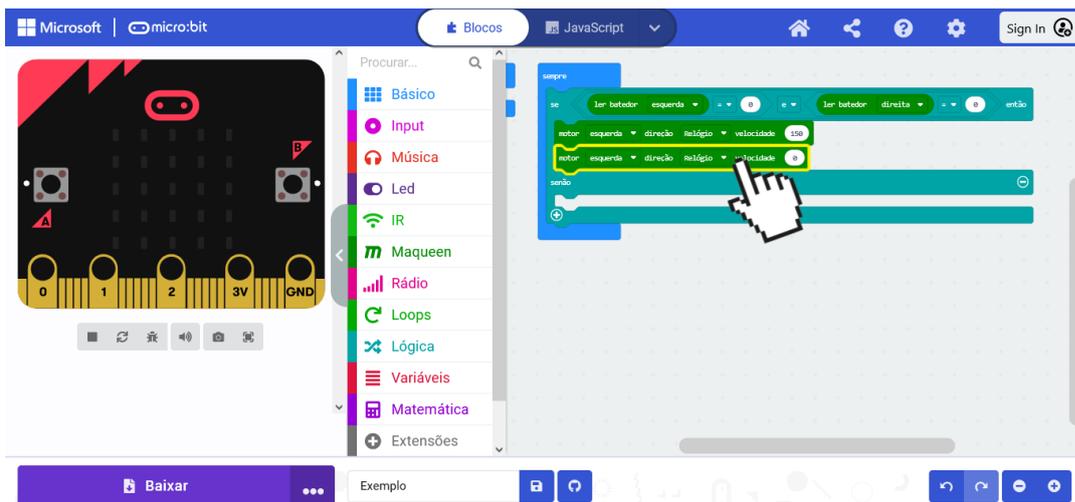


Figura 21.18: Montando um comando.

# Carro segue faixa

16. Em **motor esquerda direção Relógio velocidade 0**, altere motor **esquerda** para **direita** e a velocidade **0**, para **150**, como nas Figuras 21.19 e 21.20;

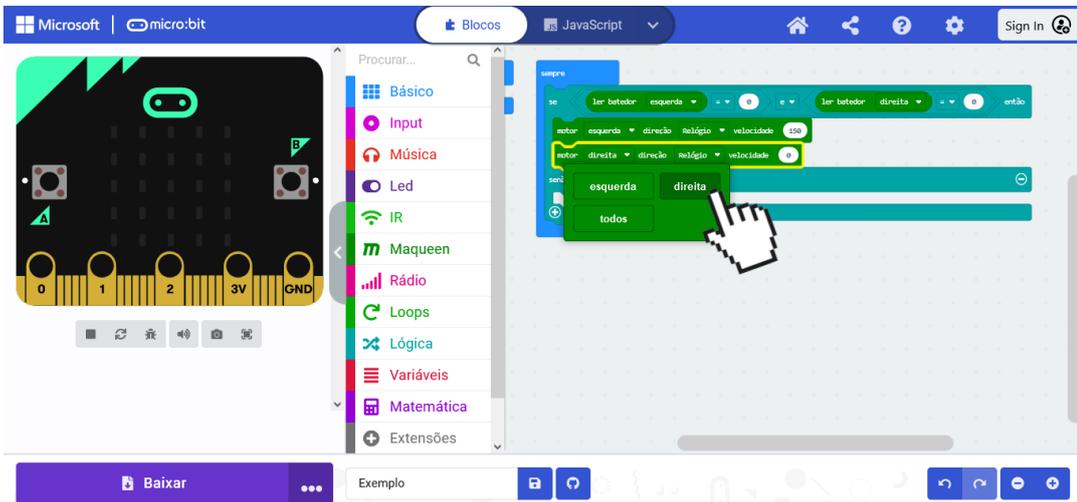


Figura 21.19: Alterando um motor.

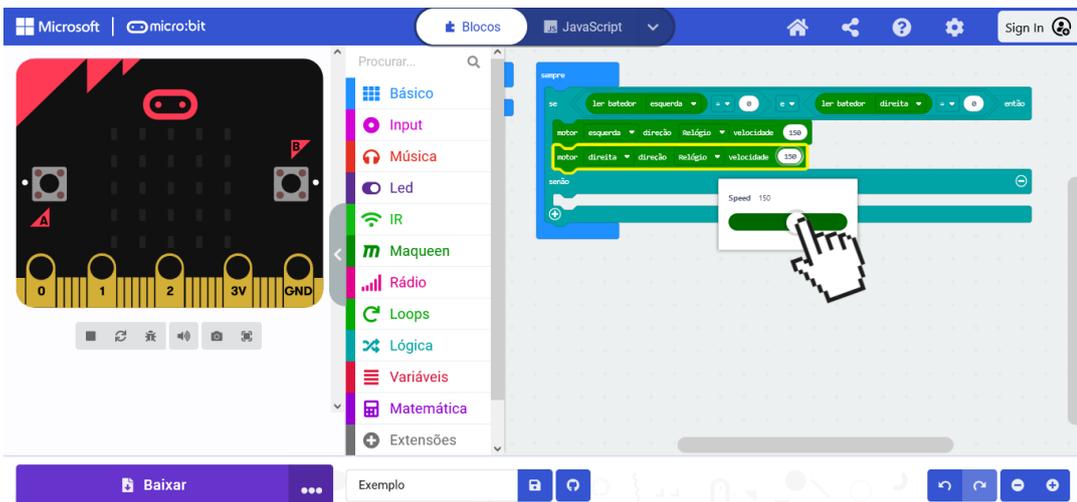


Figura 21.20: Alterando uma velocidade.



# Carro segue faixa

17. Clique no sinal de + que aparece abaixo do **senão** para adicionar ao bloco mais uma condição **senão se < > então**, Figura 21.21;

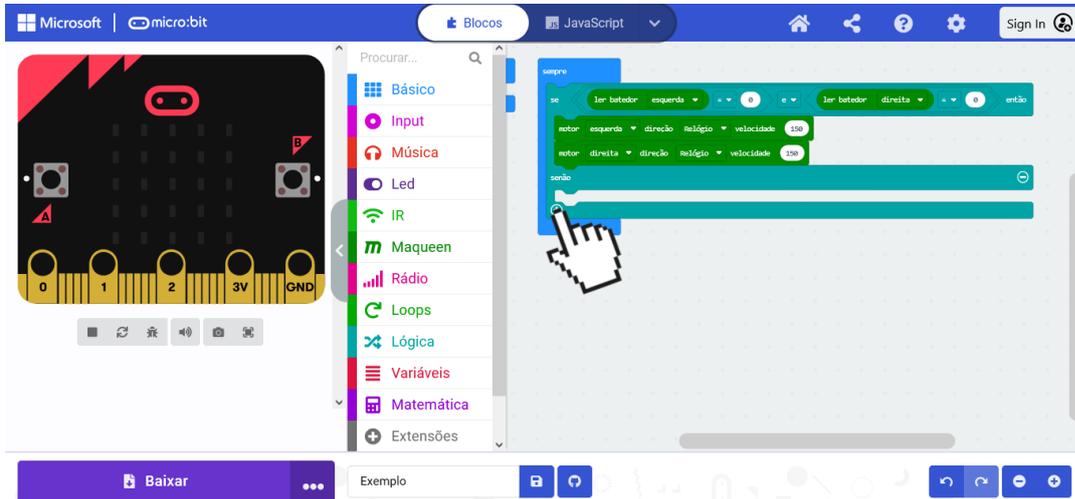


Figura 21.21: Adicionando uma condição.

18. Em **Lógica**, selecione um condicional Booleano **e**, Figura 21.22;



Figura 21.22: Selecionando um comando.

# Carro segue faixa

19. Insira o comando selecionado no hexágono da condição **senão se < > então**, como na Figura 21.23;



Figura 21.23: Montando um comando.

20. Em **Lógica**, selecione duas vezes a comparação **0 = 0**, Figura 21.24;



Figura 21.24: Selecionado comandos.



# Carro segue faixa

- 21.** Insira as comparações nos hexágonos à direita e à esquerda do condicional **e**, como na Figura 21.25;

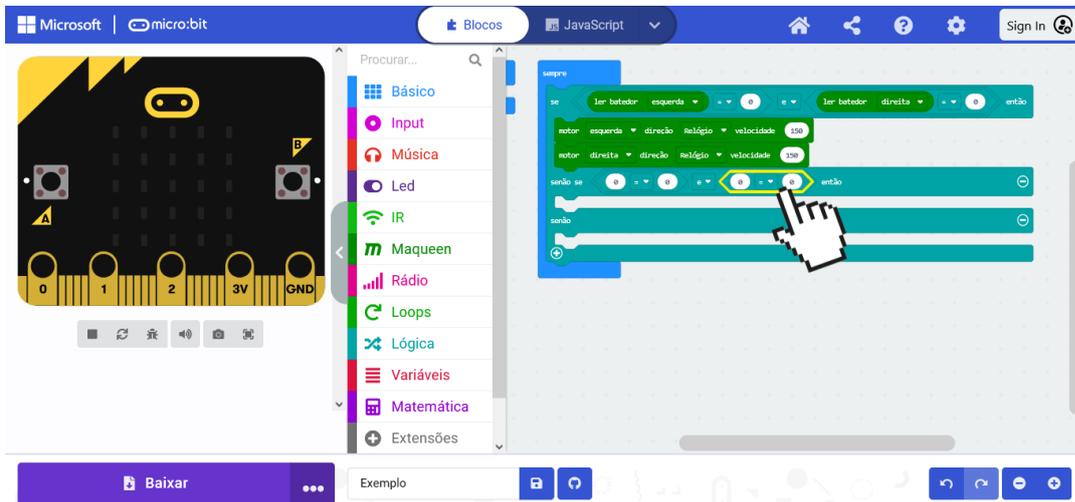


Figura 21.25: Montando comandos.

- 22.** Em **Maqueen**, selecione o comando **ler batedor esquerda**, Figura 21.26;

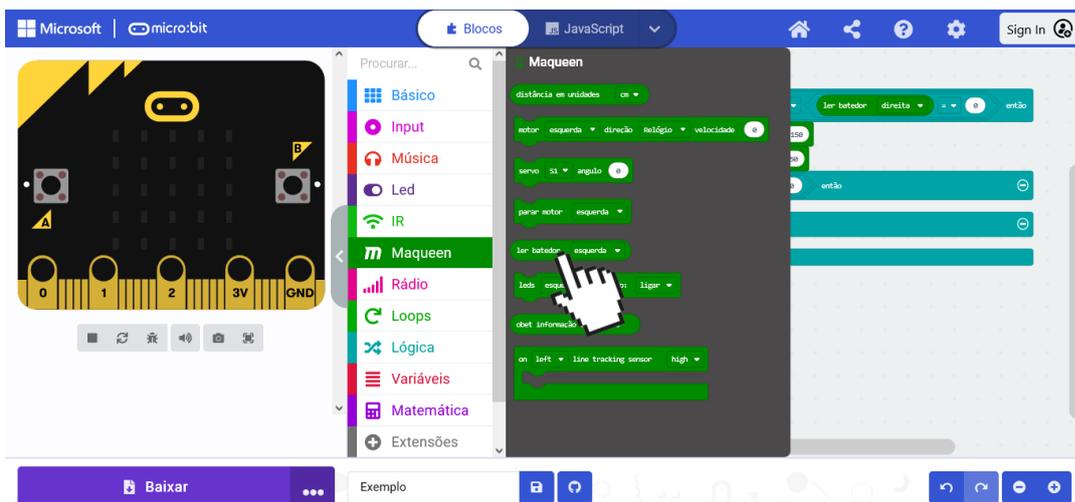


Figura 21.26: Selecionando um comando.

# Carro segue faixa

- 23.** Coloque o último comando selecionado no primeiro **0** da comparação **senão se 0 = 0 e 0 = 0 então**, Figura 21.27;

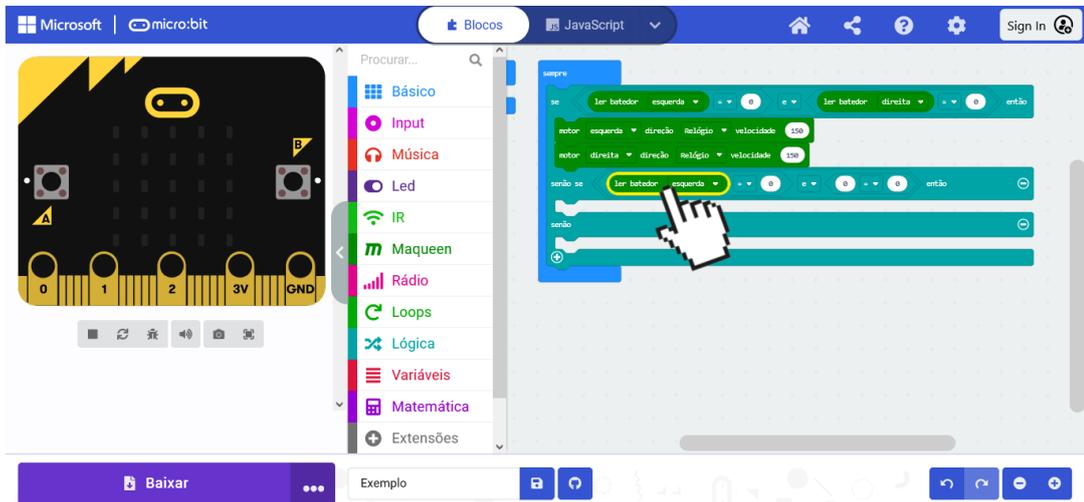


Figura 21.27: Montando um comando.

- 24.** Altere o número **0** por **1**, em **senão se ler batedor esquerda = 0**, Figura 21.28;

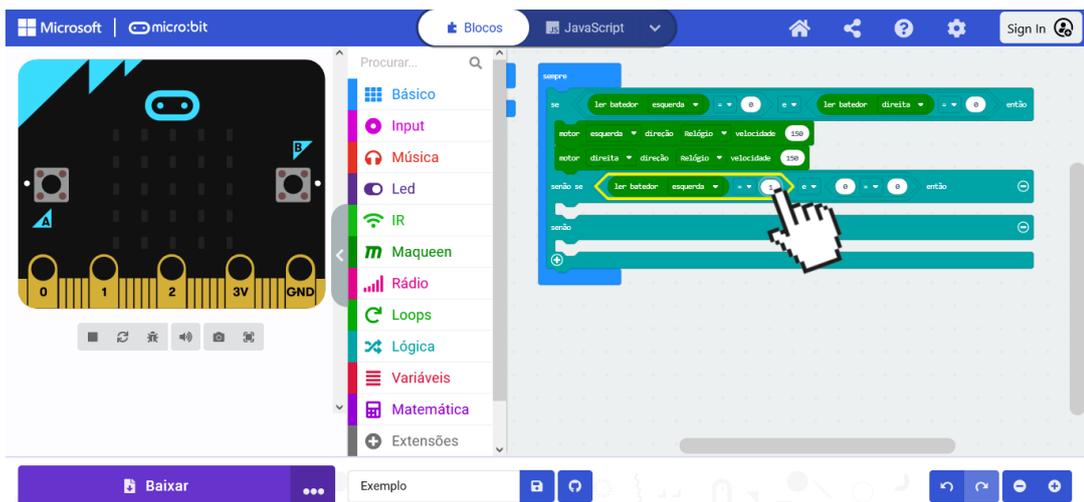


Figura 21.28: Alterando uma comparação.





# Carro segue faixa

- 26.** Altere **esquerda** por **direita** no segundo comando **ler batedor esquerda**, Figura 21.31;

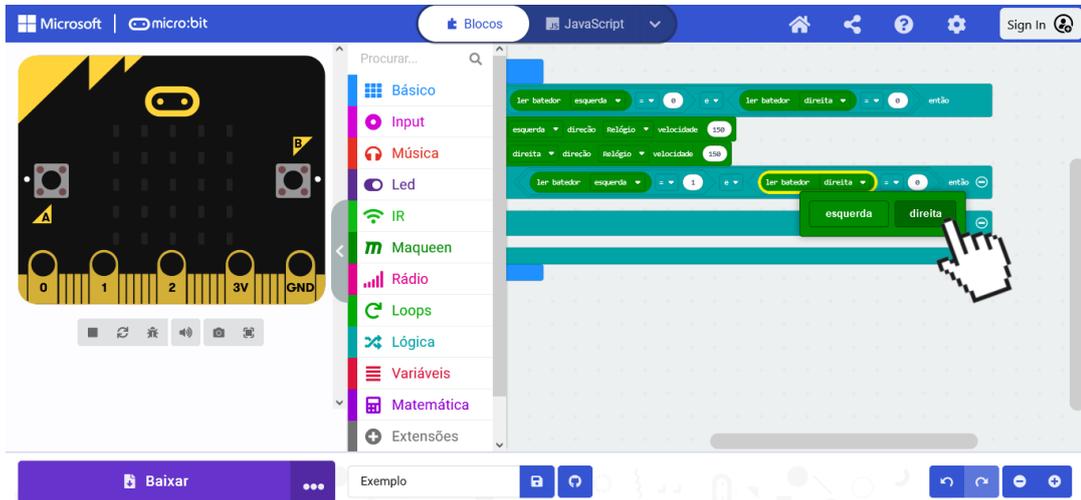


Figura 21.31: Alterando um comando.

- 27.** Em **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0**, Figura 21.32;



Figura 21.32: Selecionando um comando.



# Carro segue faixa

- 28.** Insira o comando selecionado logo abaixo de **senão se ler batedor esquerda = 1 e ler batedor direita = 0 então**, Figura 21.33;



Figura 21.33: Montando um comando.

- 29.** Em **motor esquerda direção Relógio velocidade 0**, altere a velocidade **0** para **150**, Figura 21.34;

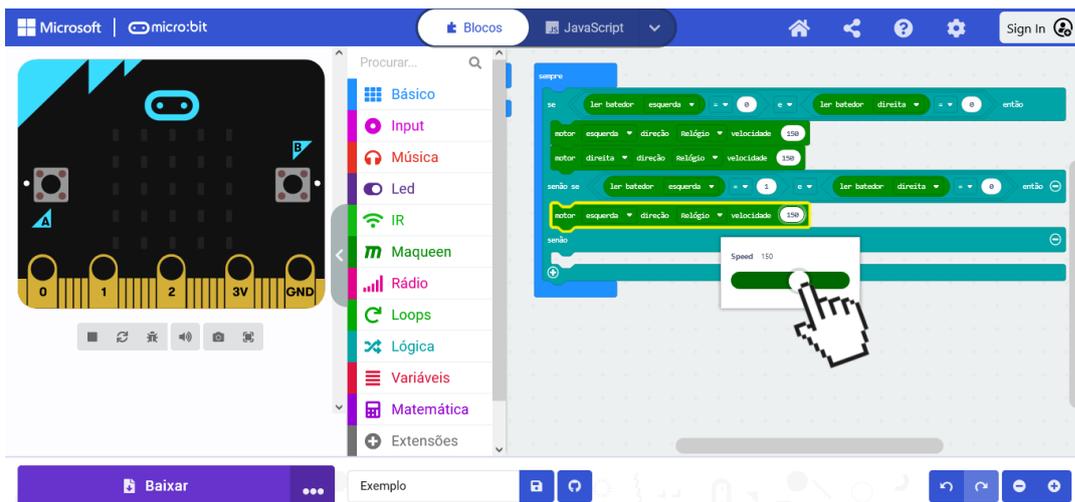


Figura 21.34: Alterando uma velocidade.

# Carro segue faixa

- 30.** Em **Maqueen**, selecione novamente o comando **motor esquerda direção Relógio velocidade 0**, Figura 21.35;

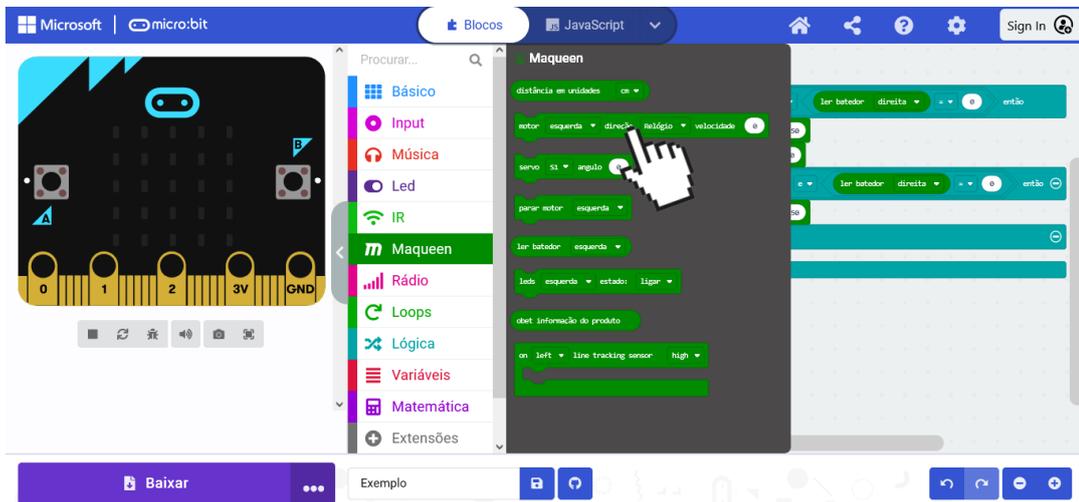


Figura 21.35: Selecionando um comando.

- 31.** Insira o último comando selecionado logo abaixo de **motor esquerda direção Relógio velocidade 150**, Figura 21.36;



Figura 21.36: Montando um comando.



# Carro segue faixa

- 32.** Em **motor esquerda direção Relógio velocidade 0**, altere motor **esquerda** para **direita**, Figura 21.37;



Figura 21.37: Alterando um motor.

# Carro segue faixa

**33.** Clique novamente no sinal de + que está abaixo de **senão**, Figuras 21.38 e 21.39;

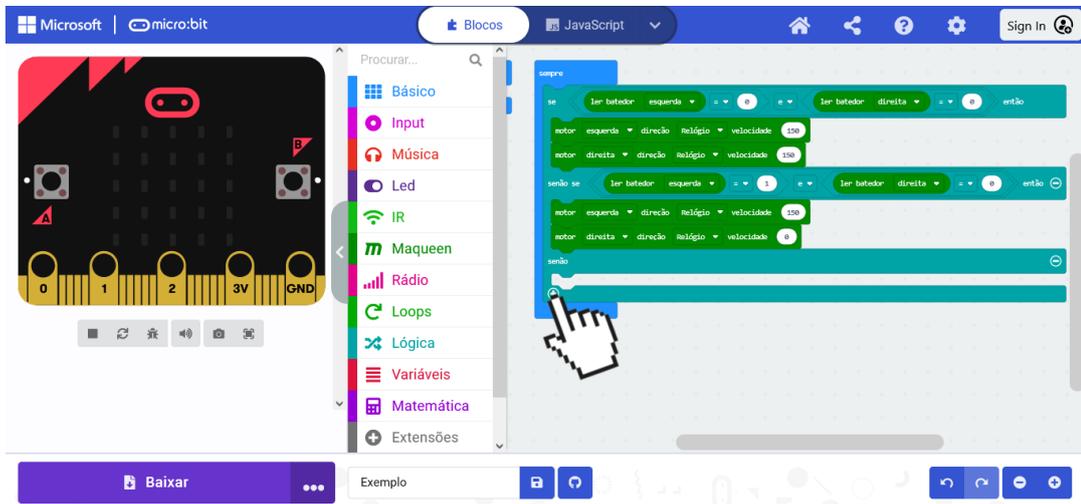


Figura 21.38: Adicionando uma condição.

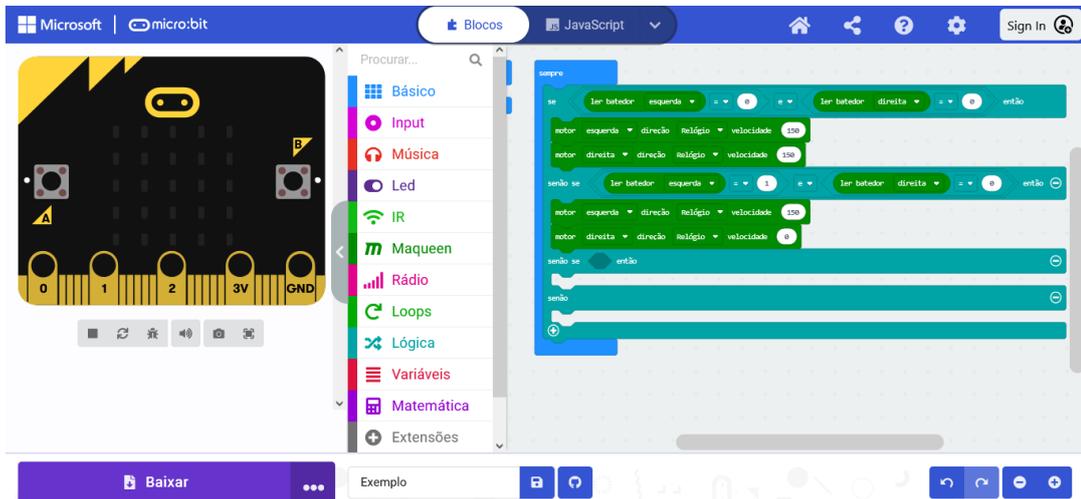


Figura 21.39: Condição adicionada.



# Carro segue faixa

**34.** Em **Lógica**, no condicional Booleano, selecione o comando **<>e<>**, Figura 21.40;



Figura 21.40: Selecionando um comando.

**35.** Insira o comando selecionado no hexágono em **senão se <> então**, Figura 21.41;

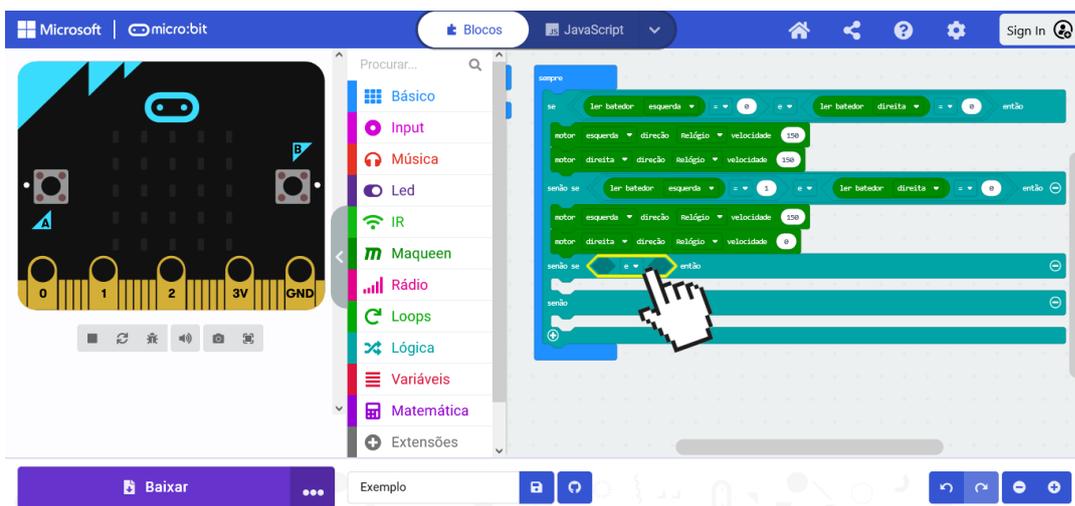


Figura 21.41: Montando um comando.

# Carro segue faixa

**36.** Em **Lógica**, selecione duas vezes a comparação **0 = 0**, Figura 21.42;

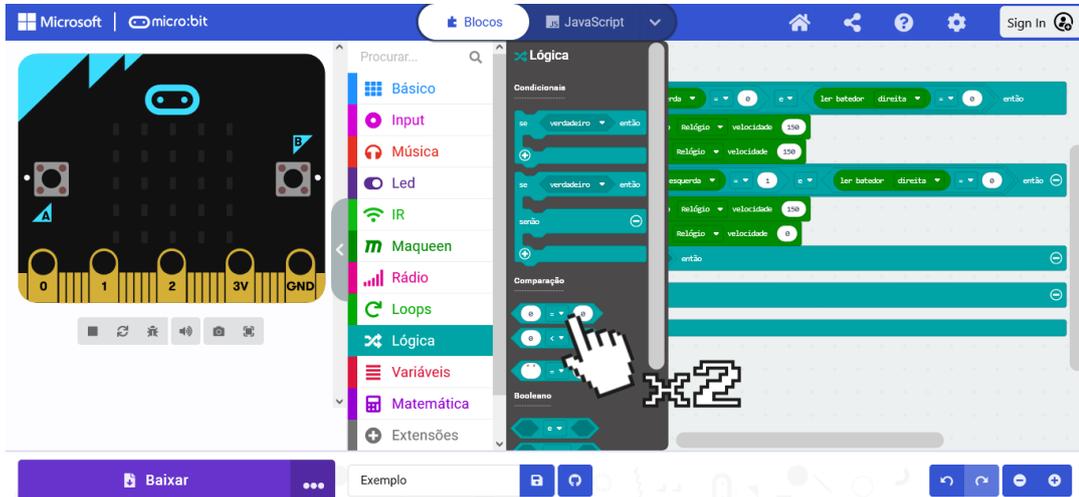


Figura 21.42: Selecionando comandos.

**37.** Insira uma comparação no primeiro hexágono do lado esquerdo de **e** e a outra no segundo, à direita, antes do **então**, Figura 21.43;



Figura 21.43: Montando comandos.



# Carro segue faixa

**38.** Em **Maqueen**, selecione o comando **ler batedor esquerda**, Figura 21.44;

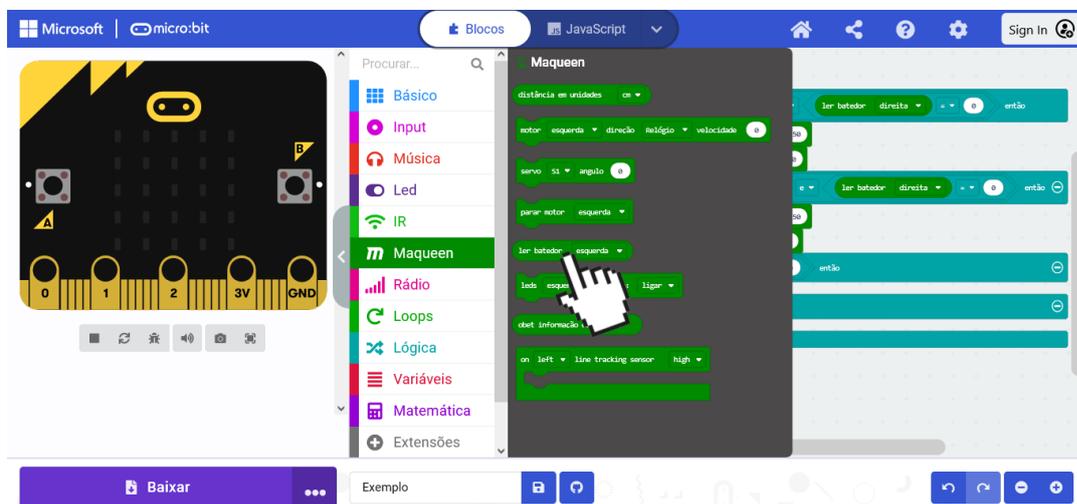


Figura 21.44: Selecionando um comando.

**39.** Insira o comando selecionado anteriormente no primeiro **0**, de **senão se 0 = 0** e **0 = 0 então**, Figura 21.45;



Figura 21.45: Montando um comando.

# Carro segue faixa

40. Novamente em **Maqueen**, selecione outro comando **ler batedor esquerda** e coloque-o sobre o número **0** que está logo depois de **e**, como nas Figuras 21.46 e 21.47;

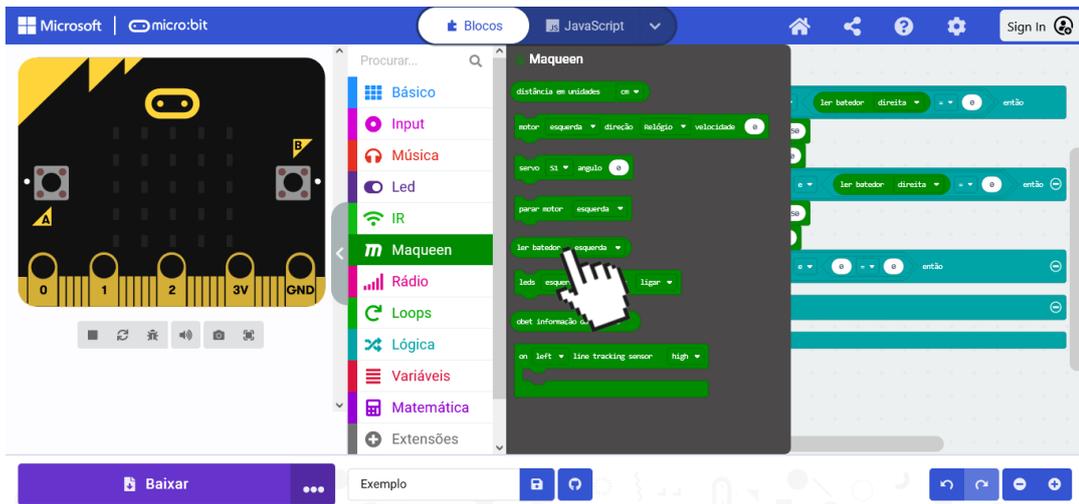


Figura 21.46: Selecionando um comando.

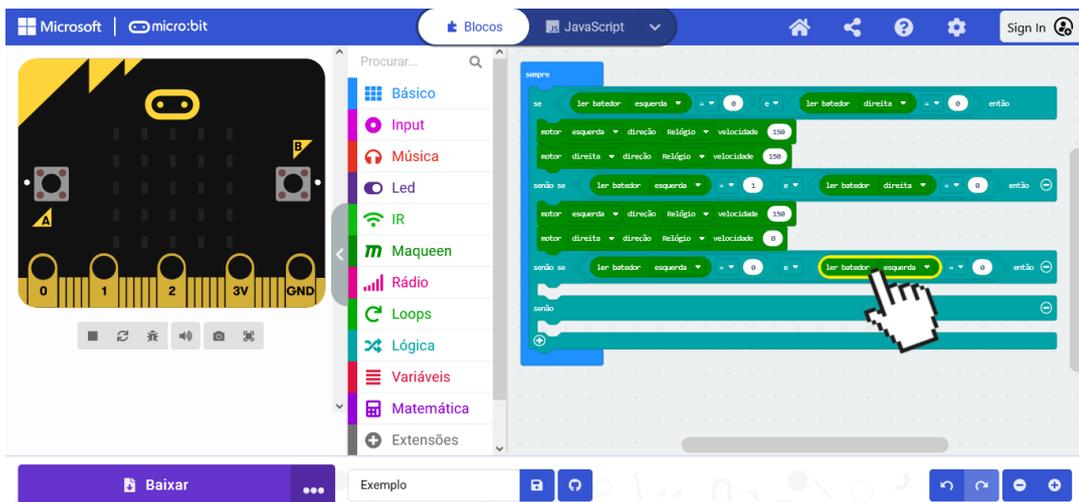


Figura 21.47: Montando um comando.



# Carro segue faixa

- 41.** No segundo comando **ler batedor esquerda**, altere **esquerda** para **direita**, como na Figura 21.48;

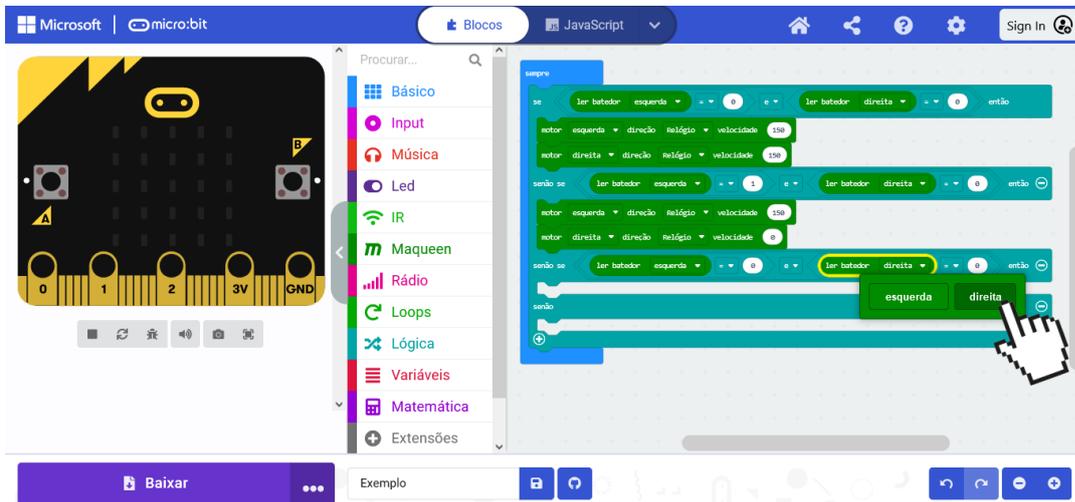


Figura 21.48: Alterando um leitor.

- 42.** No comando **ler batedor direita**, altere o número **0** para **1**, Figura 21.49;



Figura 21.49: Alterando uma comparação.

# Carro segue faixa

- 43.** Em **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0**, Figura 21.50;

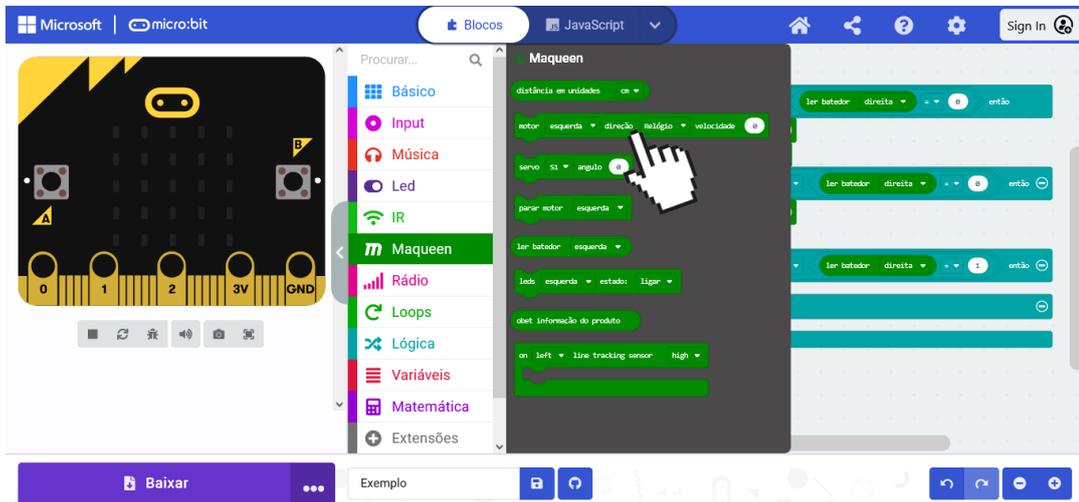


Figura 21.50: Selecionando um comando.

- 44.** Insira-o logo abaixo de **senão se ler batedor esquerda = 0 e ler batedor direita = 1 então**, Figura 21.51;

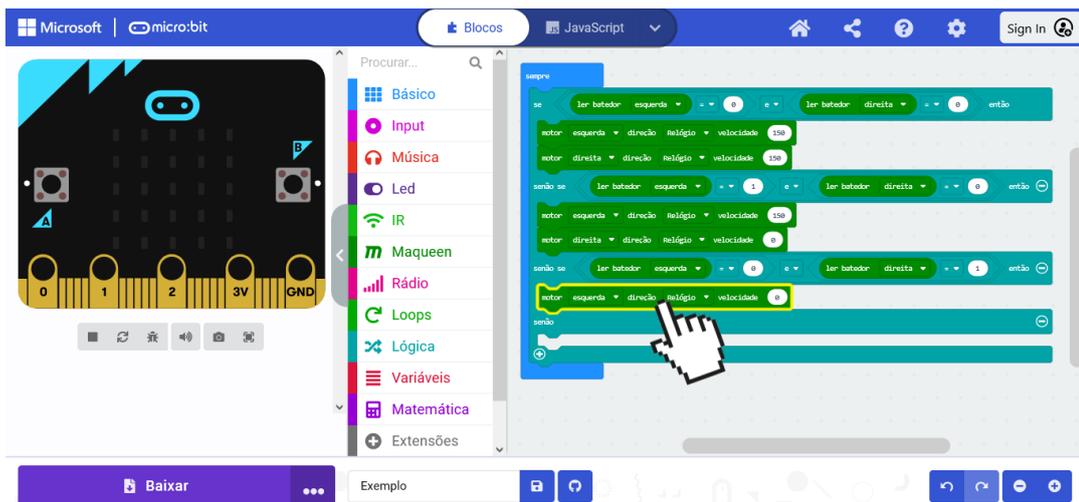


Figura 21.51: Montando um comando.



# Carro segue faixa

45. No último comando inserido, altere motor **esquerda** para **direita** e a velocidade **0** para **150**, como nas Figuras 21.52 e 21.53;

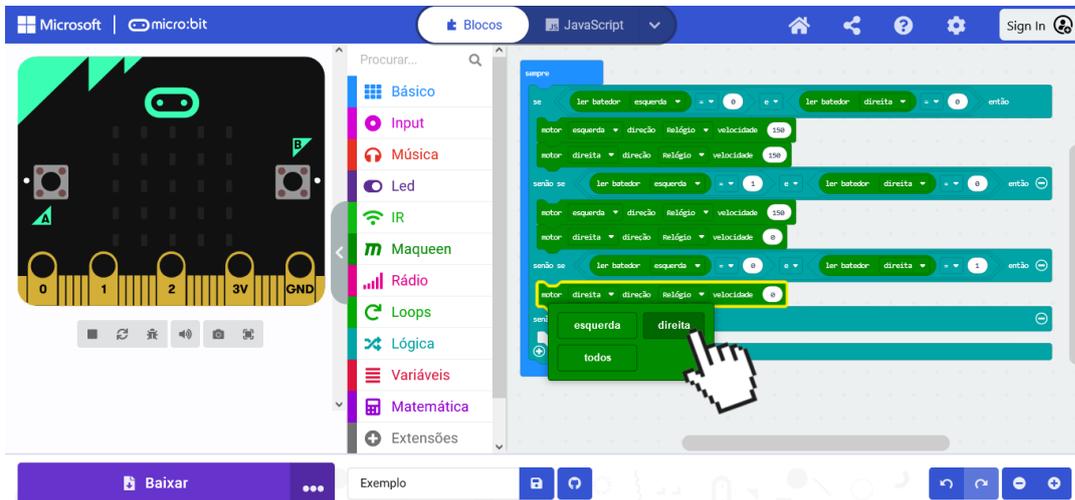


Figura 21.52: Alterando um motor.

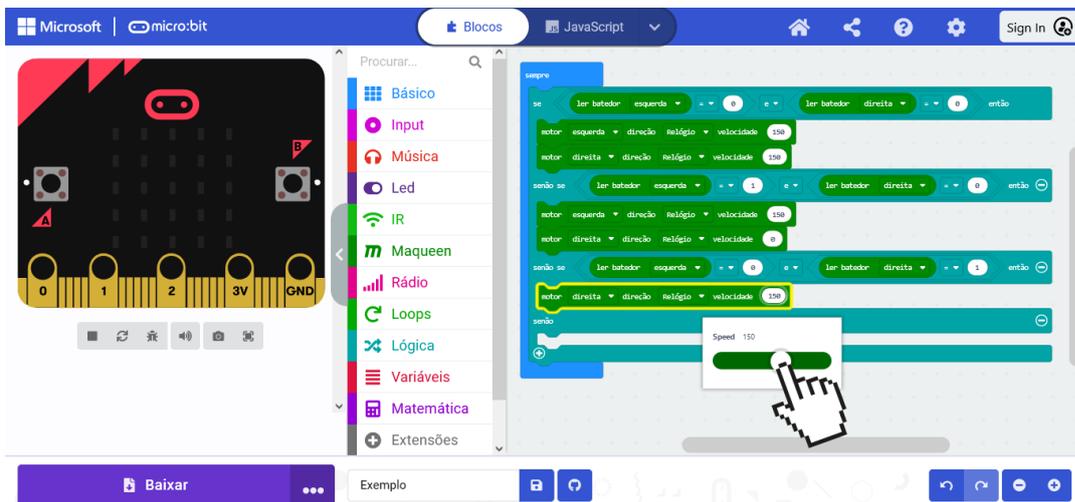


Figura 21.53: Alterando uma velocidade.

# Carro segue faixa

- 46.** Na aba **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0**, Figura 21.54;

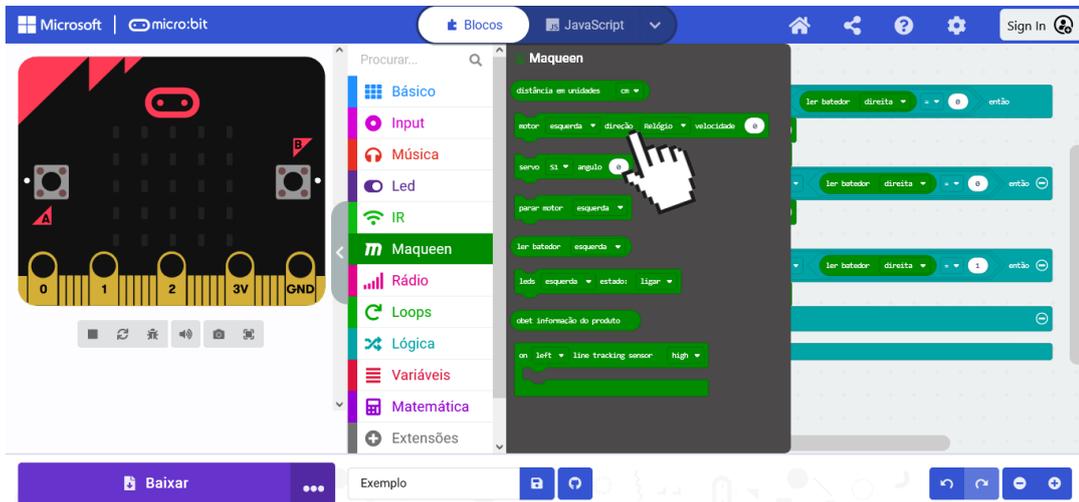


Figura 21.54: Selecionando um comando.

- 47.** Insira-o logo abaixo de **motor direita direção Relógio velocidade 150**, Figura 21.55;

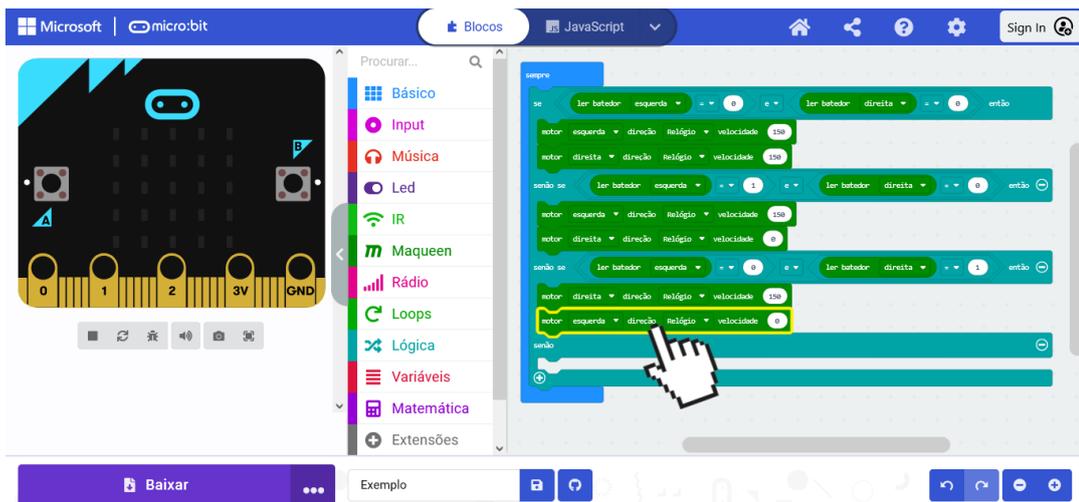


Figura 21.55: Montando um comando.



# Carro segue faixa

48. Clique no sinal de +, abaixo de **senão**, para adicionar mais uma condição, Figura 21.56;

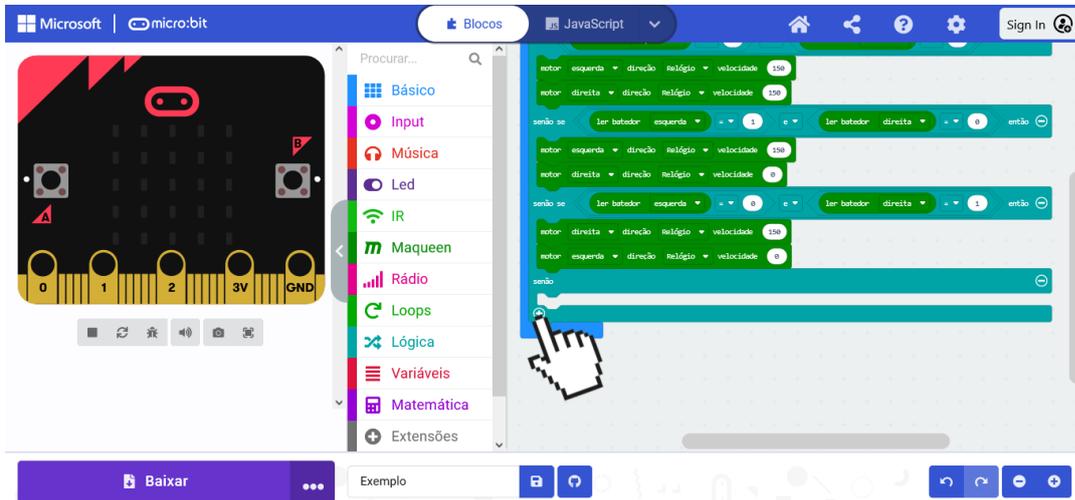


Figura 21.56: Adicionando uma condição.

49. Em **Lógica**, no condicional Booleano, selecione o comando **e**, Figura 21.57;

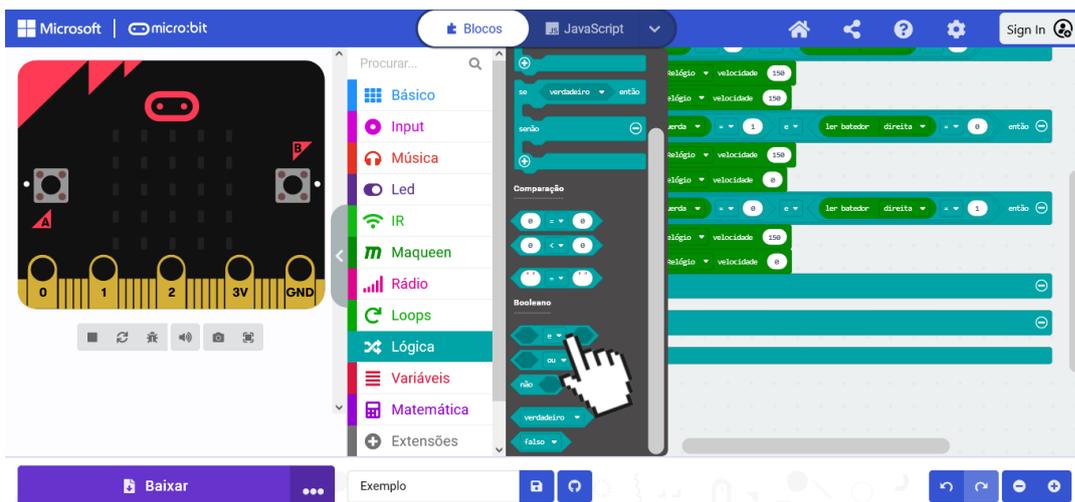


Figura 21.57: Selecionando um comando.

# Carro segue faixa

50. Insira o comando selecionado anteriormente no hexágono em **senão se < > então**, Figura 21.58;

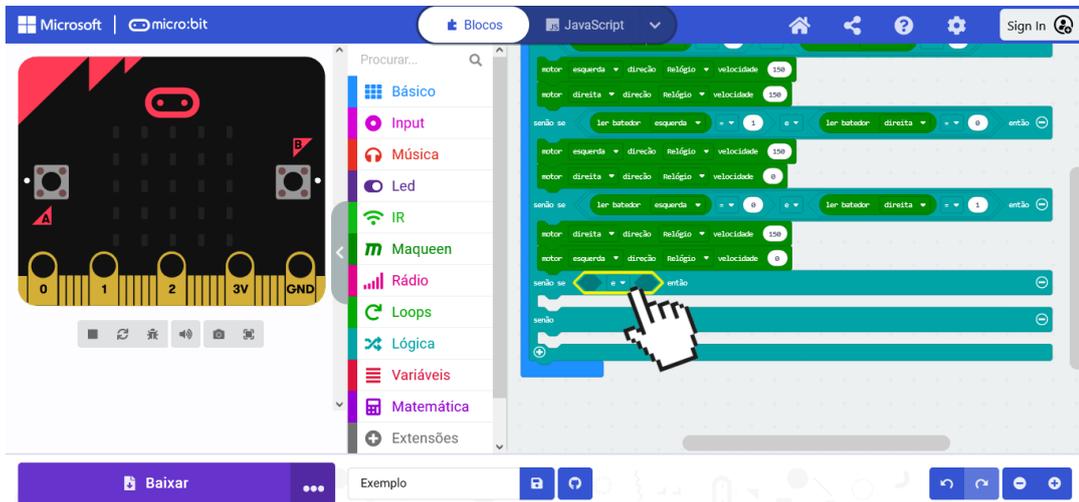


Figura 21.58: Montando uma comando.

51. Em **Lógica**, selecione duas vezes a comparação **0 = 0**, Figura 21.59;

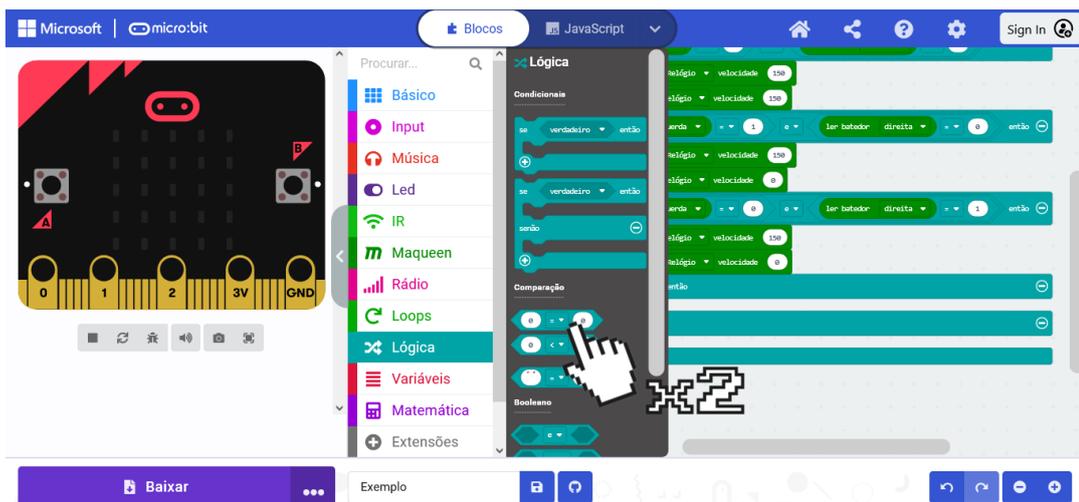


Figura 21.59: Selecionando comandos.



# Carro segue faixa

- 52.** Insira as comparações selecionadas em cada um dos hexágonos ao lado do condicional booleano **e**, como na Figura 21.60;

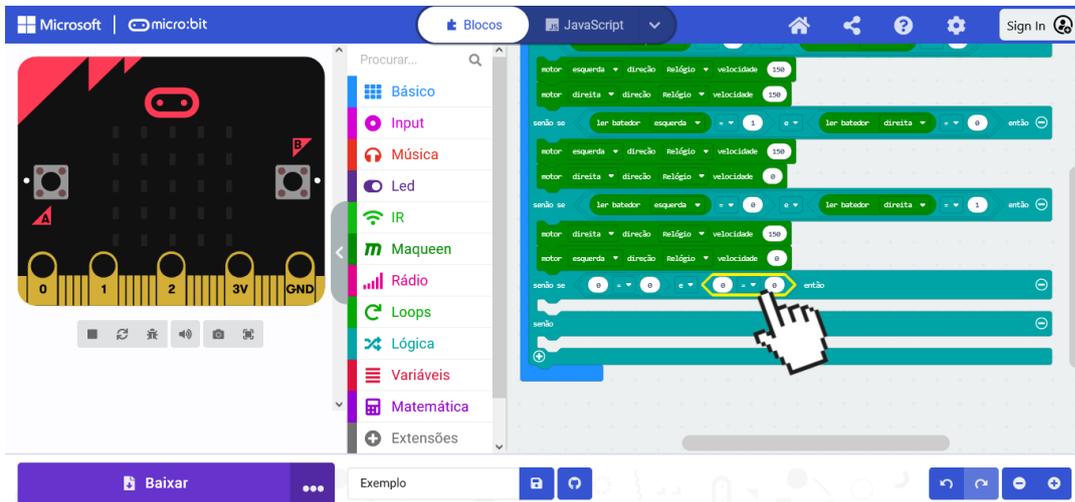


Figura 21.60: Montando comandos.

- 53.** Em **Maqueen**, selecione **ler batedor esquerda**, Figura 21.61;

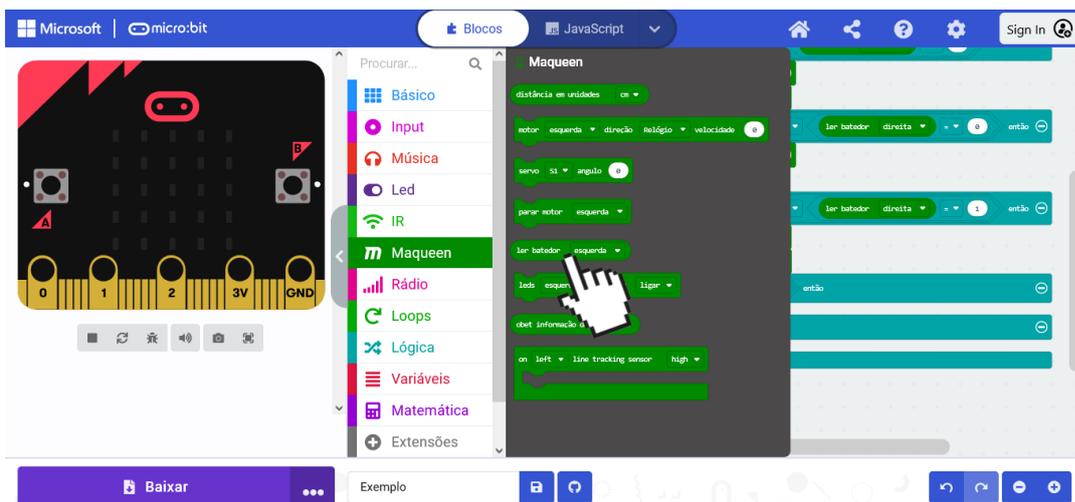


Figura 21.61: Selecionando um comando.

# Carro segue faixa

- 54.** Insira o comando selecionado anteriormente no lugar do primeiro **0**, de **senão se 0 = 0**, Figura 21.62;

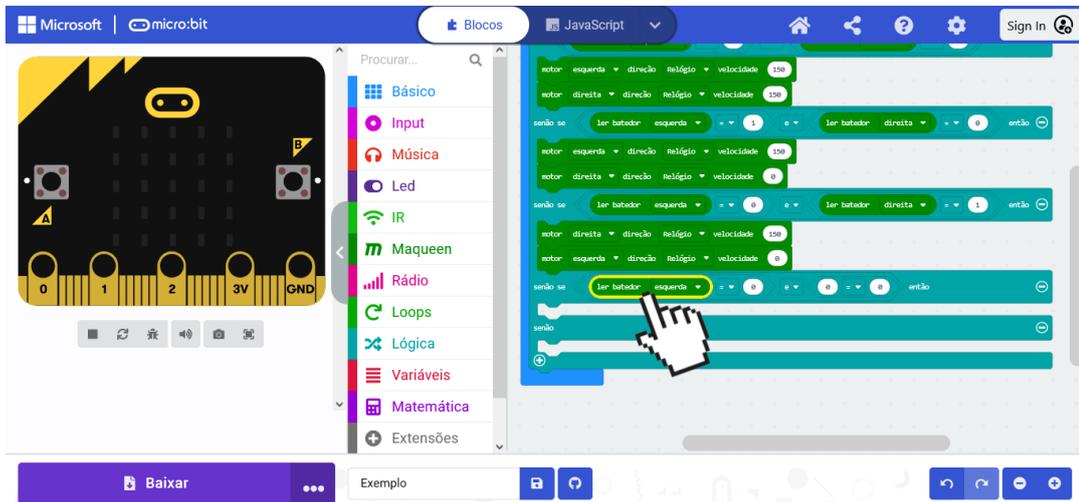


Figura 21.62: Montando um comando.

- 55.** Altere o número **0** de **senão se ler batedor esquerda = 0**, para **1**, Figura 21.63;

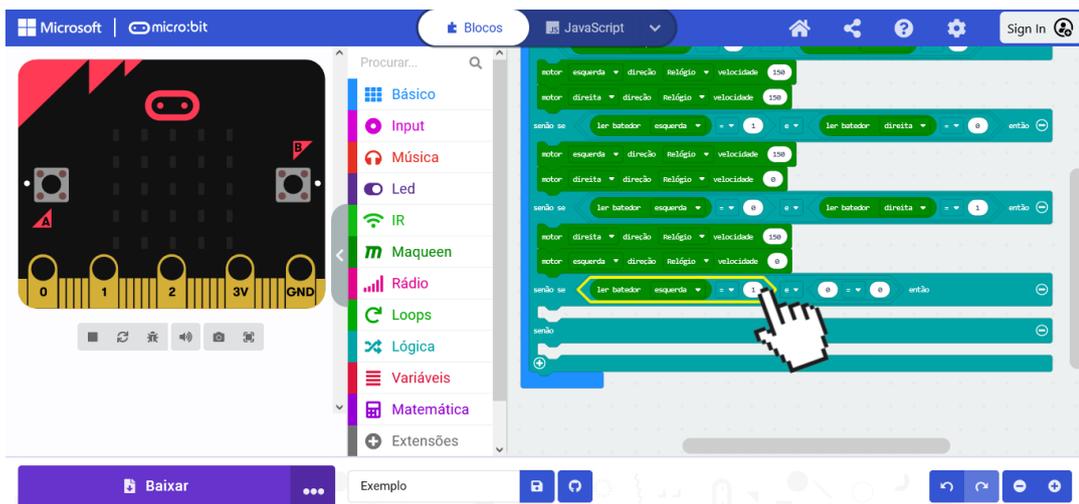


Figura 21.63: Alterando uma comparação.



# Carro segue faixa

56. Novamente na aba **Maqueen**, selecione mais um comando **ler batedor esquerda** e coloque-o no lugar do **0**, após o condicional booleano **e**, Figuras 21.64 e 21.65;

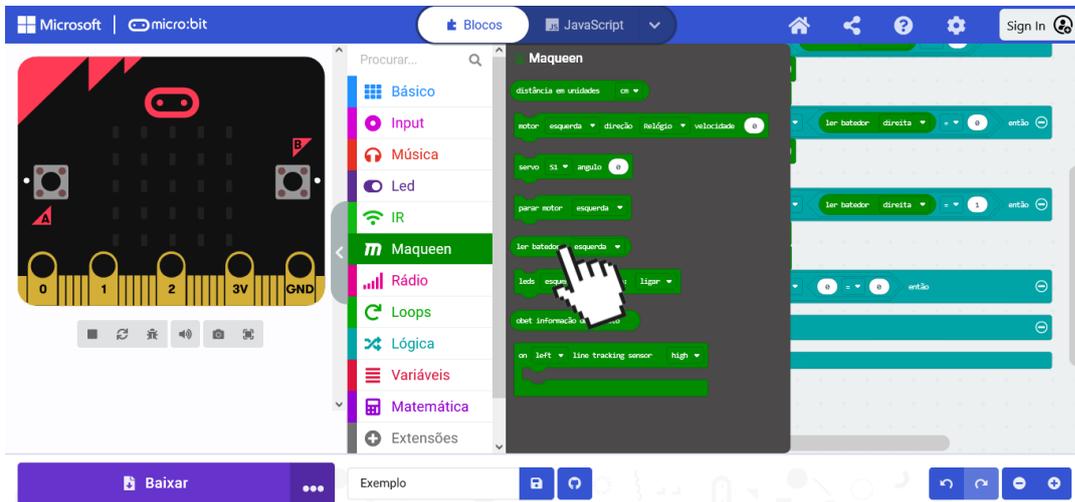


Figura 21.64: Selecionando um comando.

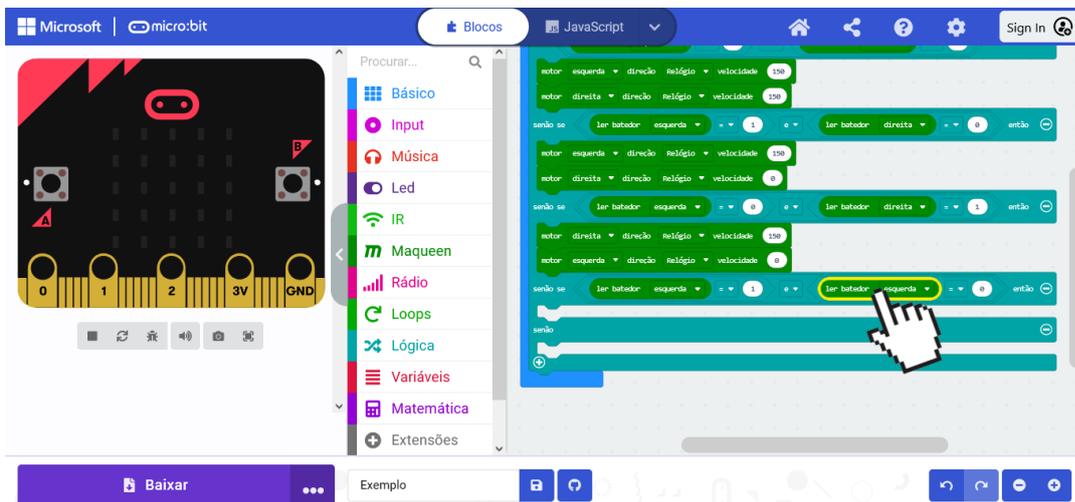


Figura 21.65: Montando um comando.

# Carro segue faixa

**57.** Altere o batedor **esquerda** para **direita**, no segundo **ler batedor**, Figura 21.66;

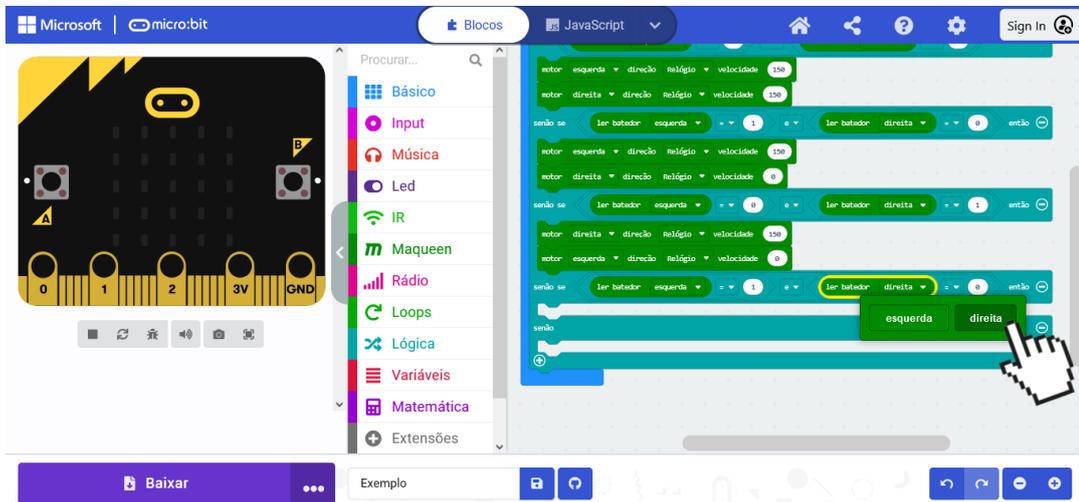


Figura 21.66: Alterando um leitor.

**58.** Altere o número **0** para **1**, como na Figura 21.67;

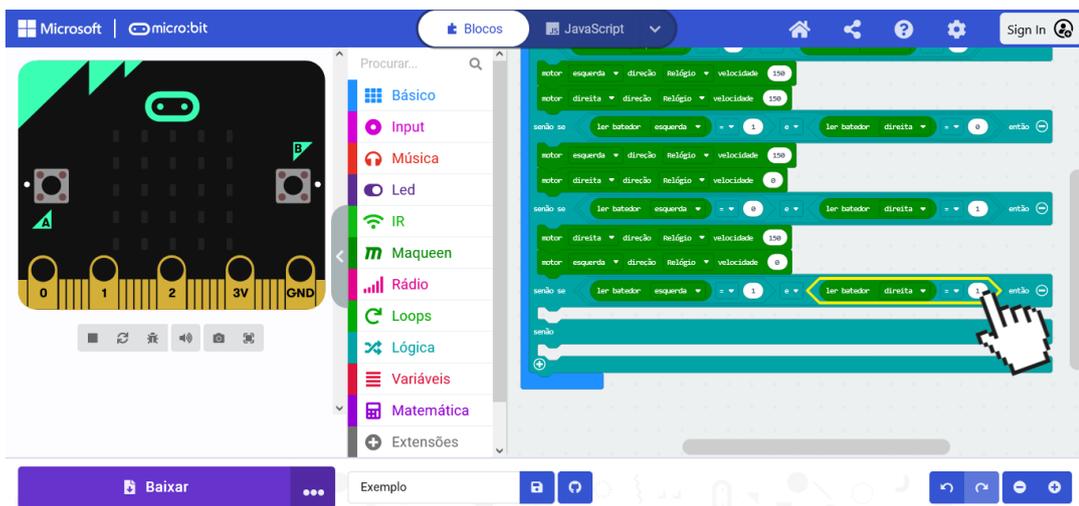


Figura 21.67: Alterando uma condição.



# Carro segue faixa

59. Em **Maqueen**, selecione **motor esquerda direção Relógio velocidade 0**, Figura 21.68;

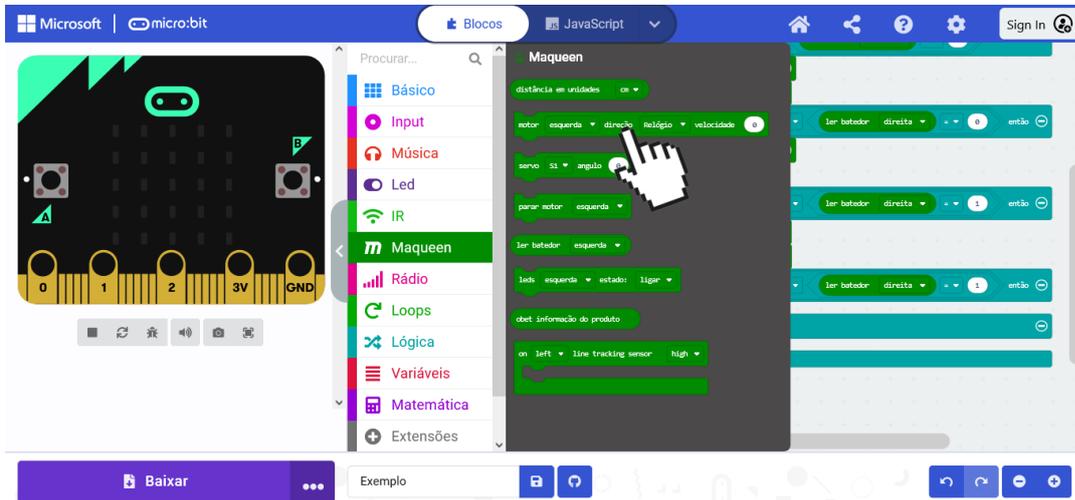


Figura 21.68: Selecionando um comando.

60. Insira o comando selecionado em **senão se ler batedor esquerda = 1 e ler batedor direita = 1 então**, Figura 21.69;

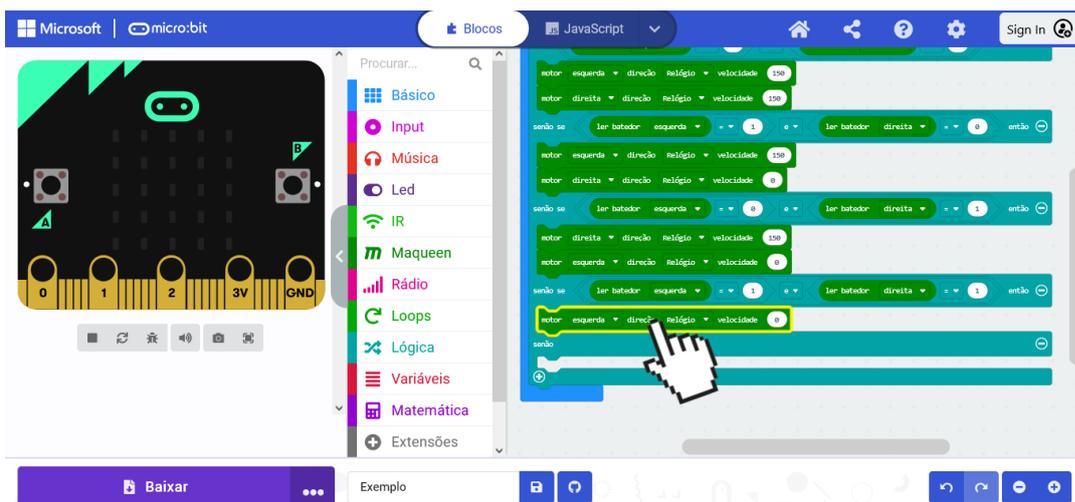


Figura 21.69: Montando um comando.

# Carro segue faixa

61. Em **motor esquerda direção Relógio velocidade 0**, altere motor **esquerda** para **direita** e a velocidade **0** para **150**, como nas Figuras 21.70 e 21.71;

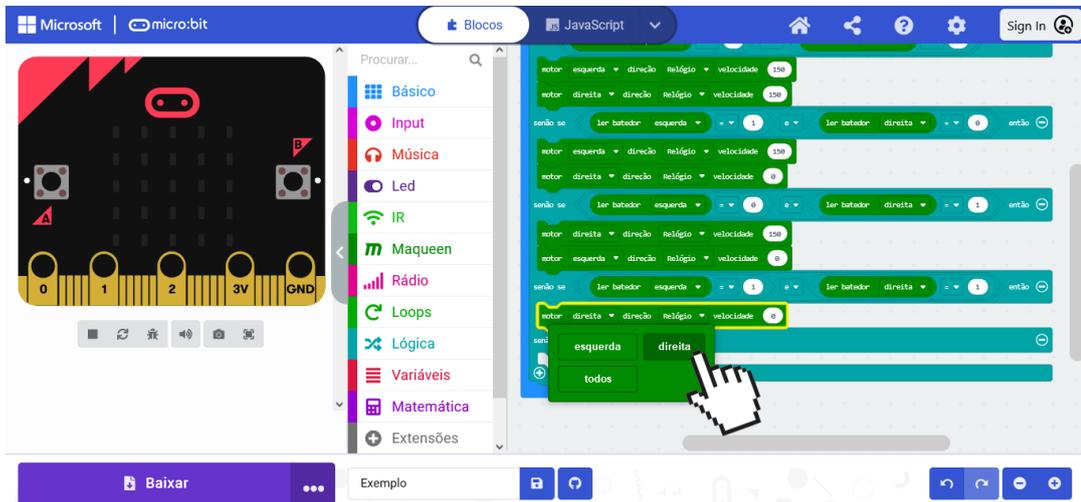


Figura 21.70: Alterando um motor.

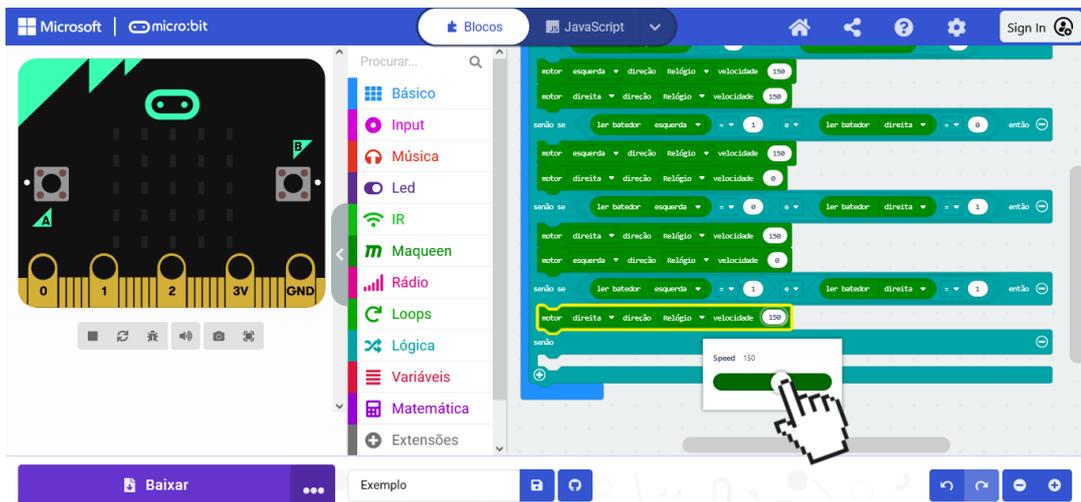


Figura 21.71: Alterando uma velocidade.



# Carro segue faixa

- 62.** Em **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0**, Figura 21.72;

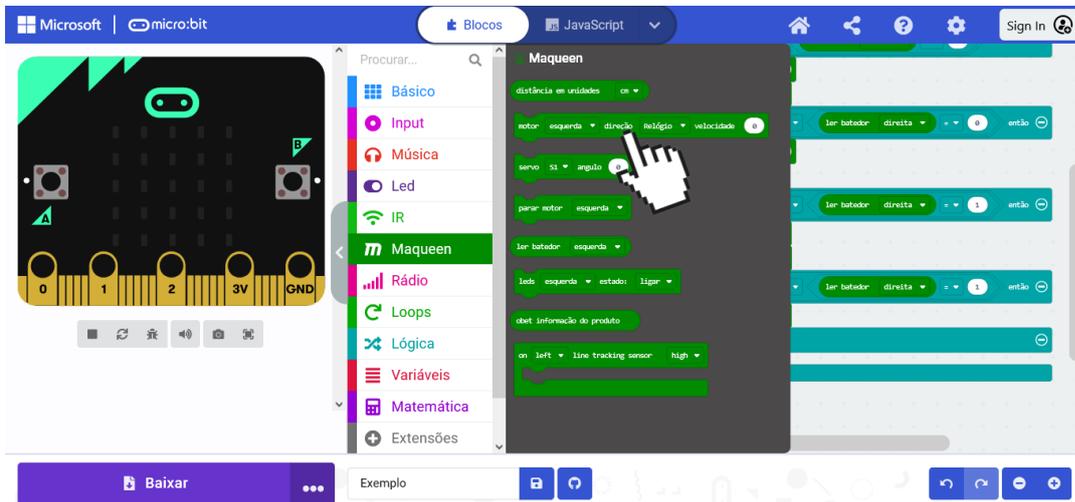


Figura 21.72: Selecionando um comando.

- 63.** Insira-o logo abaixo de **motor direita direção Relógio velocidade 150**, Figura 21.73;

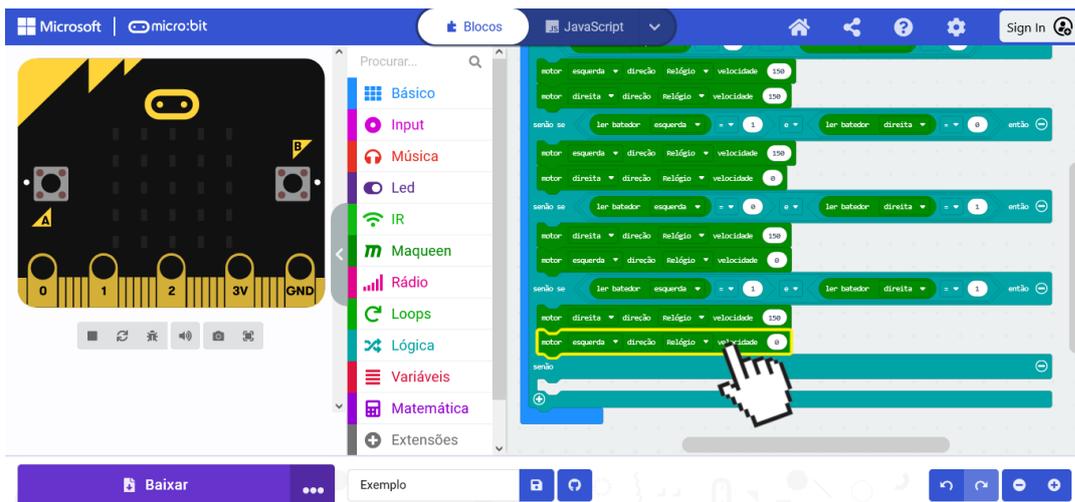


Figura 21.73: Montando um comando.

# Carro segue faixa

- 64.** Logo abaixo de **senão**, há um sinal de -. Clique sobre ele para excluir o trecho excedente do código, como nas Figuras 21.74 e 21.75.

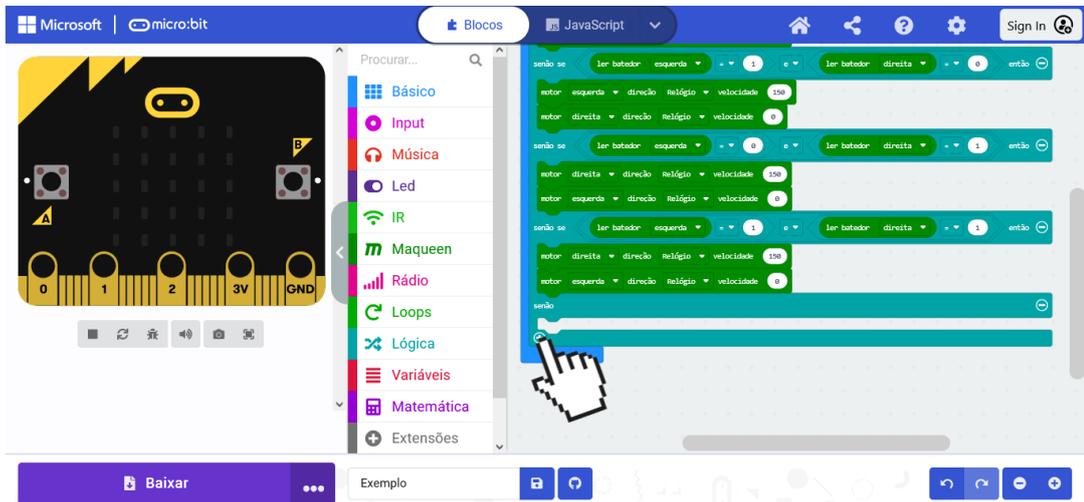


Figura 21.74: Excluindo uma condição.

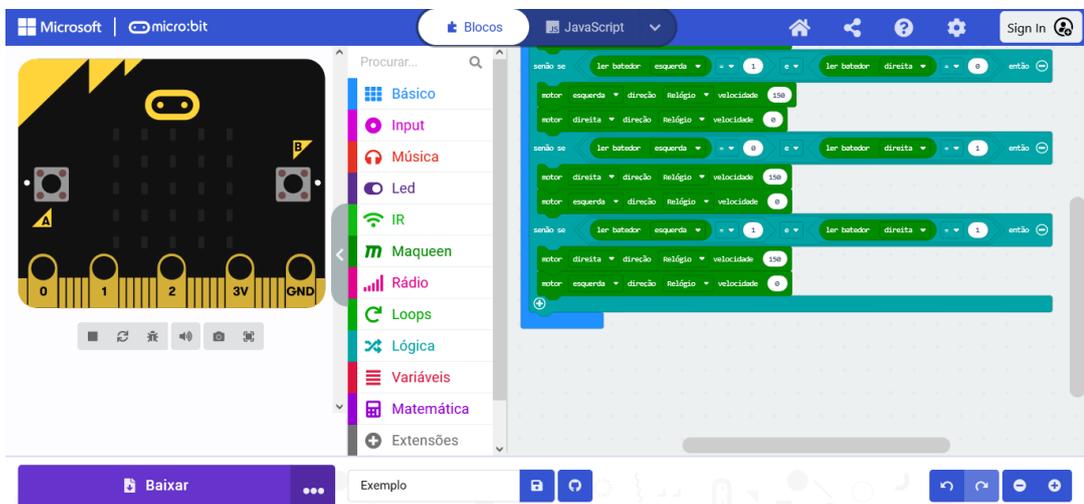


Figura 21.75: Condição excluída.

**Baixe o programa para o micro:bit e encaixe a placa no carrinho.**

[Clique aqui para acessar o código.](#)

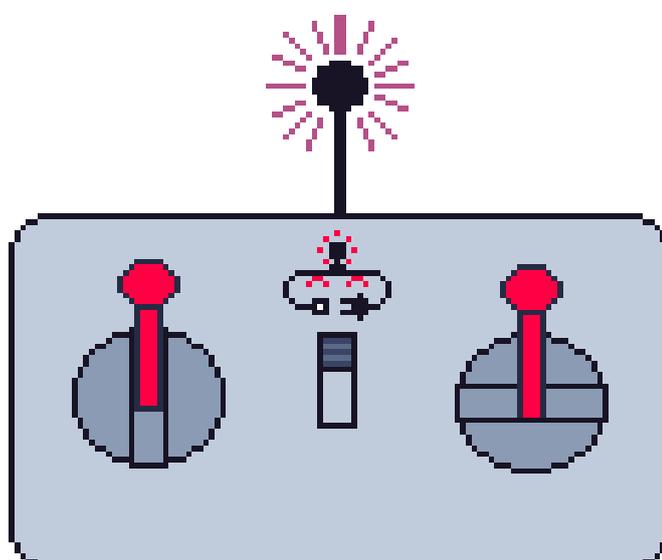




Carro segue faixa



# Capítulo 22: Carrinho de controle remoto



Já pensou como seria legal você mesmo criar um carrinho de controle remoto? Nesta atividade, programaremos um carrinho de controle remoto iguais aos comprados em lojas. Mas aqui a diversão já começa ao iniciarmos a programação.



# Carrinho de controle remoto

## Passo a passo:

1. Clique na aba **Extensões**, e busque por **Maqueen** na barra de pesquisa, como nas Figuras 22.1, 22.2 e 22.3;

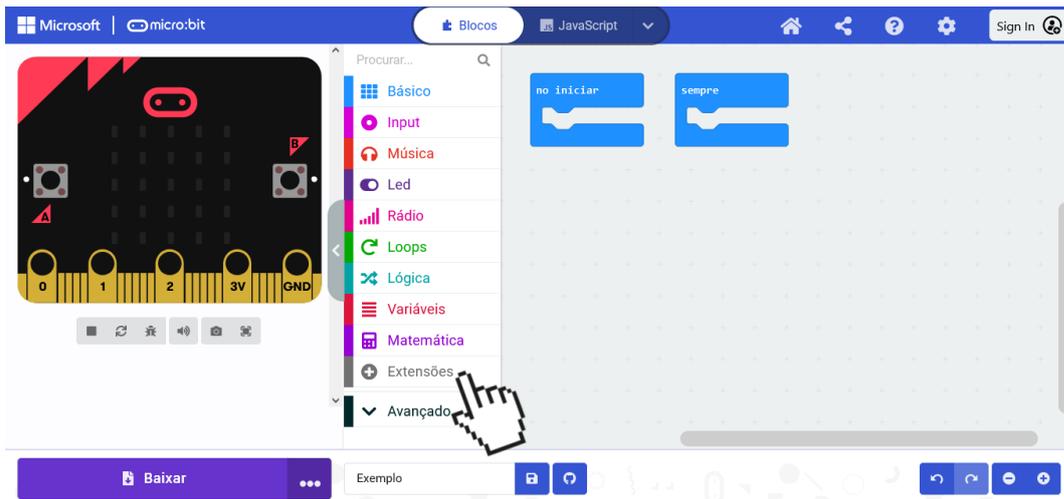


Figura 22.1: Adicionando uma extensão.

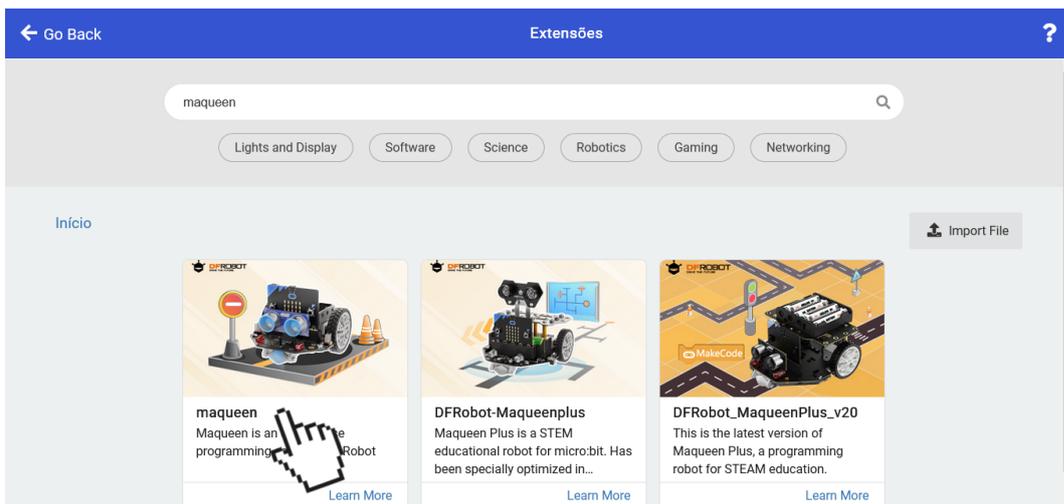


Figura 22.2: Selecionando uma extensão.

# Carrinho de controle remoto



Figura 22.3: Novas bibliotecas adicionadas.

2. Em **Rádio**, selecione o comando **definir grupo do rádio 1** e coloque-o dentro do bloco de acionamento **no iniciar**, Figuras 22.4 e 22.5;

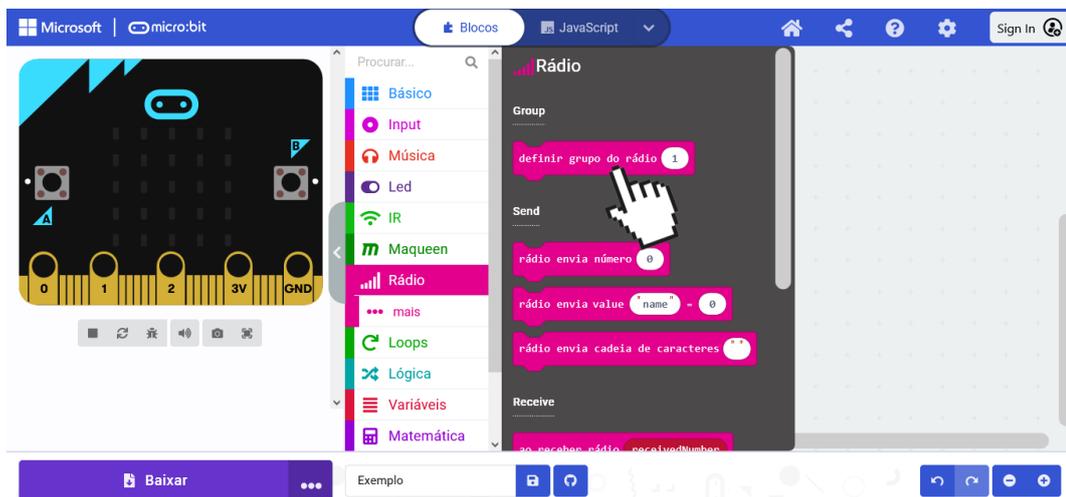
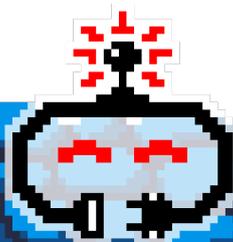


Figura 22.4: Selecionando um comando.



# Carrinho de controle remoto

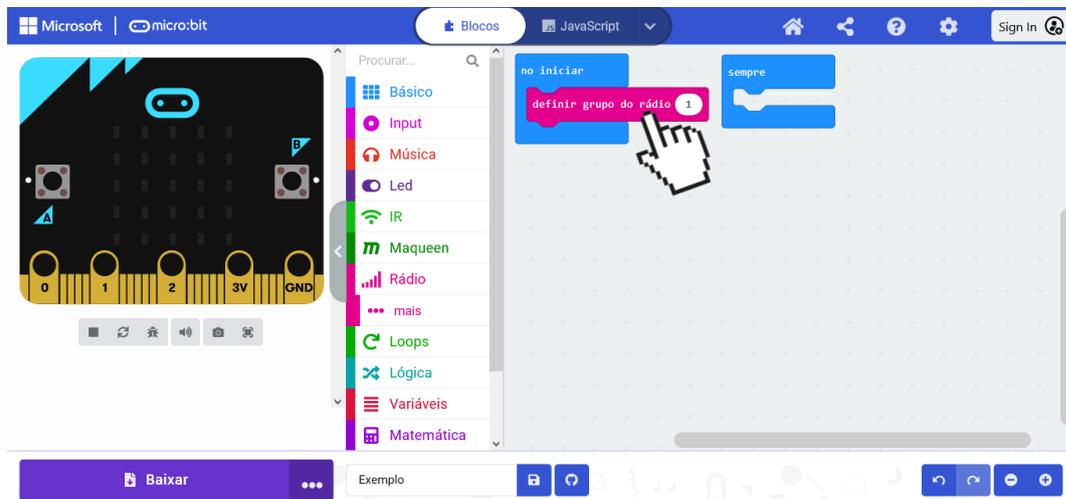


Figura 22.5: Montando um comando.

3. Nesta atividade, não será utilizado o bloco de acionamento **sempre**. Então, arraste-o para a lixeira, se preferir. Na aba **Rádio** novamente, selecione o bloco de acionamento **ao receber rádio receivedNumber**, Figura 22.6;

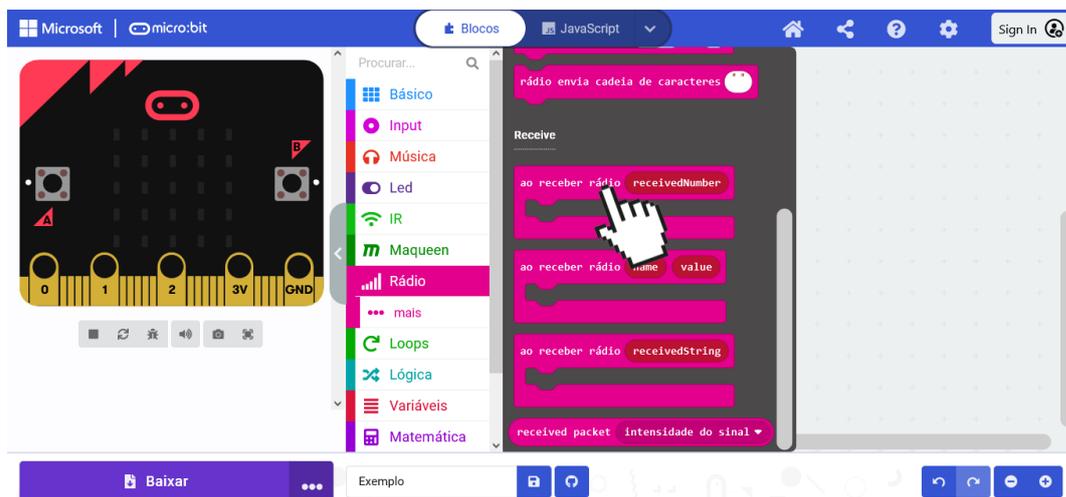


Figura 22.6: Selecionando um bloco de acionamento.

# Carrinho de controle remoto

4. Em **Lógica**, selecione o bloco **se verdadeiro então, senão**, Figura 22.7;

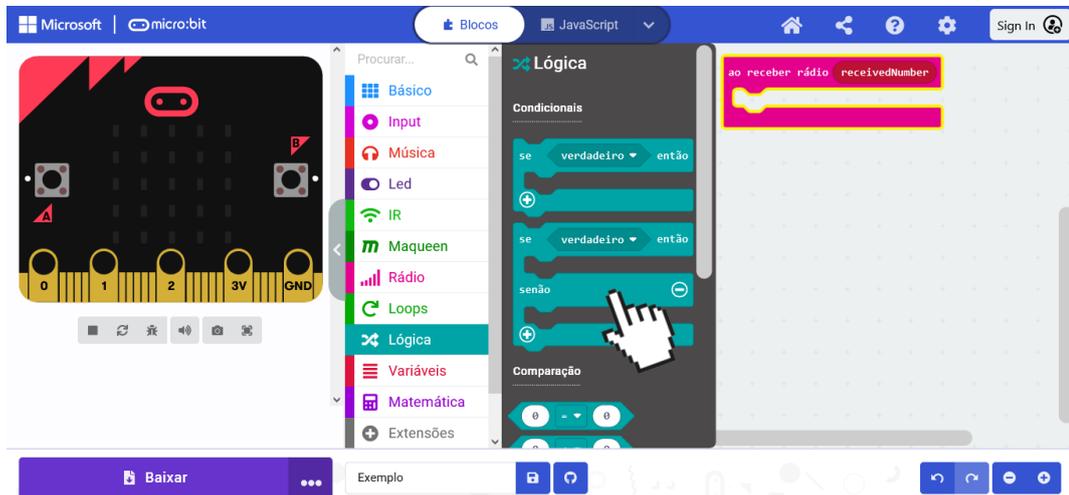


Figura 22.7: Selecionando um bloco de lógica.

5. Insira o bloco selecionado dentro do bloco de acionamento **ao receber rádio receivedNumber**, Figura 22.8;



Figura 22.8: Montando um bloco.



# Carrinho de controle remoto

6. Novamente em **Lógica**, selecione a comparação **0 = 0**, Figura 22.9;



Figura 22.9: Selecionando uma comparação.

7. Coloque a comparação selecionada no lugar de **verdadeiro** em **se verdadeiro então**, Figura 22.10;

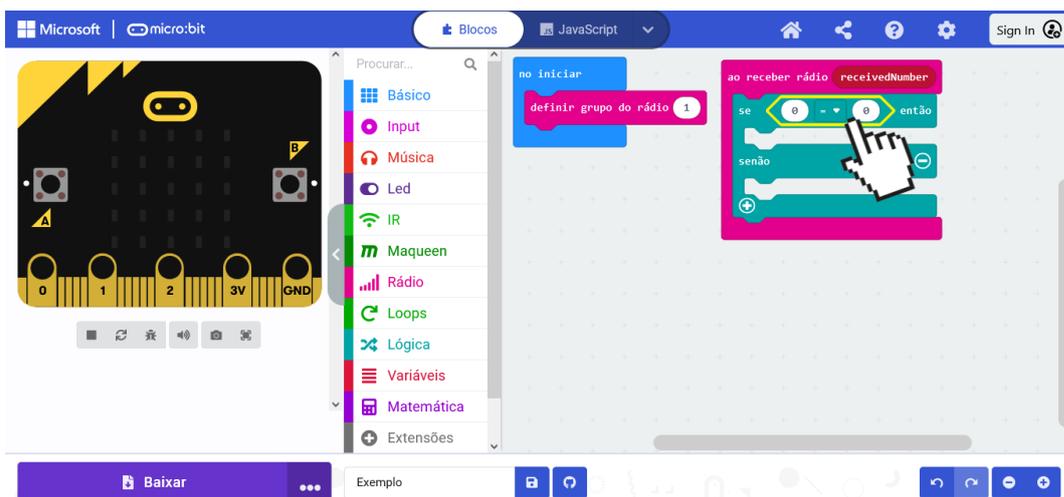


Figura 22.10: Montando um comando.

# Carrinho de controle remoto

- 8. Arraste o comando **receivNumber** para o primeiro **0** da comparação **se 0 = 0** **então**, Figuras 22.11 e 22.12;

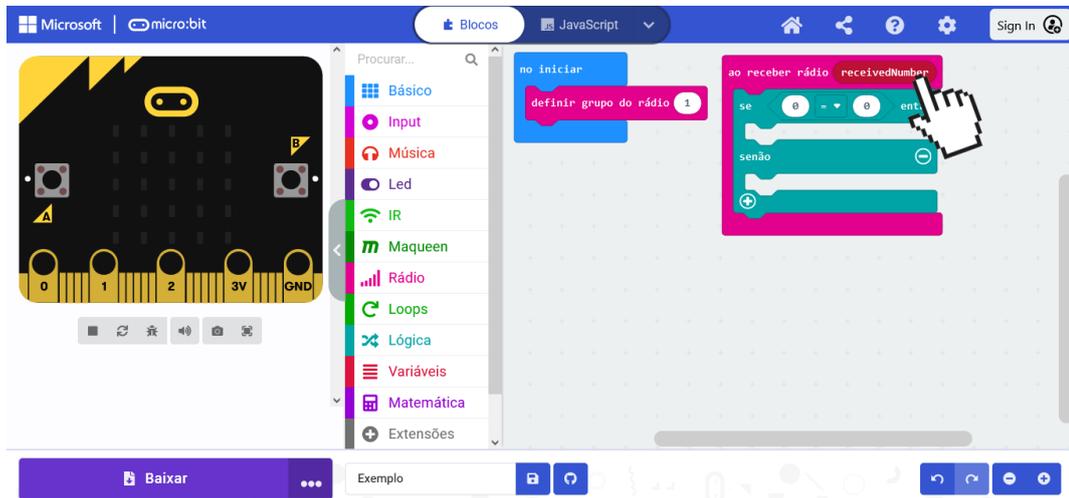


Figura 22.11: Montando um comando.

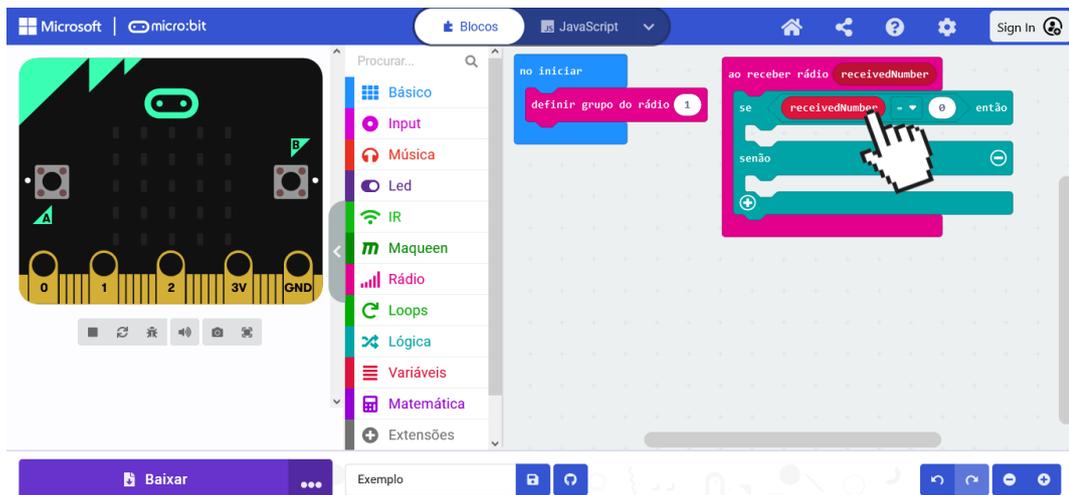


Figura 22.12: Montando um comando.



# Carrinho de controle remoto

9. Em **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0**, Figura 22.13;

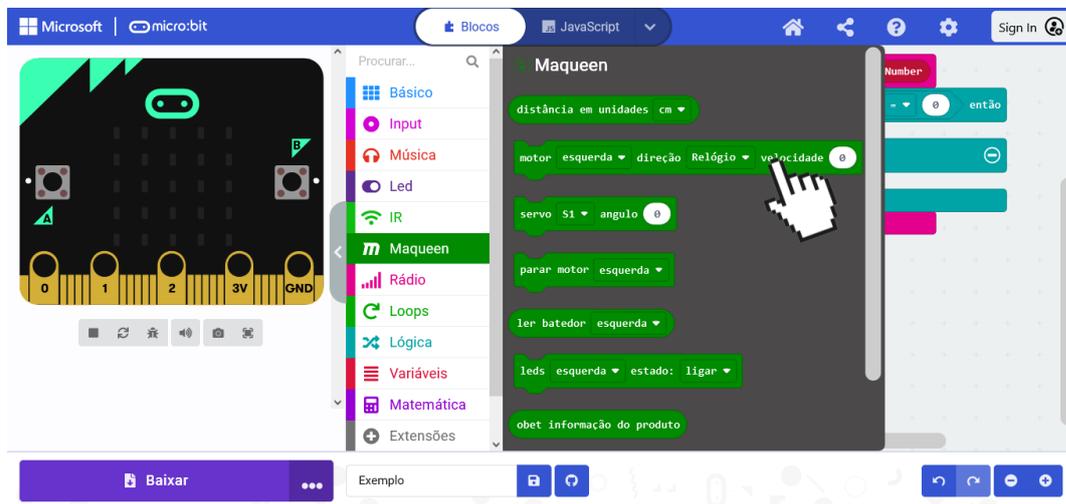


Figura 22.13: Selecionando um comando.

10. Coloque-o dentro do bloco **se receivedNumber = 0 então**, Figura 22.14;

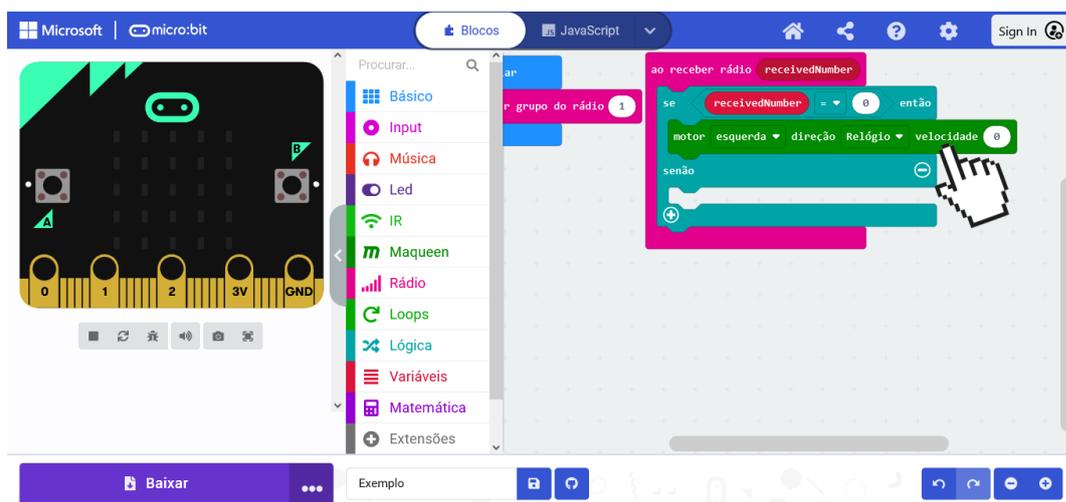


Figura 22.14: Montando um comando.

# Carrinho de controle remoto

11. Em **motor esquerda direção Relógio velocidade 0**, altere motor **esquerda** para **todos** e a velocidade **0** para **255**, Figuras 22.15 e 22.16;

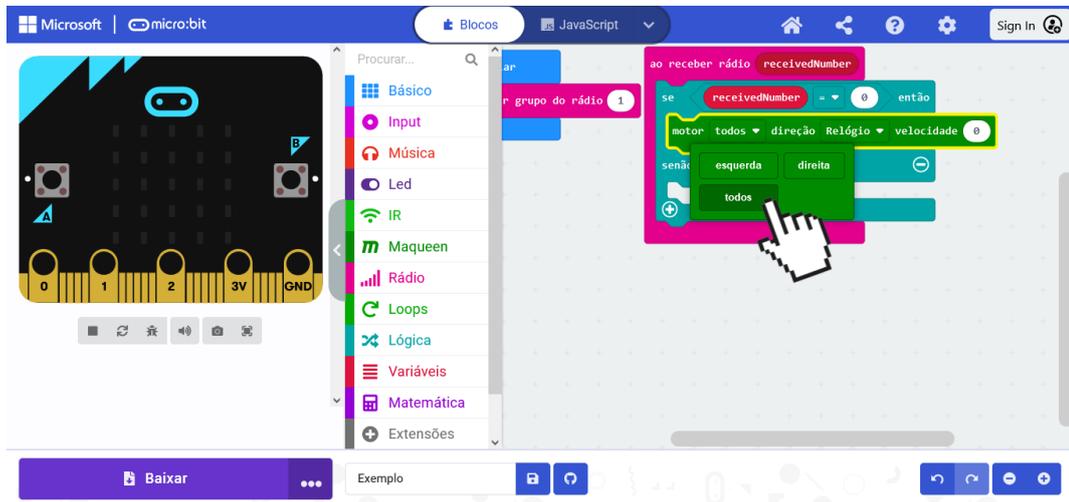


Figura 22.15: Alterando motores.



Figura 22.16: Alterando uma velocidade.



# Carrinho de controle remoto

12. Clique no sinal de +, abaixo de **senão**, para abrir um novo espaço de montagem chamado **senão se < > então**, Figura 22.17;

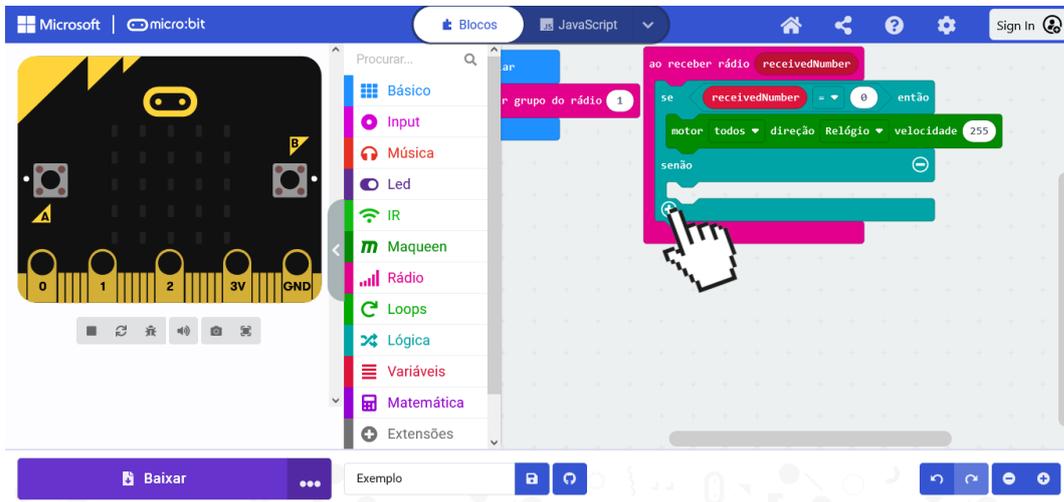


Figura 22.17: Adicionando uma condição.

13. Repita o último passo três vezes, até que fique com três **senão se < > então** e um **senão** no bloco de lógica, Figuras 22.18, 22.19 e 22.20;

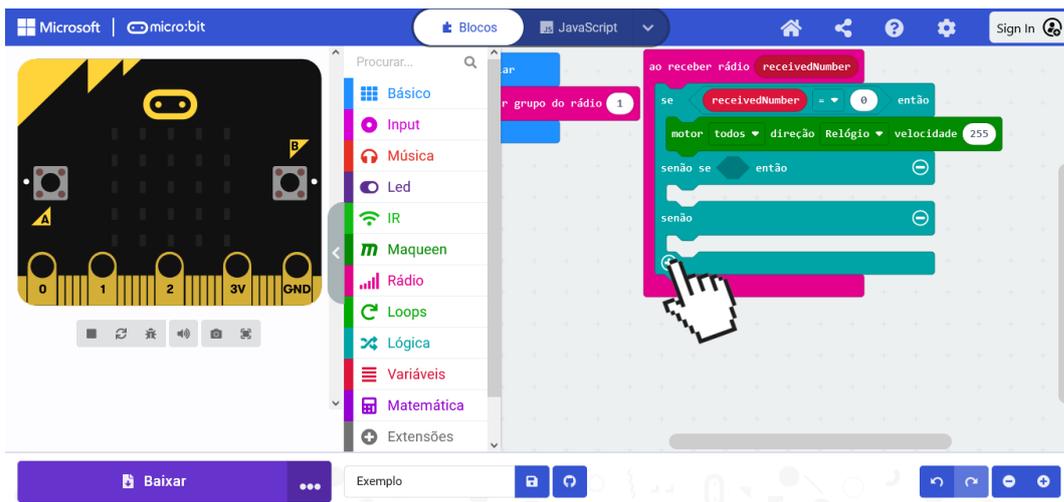


Figura 22.18: Adicionando uma condição.

# Carrinho de controle remoto

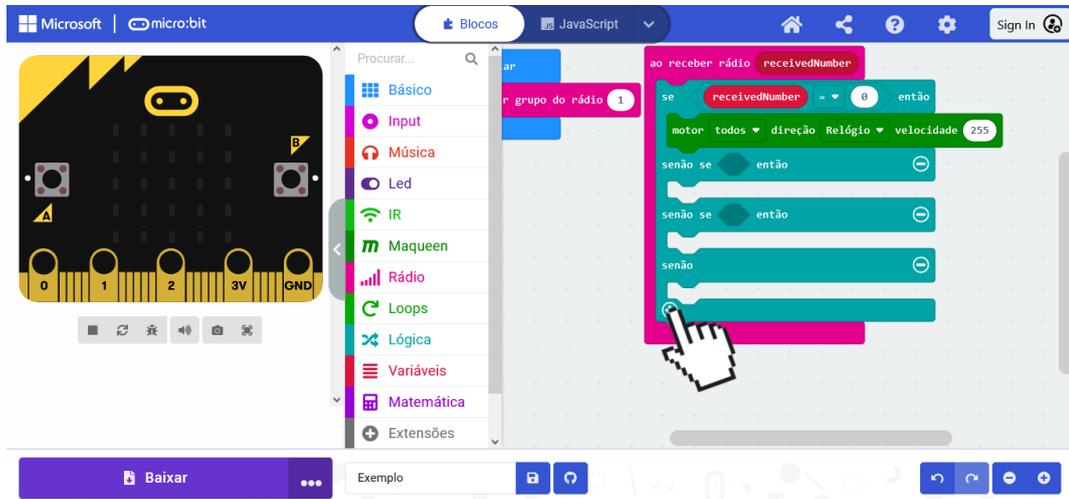


Figura 22.19: Adicionando uma condição.



Figura 22.20: Condições adicionadas.



# Carrinho de controle remoto

14. Selecione a comparação `receivedNumber = 0` e clique sobre a mesma com o botão de abrir janelas (normalmente o botão direito do mouse) para duplicá-la, Figura 22.21;

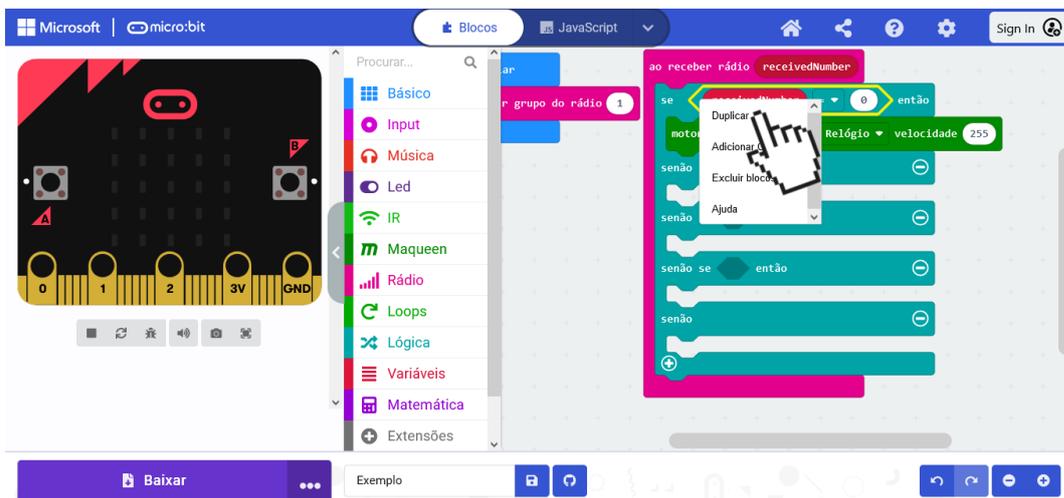


Figura 22.21: Duplicando uma comparação.

15. Arraste a comparação duplicada até o espaço `<>` de `senão se <> então`, como na Figura 22.22;

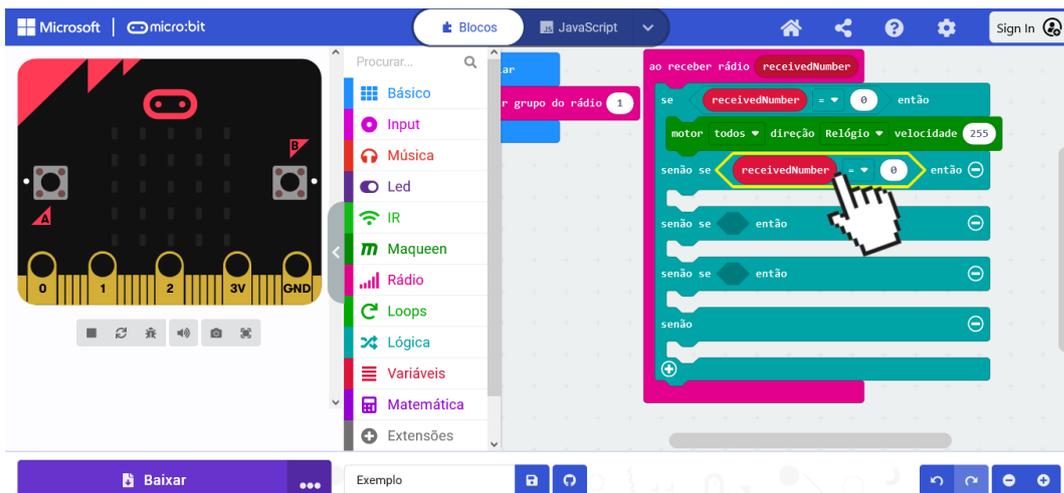


Figura 22.22: Montando uma comparação.

# Carrinho de controle remoto

16. Na comparação **senão se receivedNumber = 0**, altere o número **0** para **1**, Figura 22.23;

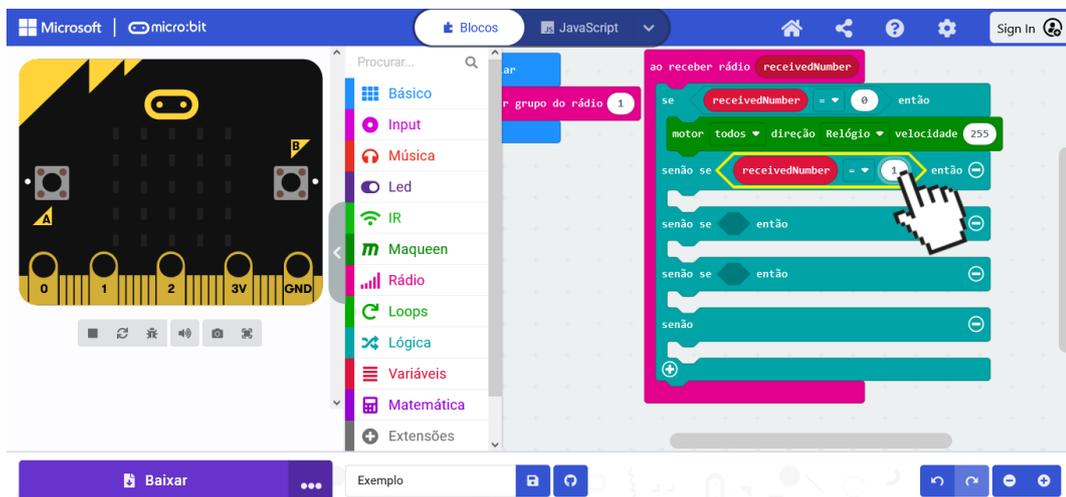


Figura 22.23: Alterando uma comparação.

17. Em **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0** e coloque-o dentro do bloco **senão se receivedNumber = 1 então**, Figuras 22.24 e 22.25;

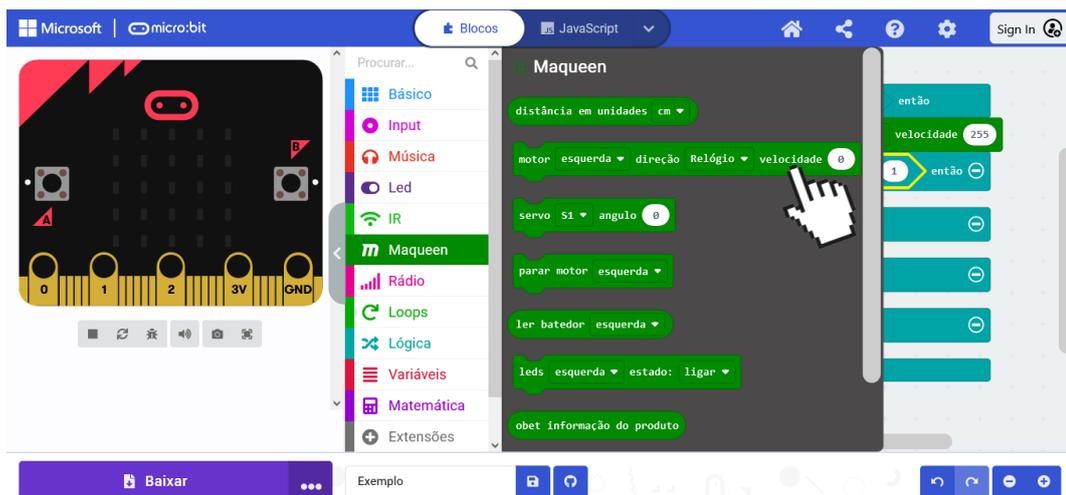
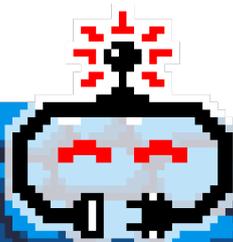


Figura 22.24: Selecionando um comando.



# Carrinho de controle remoto

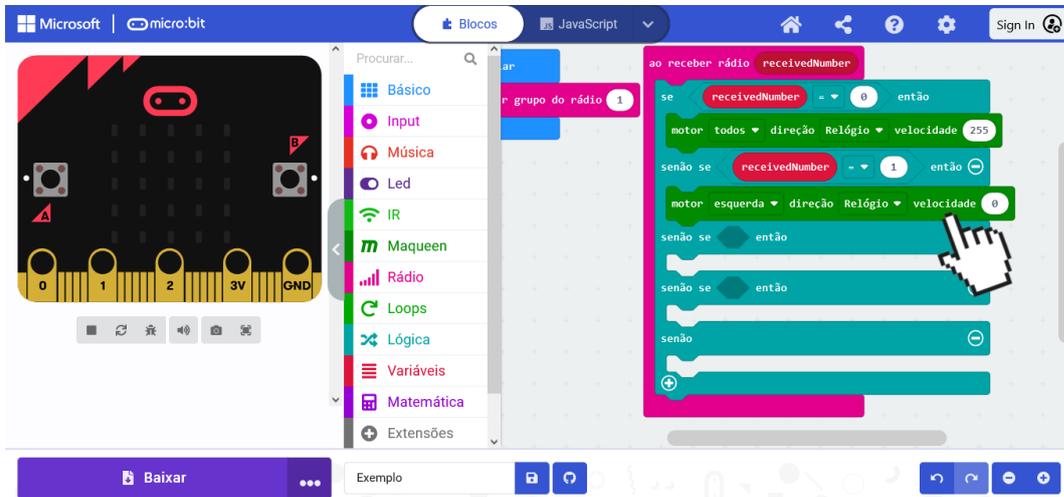


Figura 22.25: Montando um comando.

- 18.** Em **motor esquerda direção Relógio velocidade 0**, altere motor **esquerda** para **todos**, direção **Relógio** para **Contra-relógio** e velocidade **0** para **255**, como nas Figuras 22.26, 22.27 e 22.28;



Figura 22.26: Alterando motores.

# Carrinho de controle remoto

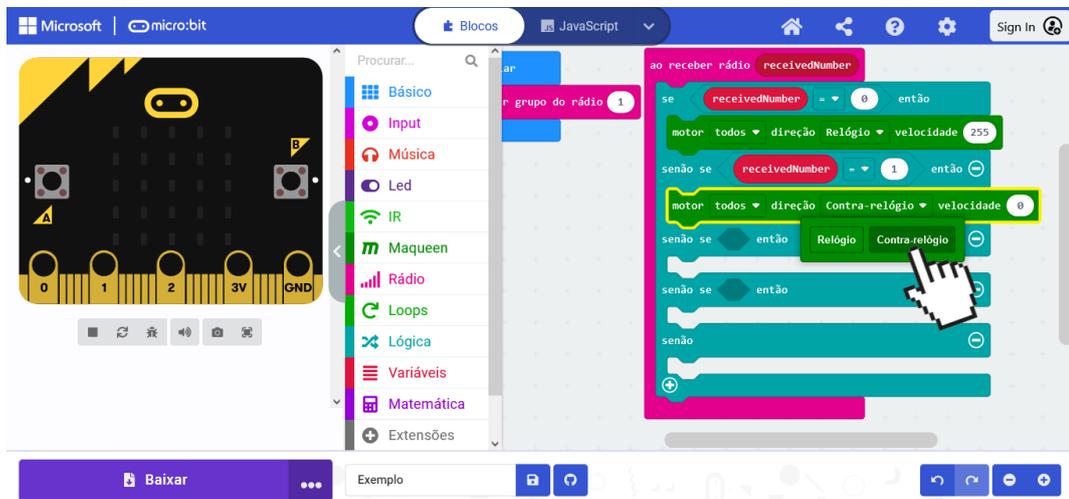


Figura 22.27: Alterando uma direção.

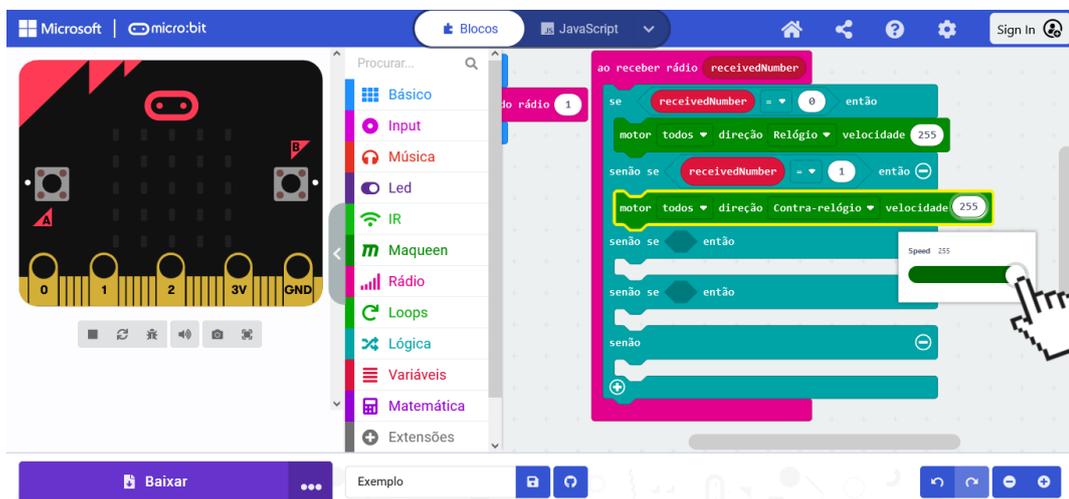


Figura 22.28: Alterando uma velocidade.



# Carrinho de controle remoto

19. Duplique novamente a comparação **receivedNumber = 0**, Figura 22.29;

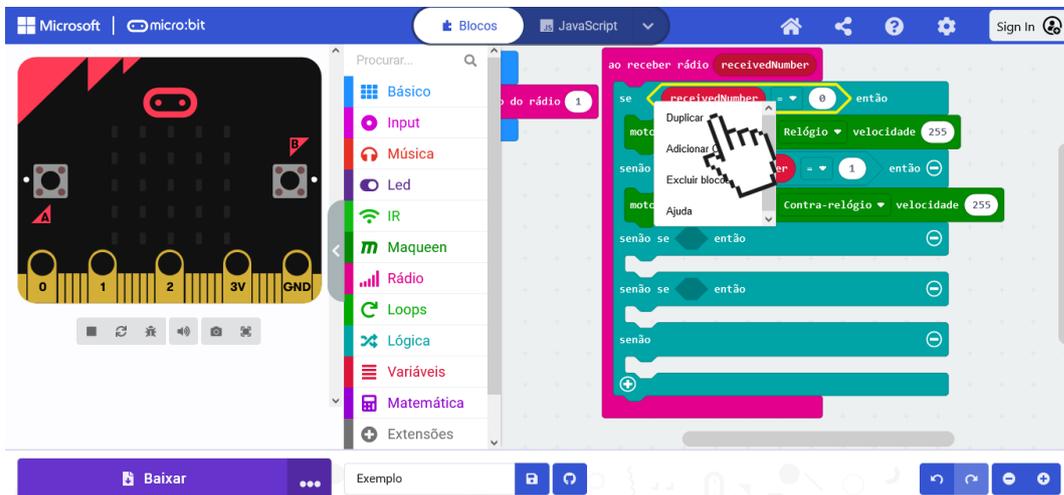


Figura 22.29: Duplicando uma comparação.

20. Insira a comparação duplicada dentro do espaço **<>** da próxima condição **senão se <> então**, Figura 22.30;

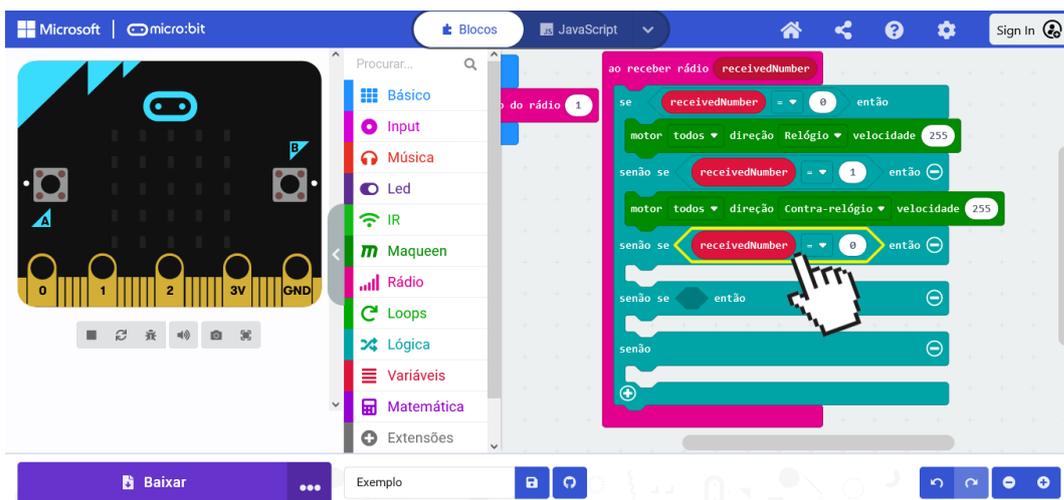


Figura 22.30: Montando uma comparação.

# Carrinho de controle remoto

- 21.** Nesta última condição **senão se receivedNumber = 0 então**, altere o número **0** para **2**, Figura 22.31;

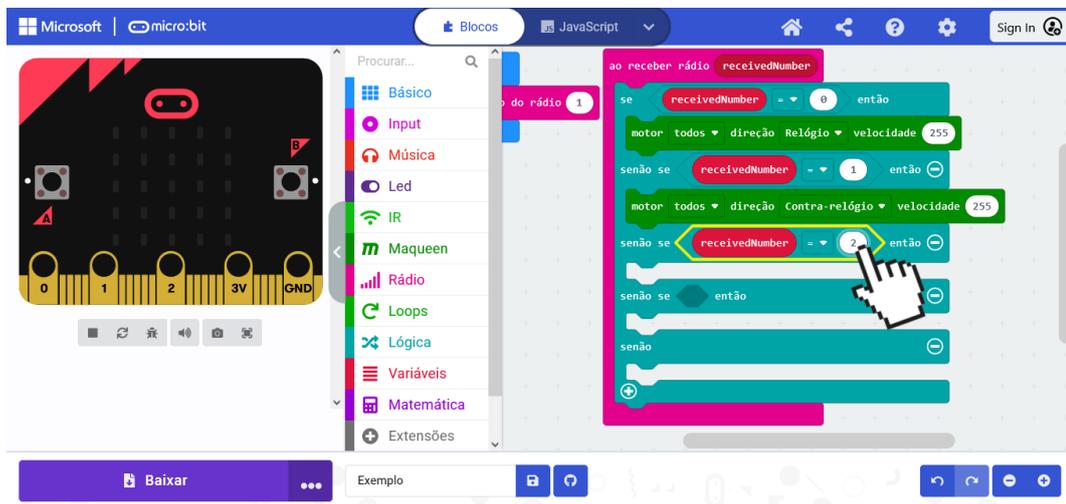


Figura 22.31: Alterando uma comparação.

- 22.** Em **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0**, Figura 22.32;

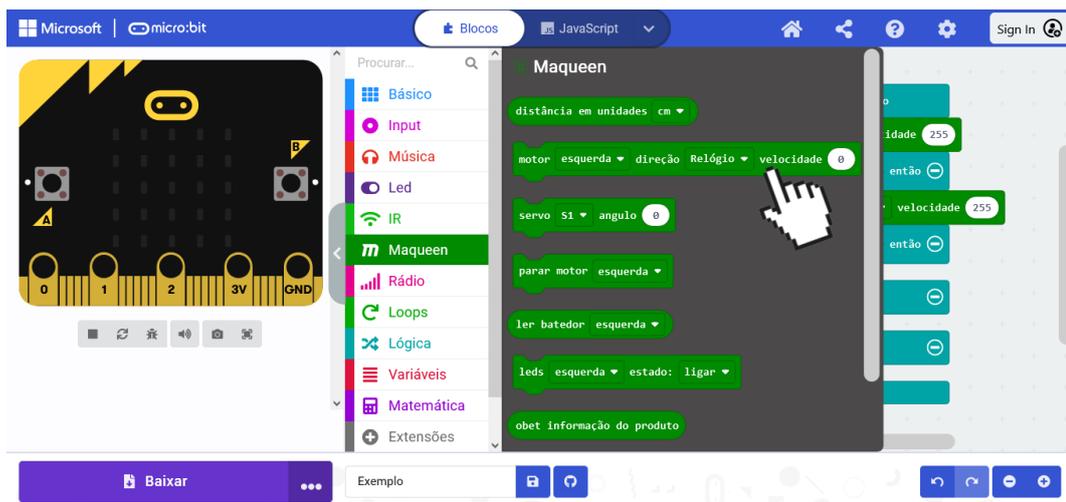
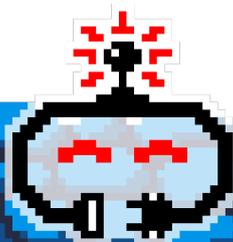


Figura 22.32: Selecionando um comando.



# Carrinho de controle remoto

- 23.** Insira o último comando selecionado dentro do bloco **senão se receivedNumber = 2 então**, Figura 22.33;

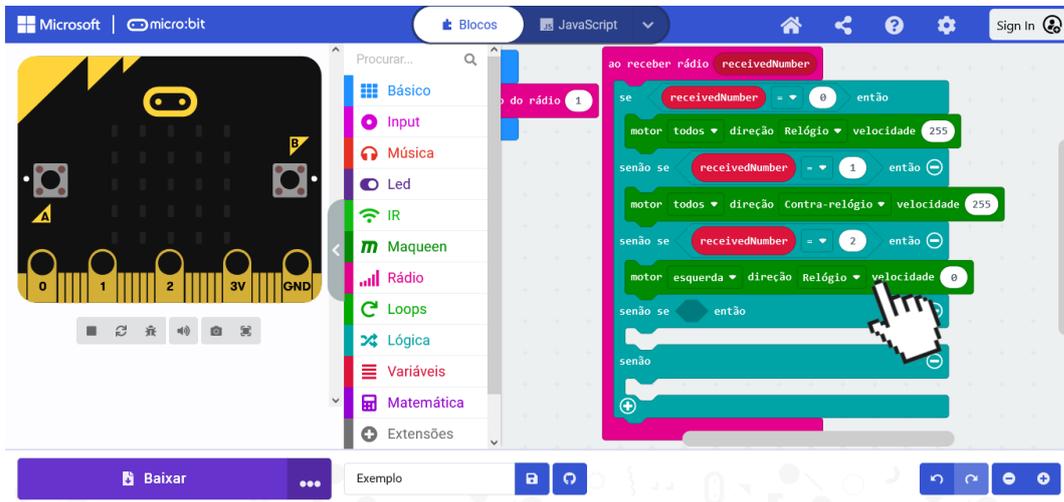


Figura 22.33: Montando um comando.

- 24.** Em **motor esquerda direção Relógio velocidade 0**, altere a velocidade **0** para **100**, Figura 22.34;

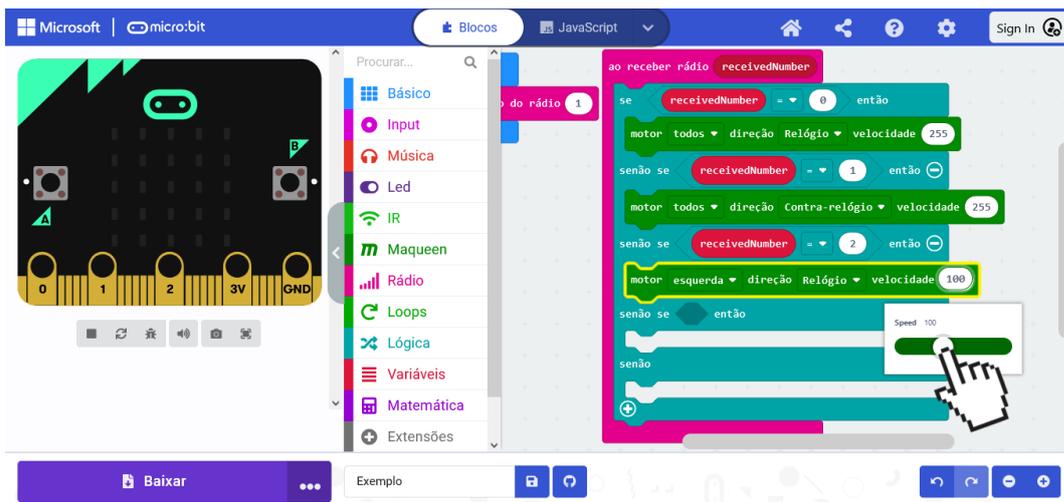


Figura 22.34: Alterando uma velocidade.

# Carrinho de controle remoto

25. Em **Maqueen**, selecione novamente o comando **motor esquerda direção Relógio velocidade 0**, Figura 22.35;

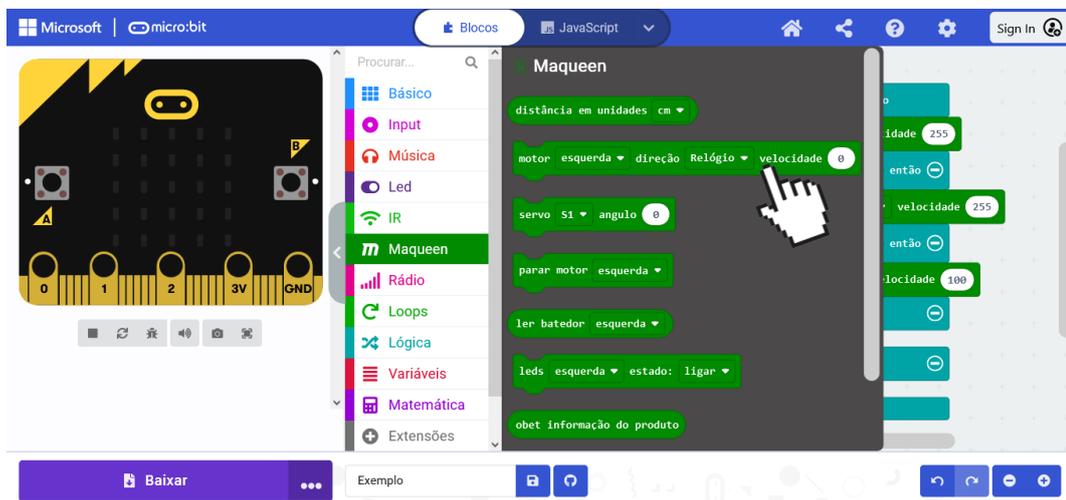


Figura 22.35: Selecionando um comando.

26. Insira o comando selecionado logo abaixo de **motor esquerda direção Relógio velocidade 100**, Figura 22.36;

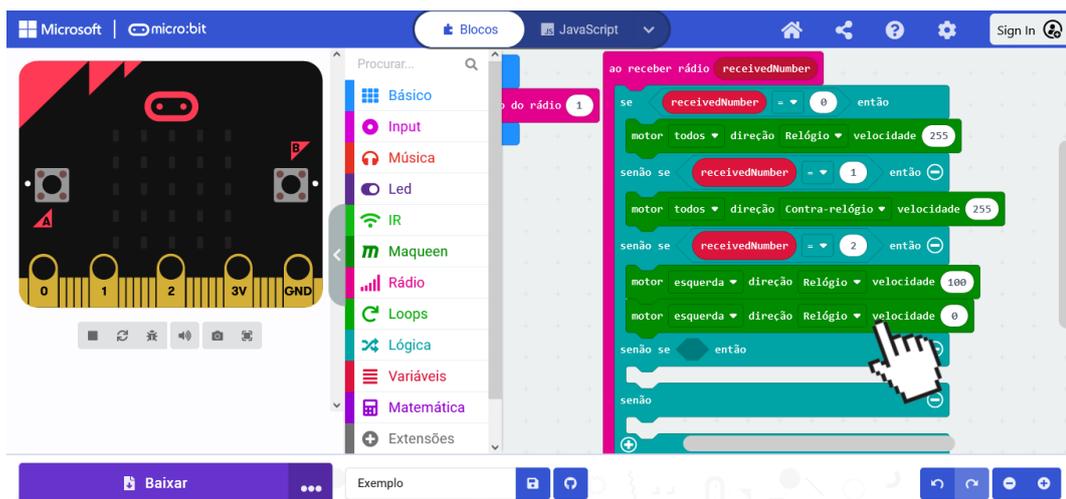
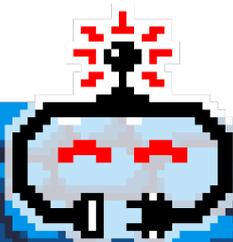


Figura 22.36: Montando um comando.



# Carrinho de controle remoto

- 27.** Em **motor esquerda direção Relógio velocidade 0**, altere motor **esquerda** para **direita**, Figura 22.37;

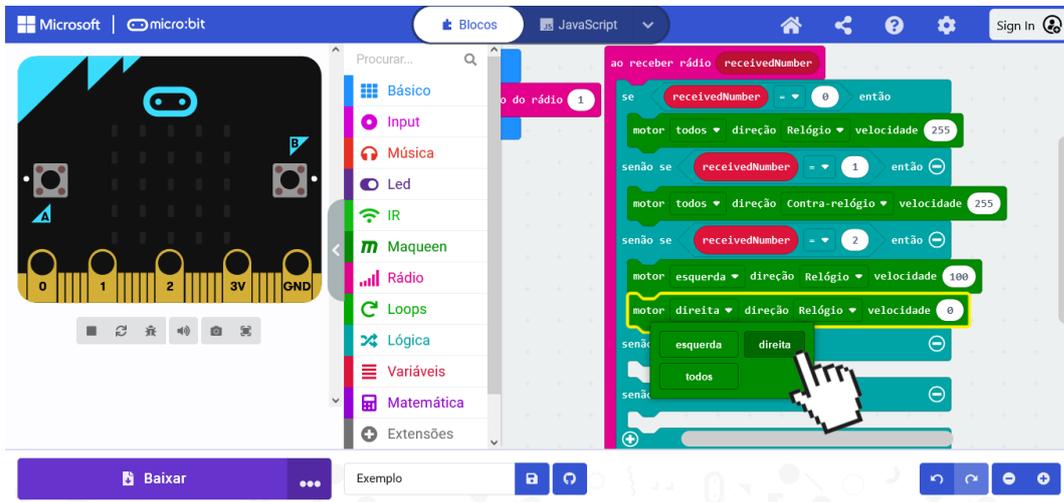


Figura 22.37: Alterando um motor.

- 28.** Duplique novamente a comparação **receivedNumber= 0** e a insira no último espaço de **senão se <> então**, Figuras 22.38 e 22.39;

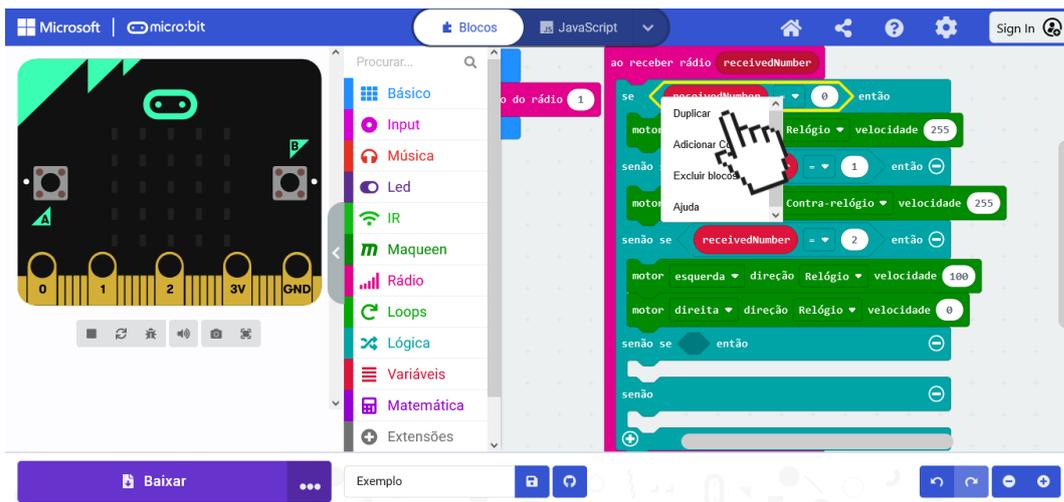


Figura 22.38: Duplicando uma comparação.

# Carrinho de controle remoto

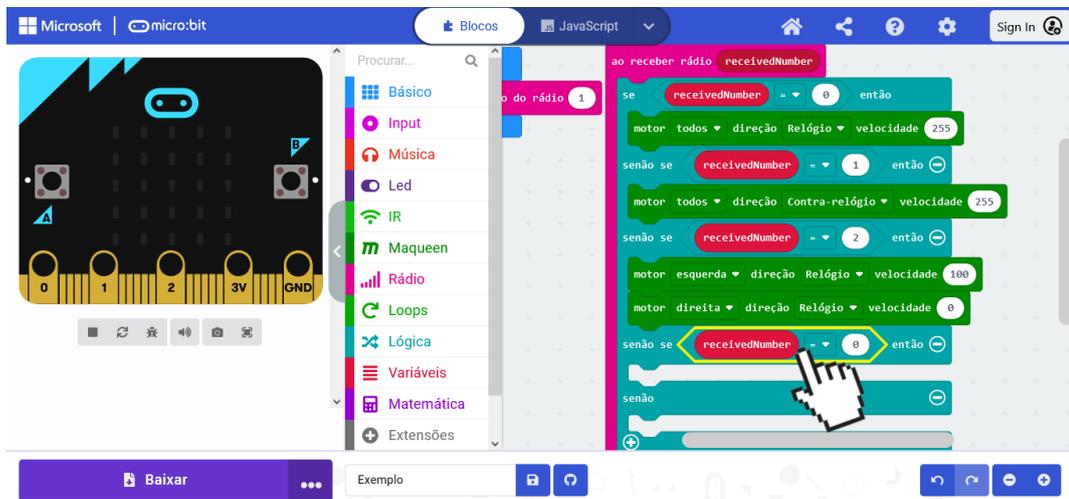


Figura 22.39: Montando um comando.

**29.** Nesta última comparação, altere o número 0 para 3, Figura 22.40;

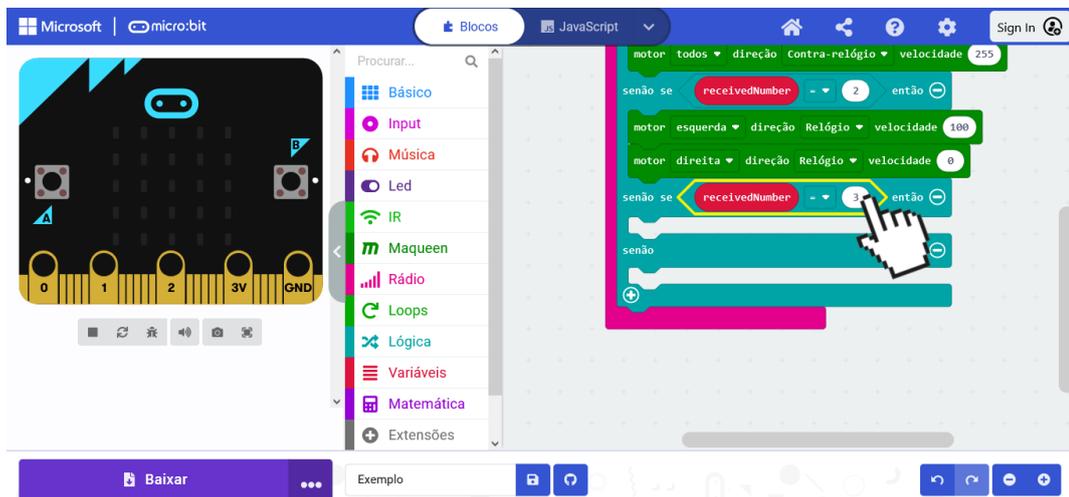


Figura 22.40: Alterando uma comparação.



# Carrinho de controle remoto

- 30.** Em **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0**, Figuras 22.41;

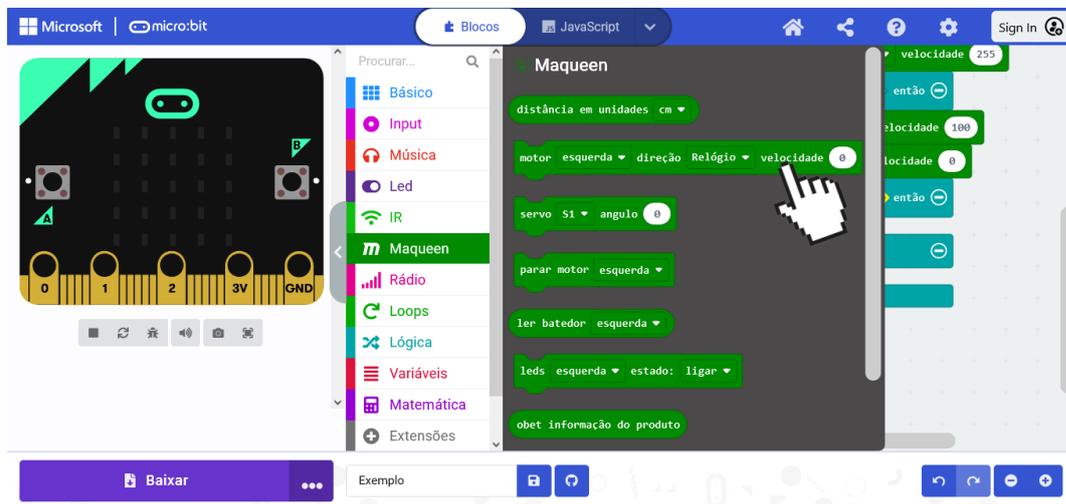


Figura 22.41: Selecionando um comando.

- 31.** Coloque-o dentro do bloco **senão se receivedNumber = 3 então**, Figura 22.42;

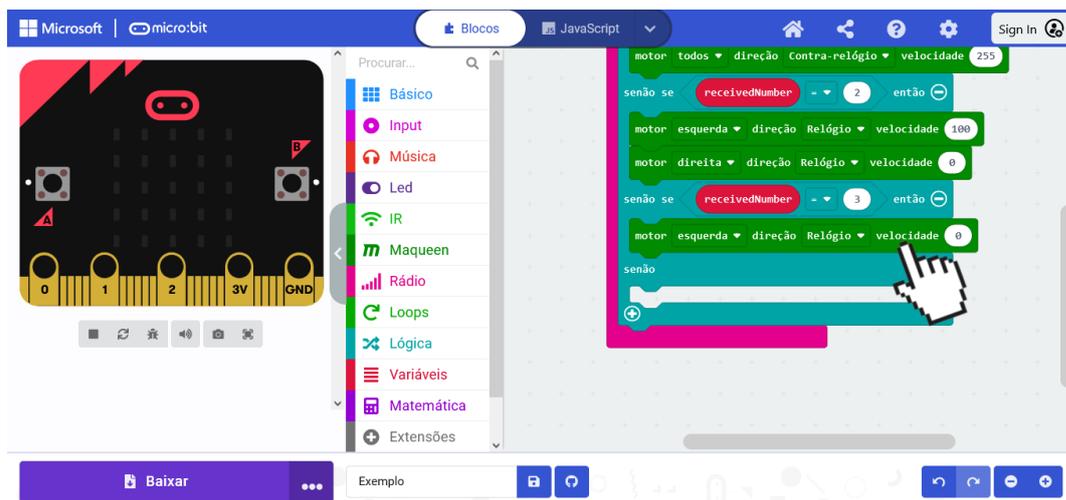


Figura 22.42: Montando um comando.

# Carrinho de controle remoto

**32.** No último comando selecionado, altere motor **esquerda** para **direita** e a velocidade **0** para **100**, Figuras 22.43 e 22.44;

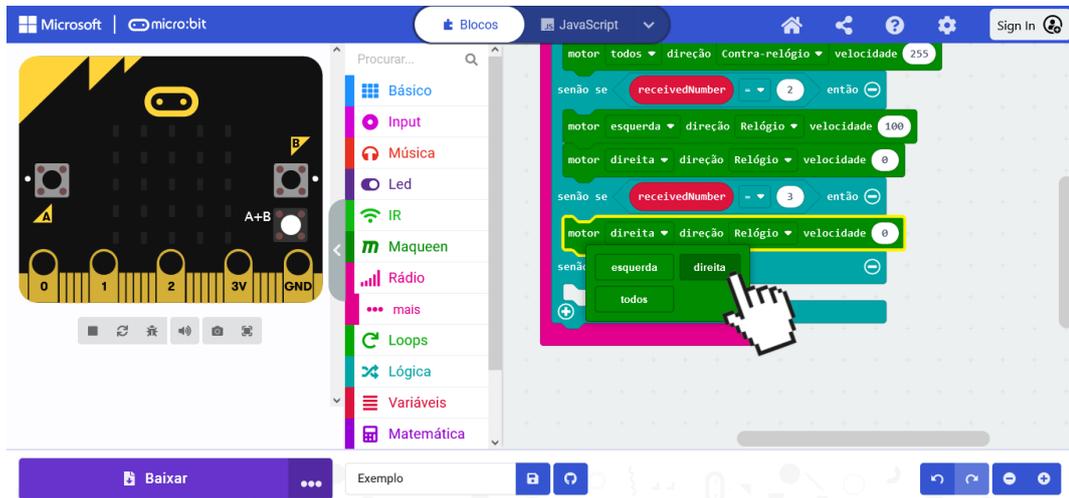


Figura 22.43: Alterando um motor.

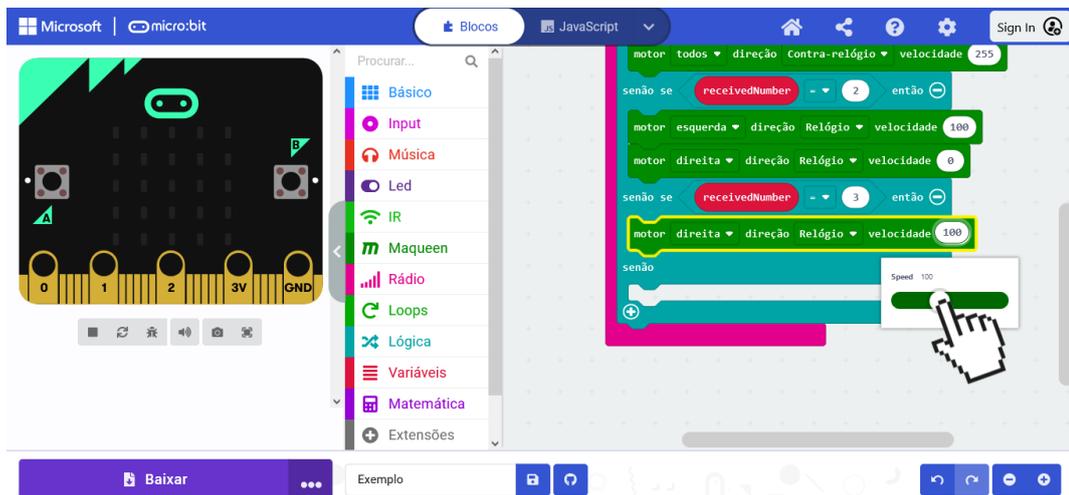
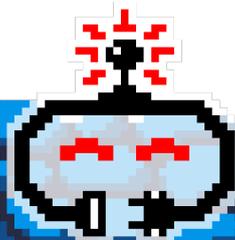


Figura 22.44: Alterando uma velocidade.



# Carrinho de controle remoto

- 33.** Novamente em **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0**, Figura 22.45;

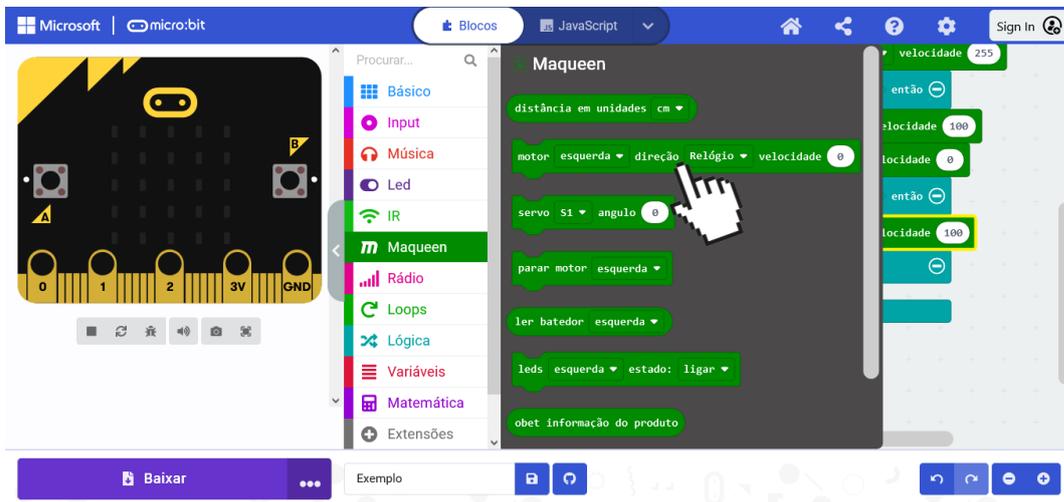


Figura 22.45: Selecionando um comando.

- 34.** Coloque o comando selecionado logo abaixo de **motor direita direção Relógio velocidade 100**, Figura 22.46;

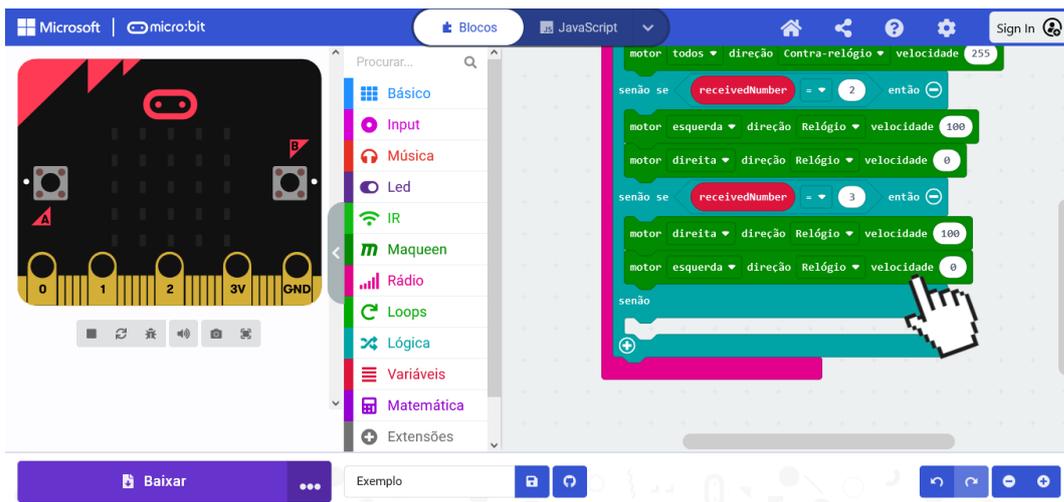


Figura 22.46: Montando um comando.

# Carrinho de controle remoto

- 35.** Em **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0**, Figura 22.47;



Figura 22.47: Selecionando um comando.

- 36.** Adicione o comando selecionado no último bloco de **senão**, Figura 22.48;

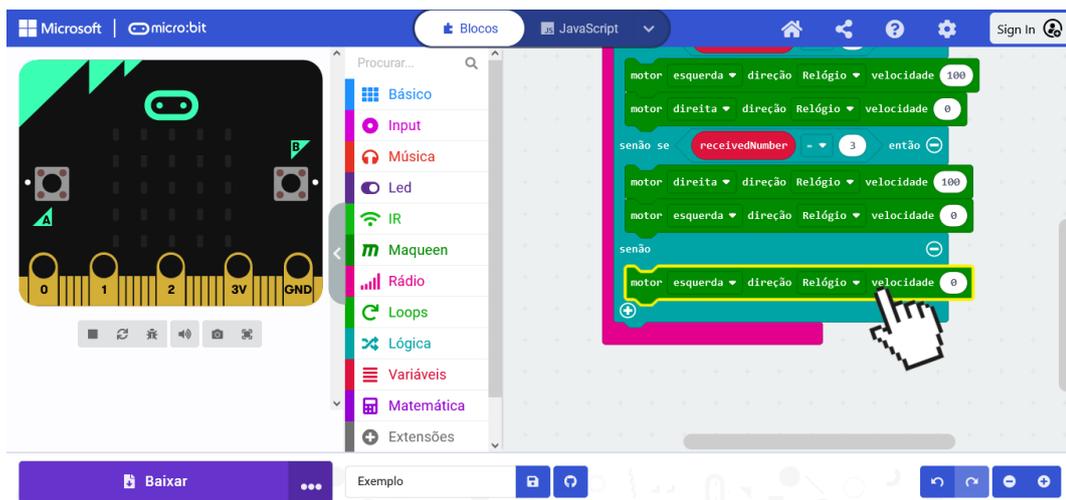


Figura 22.48: Montando um comando.



# Carrinho de controle remoto

**37.** Altere motor **esquerda** para **todos** e direção **Relógio** para **Contra-relógio**, Figuras 22.49 e 22.50;

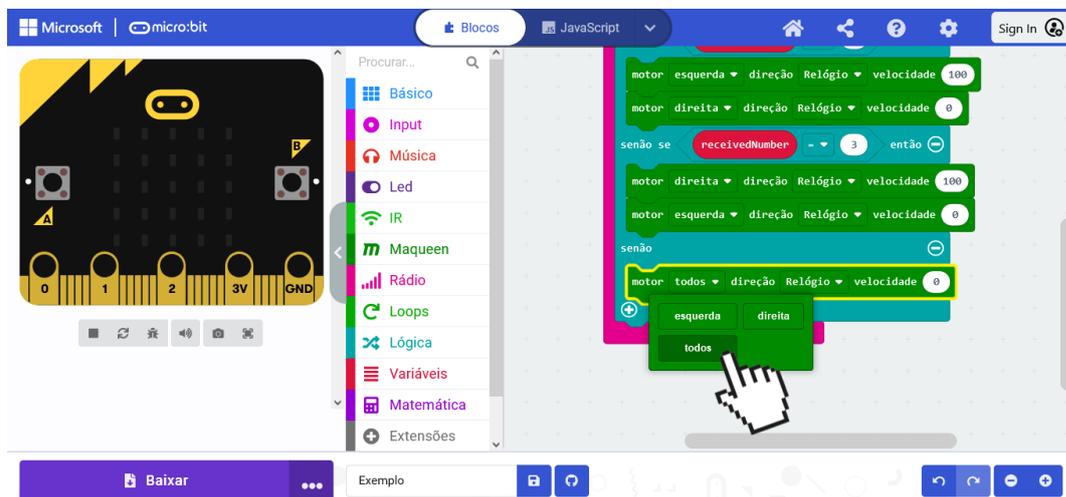


Figura 22.49: Alterando um motor.

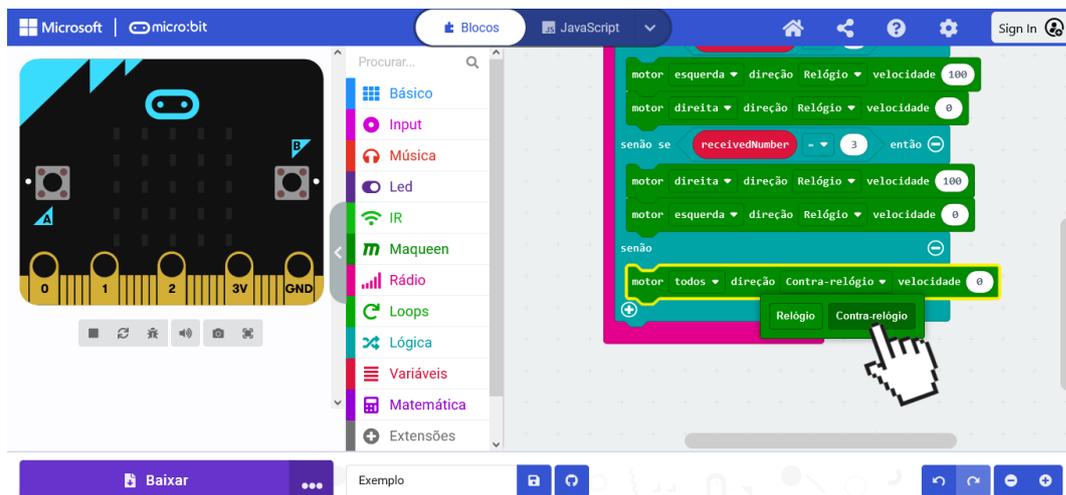


Figura 22.50: Alterando uma direção.

# Carrinho de controle remoto

38. Em **Maqueen**, selecione o comando **motor esquerda direção Relógio velocidade 0**, Figura 22.51;

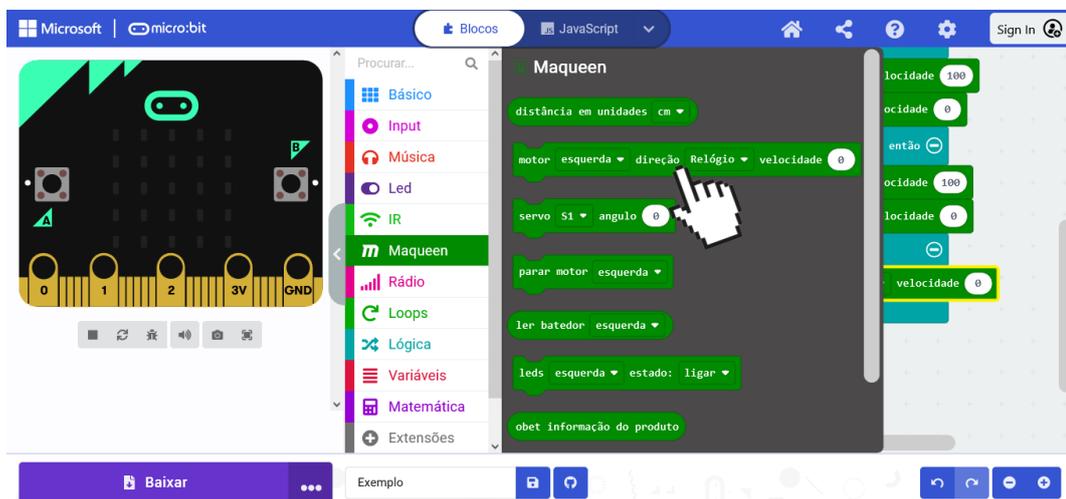


Figura 22.51: Selecionando um comando.

39. Insira o comando selecionado abaixo de **motor todos direção Contra-relógio velocidade 0**, Figura 22.52;

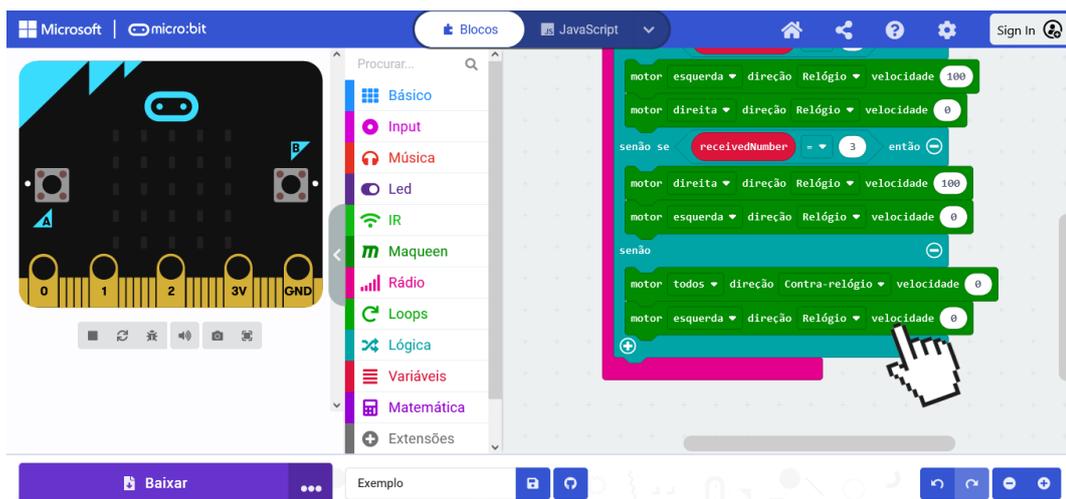


Figura 22.52: Montando um comando.



# Carrinho de controle remoto

40. Em **motor esquerda direção Relógio velocidade 0**, altere motor **esquerda** para **todos**, Figura 22.53;

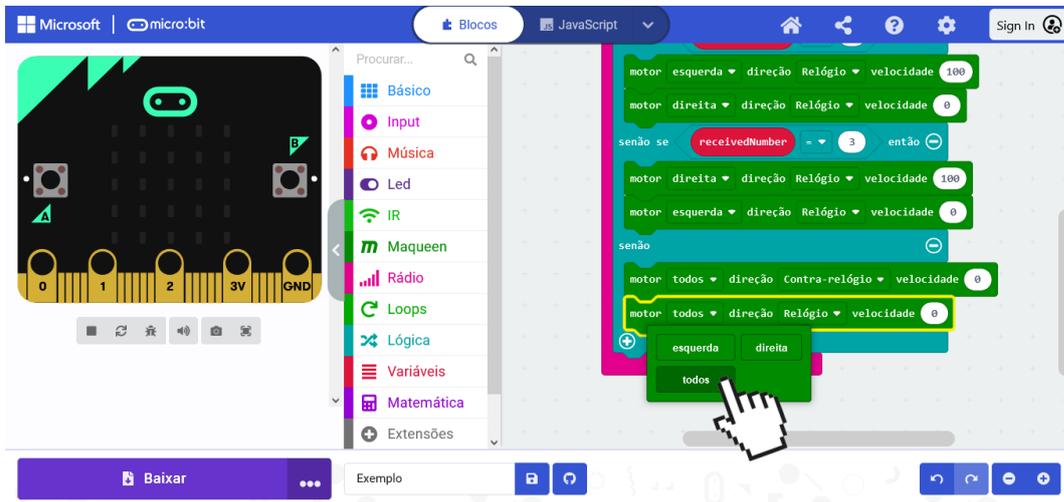


Figura 22.53: Alterando um motor.

41. Na aba **Input**, selecione o bloco de acionamento **no botão A pressionado**, Figuras 22.54 e 22.55;



Figura 22.54: Selecionando um bloco de acionamento.

# Carrinho de controle remoto

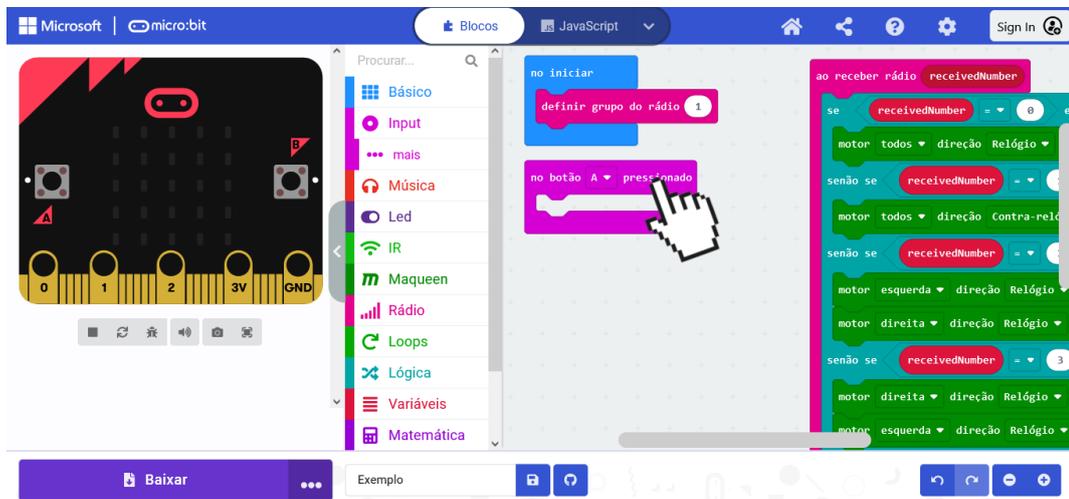


Figura 22.55: Montando um bloco de acionamento.

**42.** Em **Rádio**, selecione o comando **rádio envia número 0**, Figura 22.56;



Figura 22.56: Selecionando um comando.



# Carrinho de controle remoto

43. Insira o último comando selecionado dentro do bloco de acionamento **no botão A pressionado**, Figura 22.57;



Figura 22.57: Montando um comando.

44. Em **Input**, selecione o bloco de acionamento **no botão A pressionado** Figuras 22.58 e 22.59;



Figura 22.58: Selecionando um bloco de acionamento.

# Carrinho de controle remoto

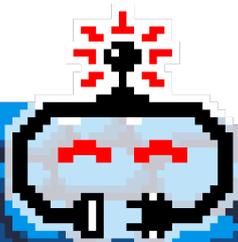


Figura 22.59: Montando um bloco de acionamento.

**45.** No último bloco selecionado, altere o botão **A** para **B**, Figura 22.60;



Figura 22.60: Alterando botões.



# Carrinho de controle remoto

46. Em **Rádio**, selecione o comando **rádio envia número 0**, Figura 22.61;

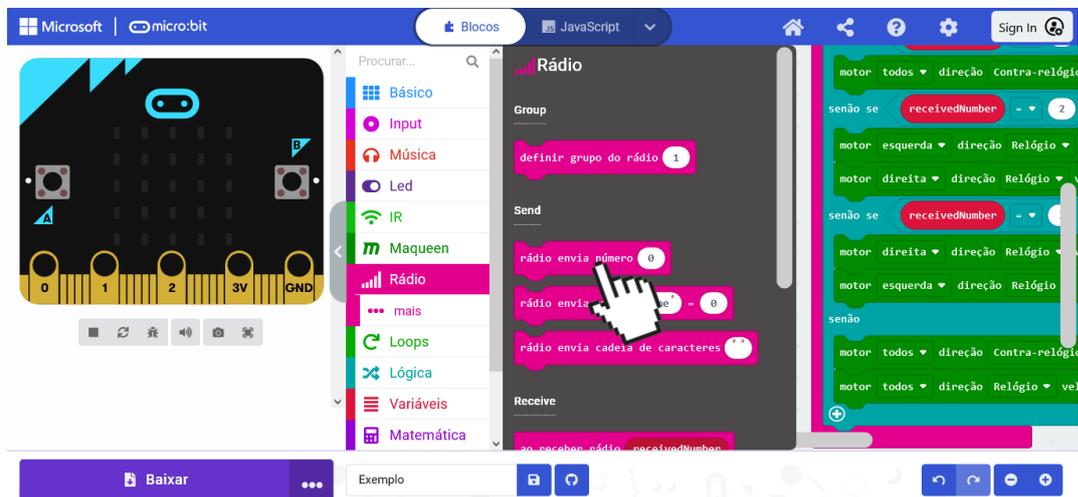


Figura 22.61: Selecionando um comando.

47. Insira o último comando selecionado, dentro do bloco de acionamento **no botão B pressionado**, Figura 22.62;

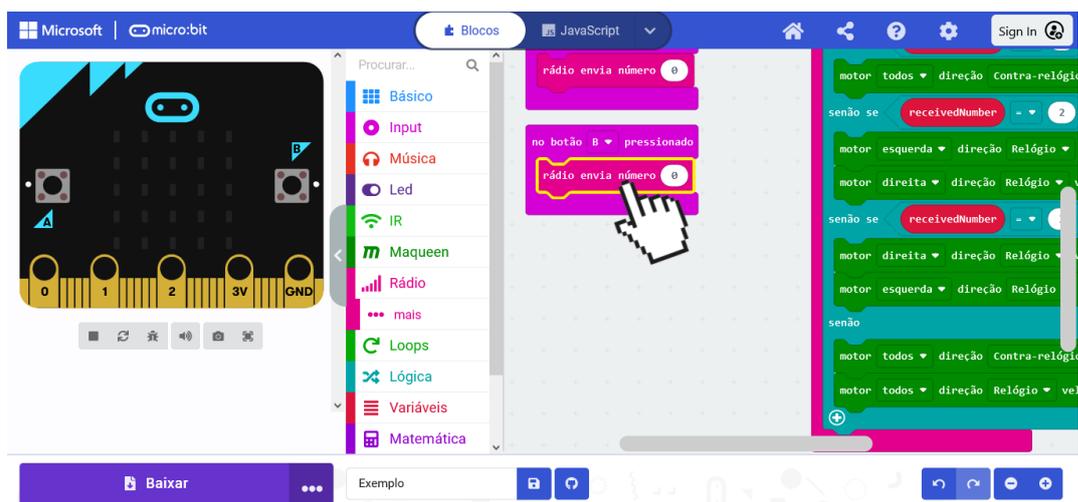


Figura 22.62: Montando um comando.

# Carrinho de controle remoto

48. Em **rádio envia número 0**, que está dentro de **no botão B pressionado**, altere o número **0** para **1**, Figura 22.63;

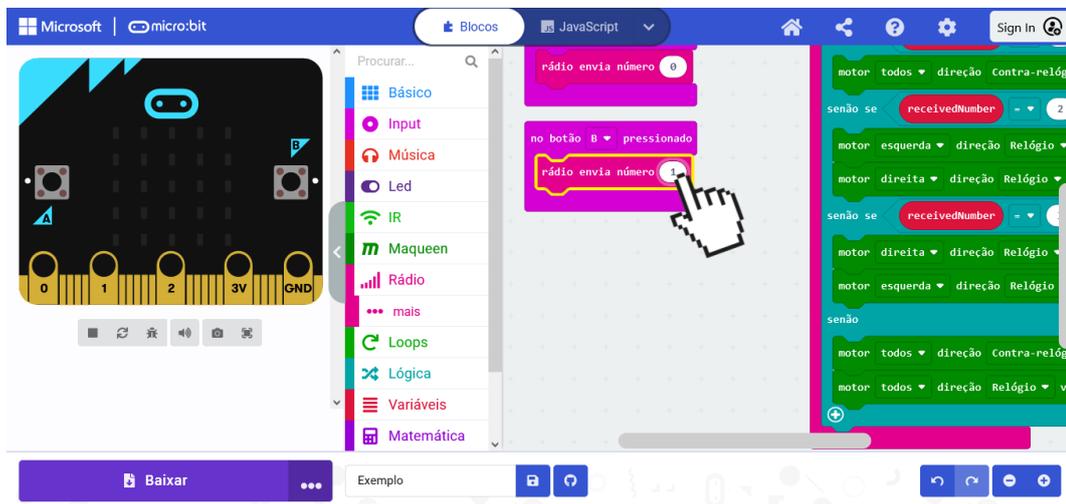


Figura 22.63: Alterando o número de envio de um rádio.

49. Em **Input**, selecione o bloco de acionamento **em agitar**, Figuras 22.64 e 22.65;



Figura 22.64: Selecionando um bloco de acionamento.



# Carrinho de controle remoto



Figura 22.65: Montando um bloco de acionamento.

- 50.** No último bloco selecionado, altere **agitar** para **inclinar para a esquerda**, Figura 22.66;

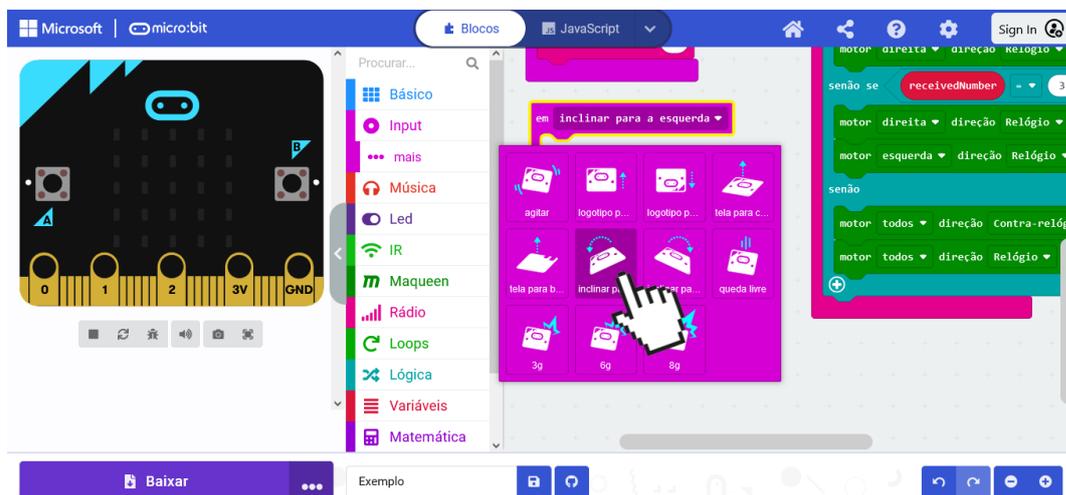


Figura 22.66: Alterando um acionamento.

# Carrinho de controle remoto

**51.** Em **Rádio**, selecione o comando **radio envia número 0**, Figura 22.67;



Figura 22.67: Selecionando um comando.

**52.** Insira o último comando selecionado dentro de **em inclinar para a esquerda**, Figura 22.68;

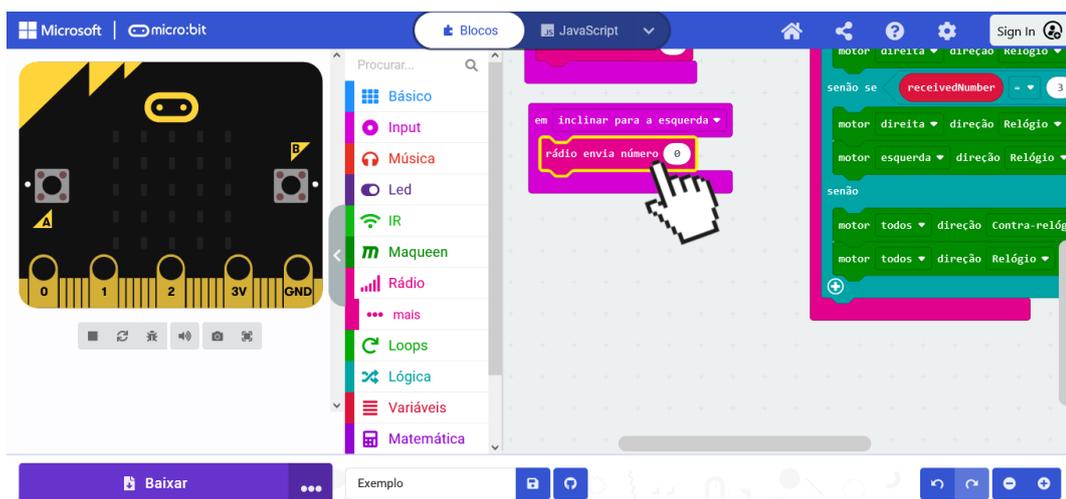
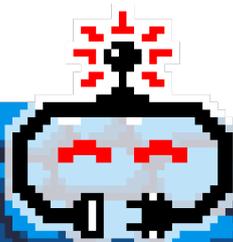


Figura 22.68: Montando um comando.



# Carrinho de controle remoto

- 53.** Altere o número de envio de **0** para **3** no bloco **em inclinar para a esquerda**, Figura 22.69;

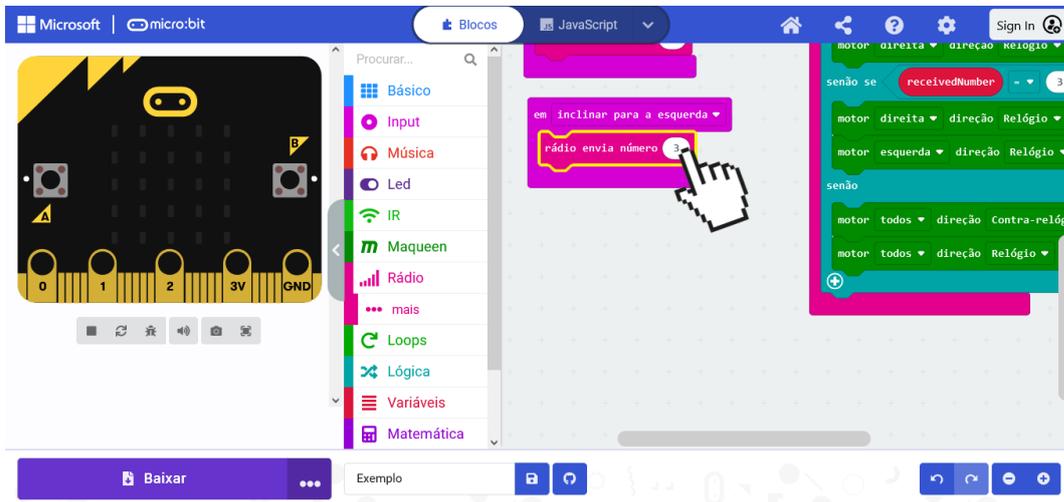


Figura 22.69: Alterando o número de envio de um rádio.

- 54.** Em **Input**, selecione o bloco de acionamento **em agitar**, Figuras 22.70 e 22.71;



Figura 22.70: Selecionando um bloco de acionamento.

# Carrinho de controle remoto

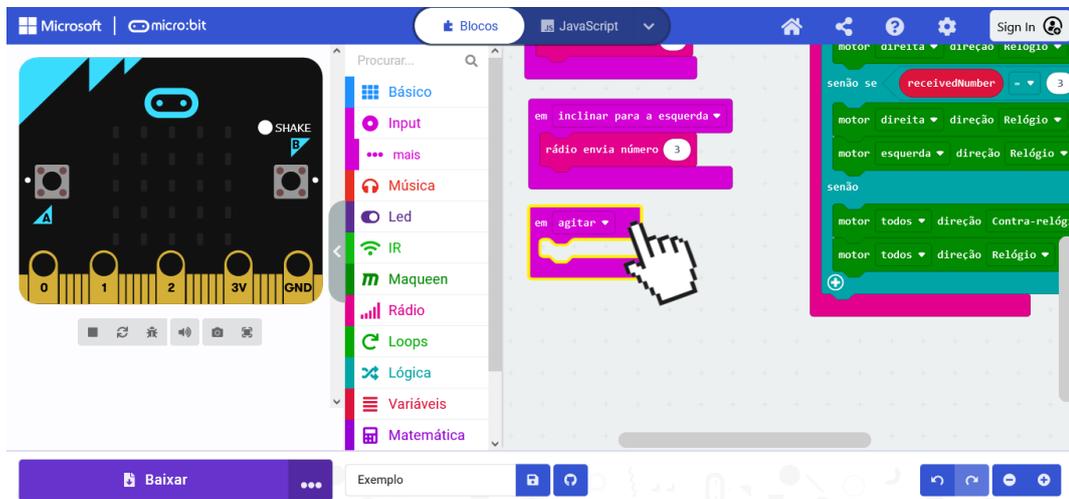


Figura 22.71: Montando um bloco de acionamento.

- 55.** No último bloco selecionado, altere **agitar** para **inclinar para a direita**, Figura 22.72;

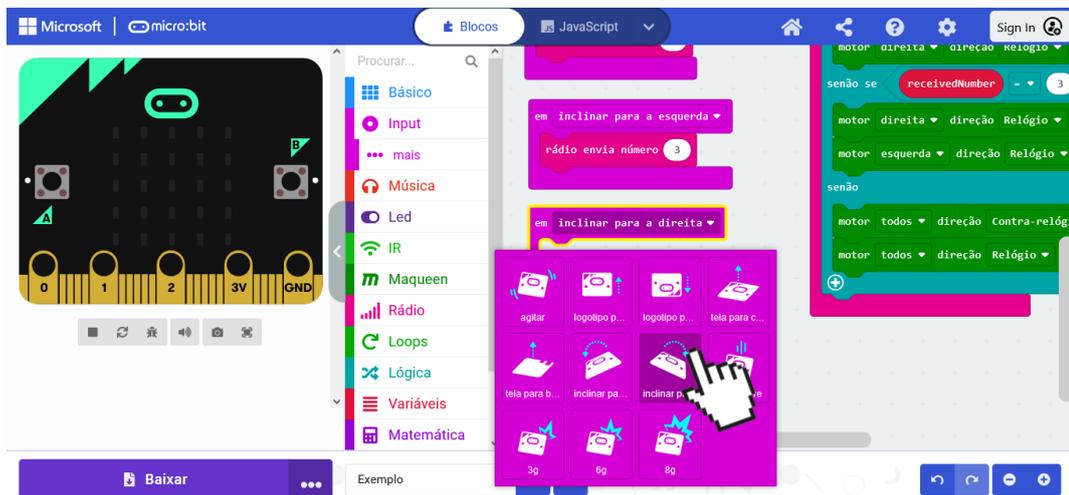


Figura 22.72: Alterando um acionamento.



# Carrinho de controle remoto

**56.** Em **Rádio**, selecione o comando **rádio envia número 0**, Figura 22.73;

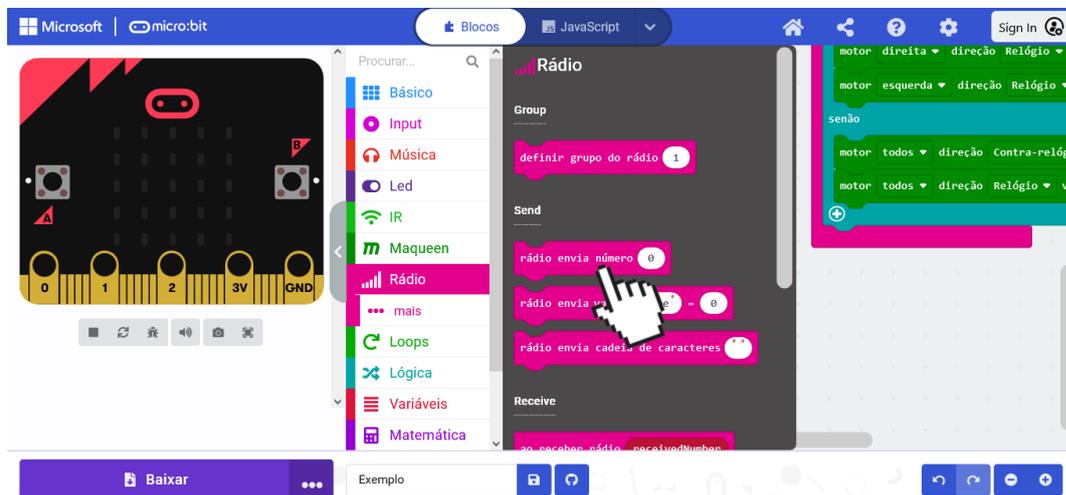


Figura 22.73: Selecionando um comando.

**57.** Adicione o último comando selecionado dentro do bloco de acionamento **em inclinar para a direita**, Figura 22.74;

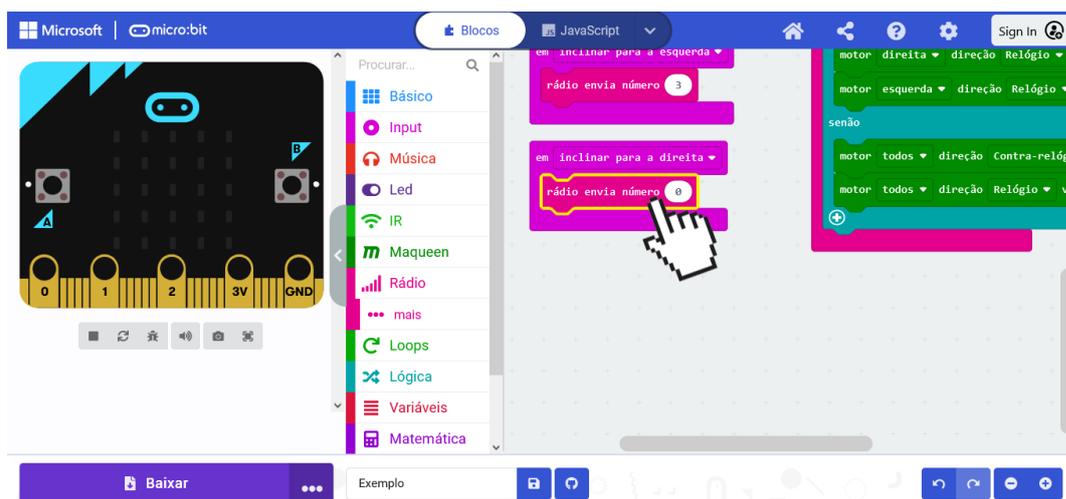


Figura 22.74: Montando um comando.

# Carrinho de controle remoto

58. Em **rádio envia número 0**, que está dentro do bloco **em inclinar para a direita**, altere o número de envio de **0** para **2**, Figura 22.75;

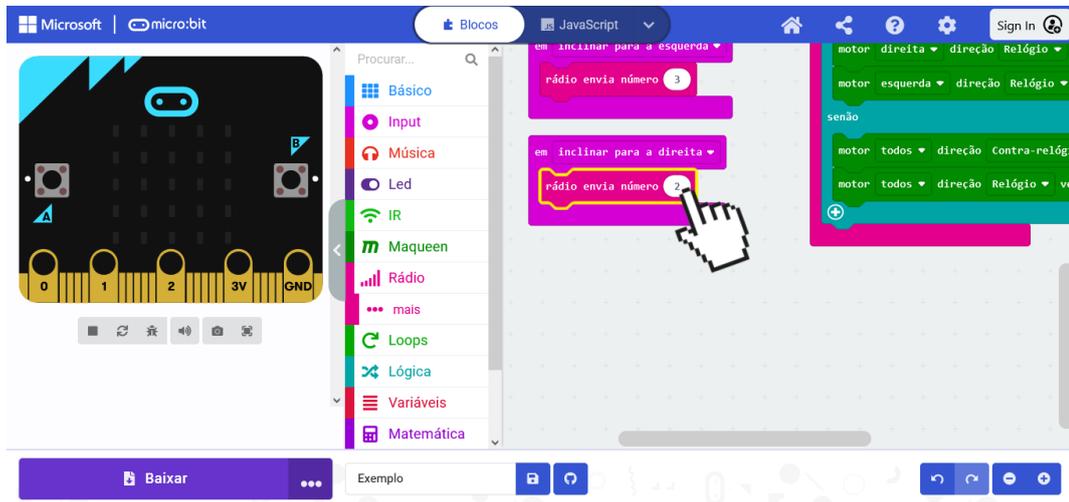


Figura 22.75: Alterando o número de envio de um rádio.

59. Em **Input**, selecione o bloco de acionamento **no botão A pressionado**, Figuras 22.76 e 22.77;



Figura 22.76: Selecionando um bloco de acionamento.



# Carrinho de controle remoto

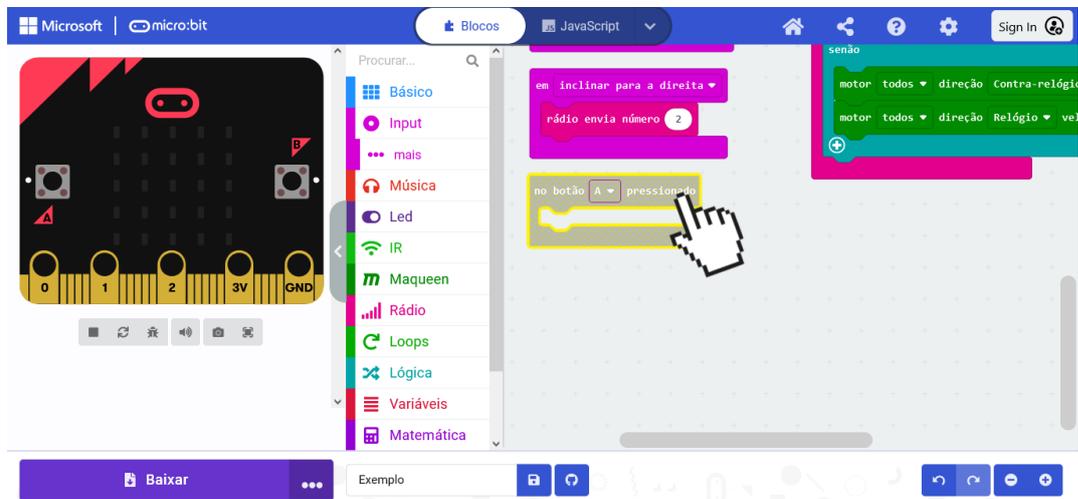


Figura 22.77: Montando um bloco de acionamento.

60. Em **no botão A pressionado**, altere o botão A para **A+B**, Figura 22.78;

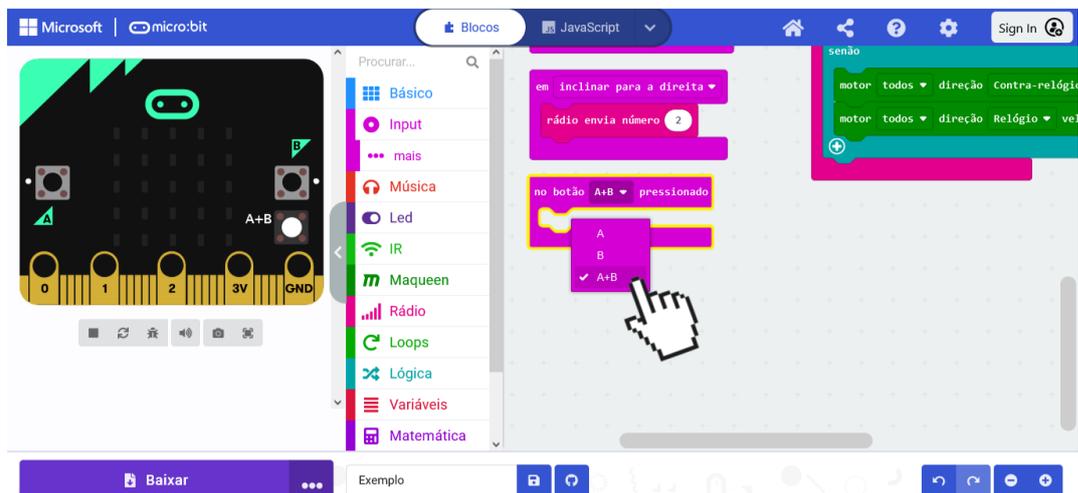


Figura 22.78: Alterando um acionamento.

# Carrinho de controle remoto

**61.** Em **Rádio**, selecione o comando **rádio envia número 0**, Figura 22.79;

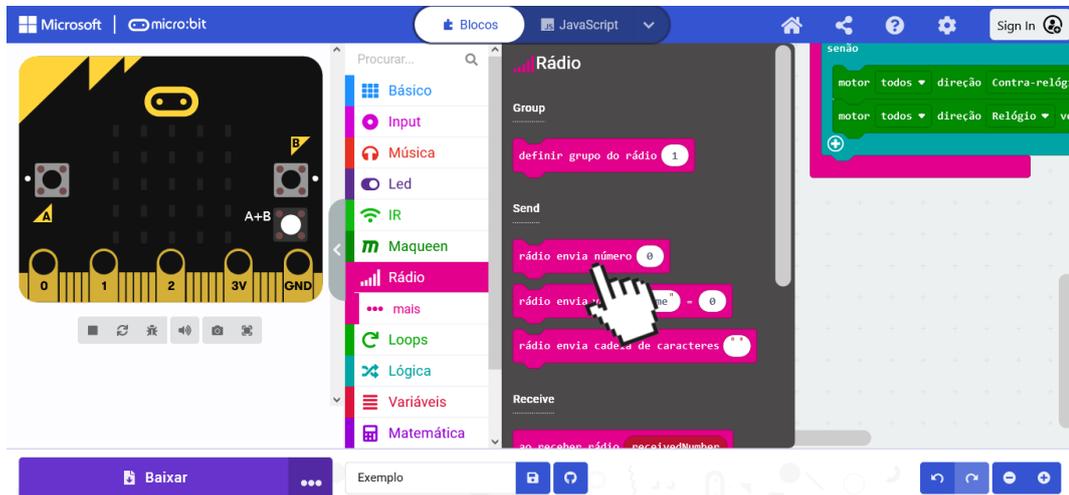


Figura 22.79: Selecionando um comando.

**62.** Adicione o último comando selecionado dentro do bloco de acionamento **no botão A+B pressionado**, Figura 22.80;

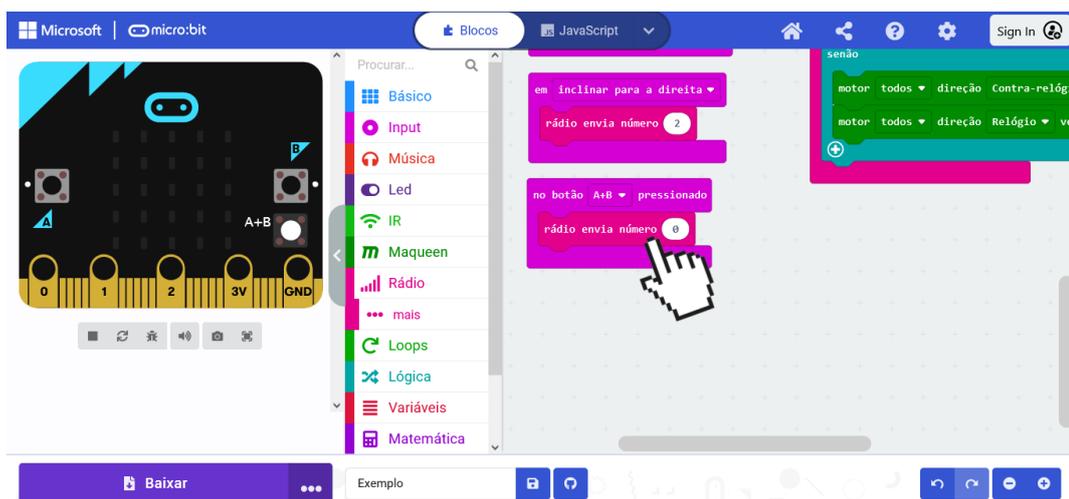


Figura 22.80: Montando um comando.



# Carrinho de controle remoto

**63.** No ultimo comando de **rádio envia número 0**, que está dentro do bloco **no botão A+B pressionado**, altere o número **0** para **4**, Figuras 22.81 e 22.82.

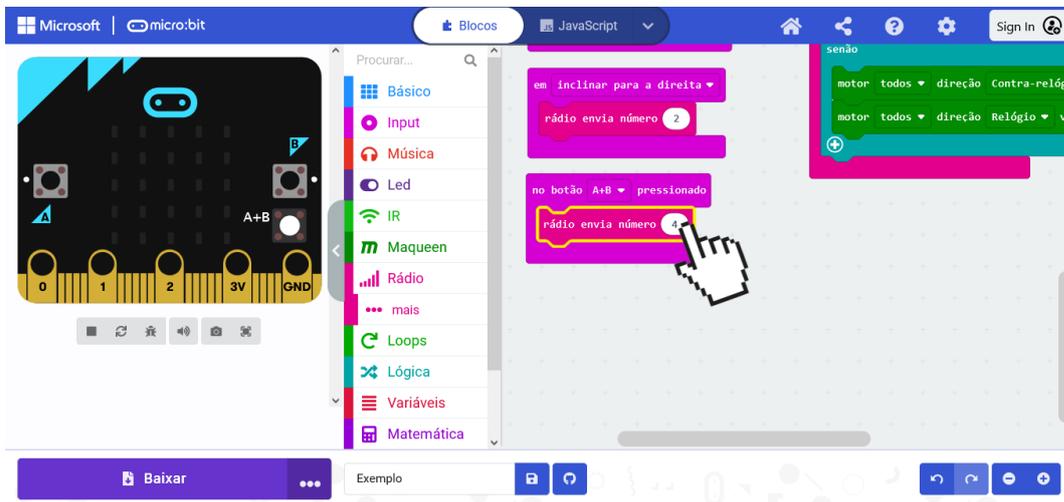


Figura 22.81: Alterando o número de envio de um rádio.

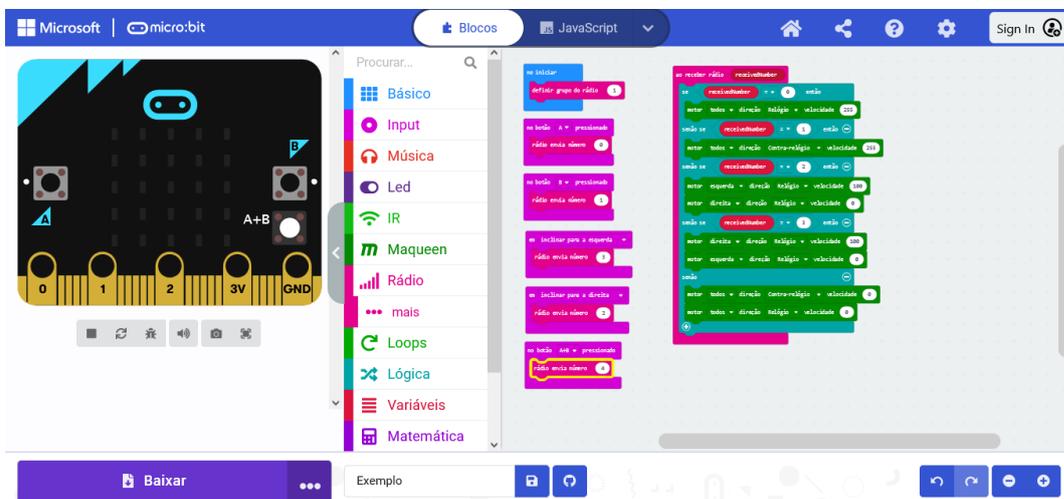


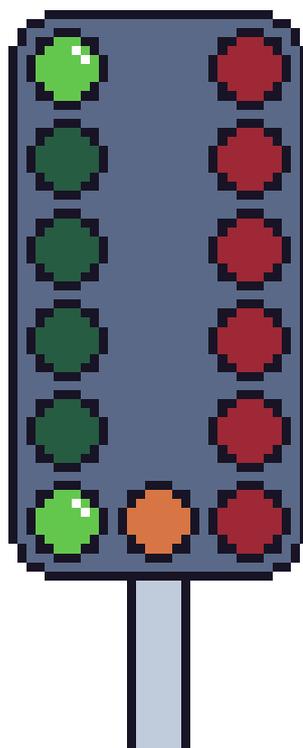
Figura 22.82: Código completo do carrinho de controle remoto.

**Baixe o código para duas placas micro:bit, uma irá encaixada no carrinho e a outra será o controle remoto. Divirta-se!**

[Clique aqui para acessar o código.](#)

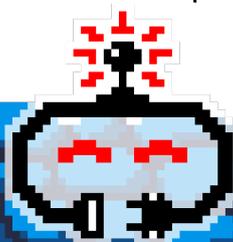


# Capítulo 23: Semáforo



Um semáforo é um sinal de trânsito que funciona como um instrumento de controle do tráfego de automóveis e pedestres nas estradas.

Para os carros, a cor vermelha indica o pare, a amarela é atenção e verde é siga. Um semáforo igualzinho ao de trânsito é o que faremos nesta atividade.



# Semáforo

## Passo a passo:

1. Clique na aba **Extensões** e busque por **neopixel** na barra de pesquisa, como nas Figuras 23.1 e 23.2;

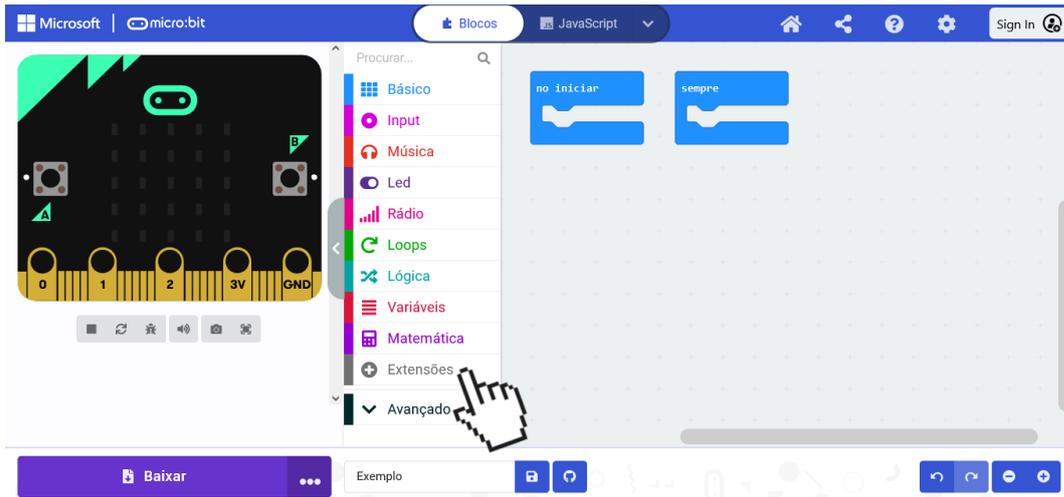


Figura 23.1: Selecionando uma extensão.

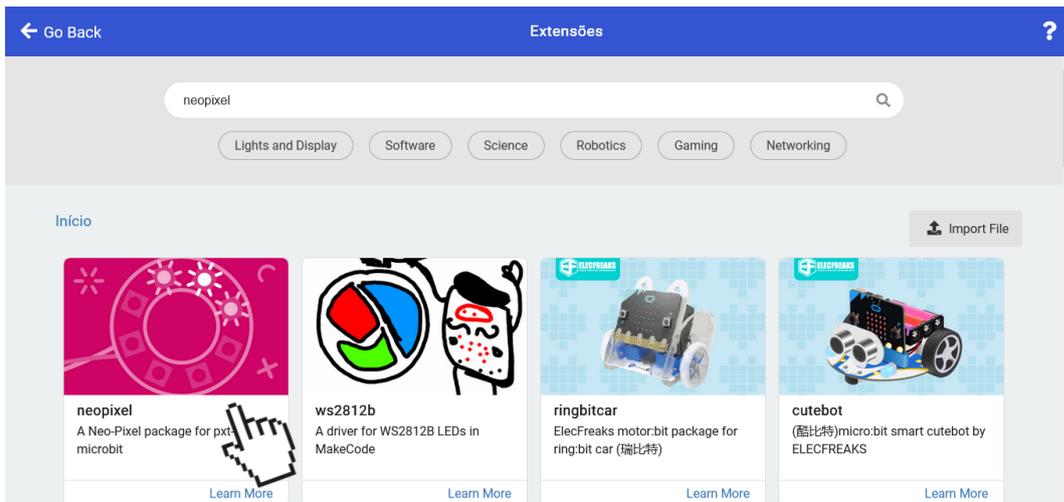


Figura 23.2: Adicionando uma extensão.

# Semáforo

2. Em Neopixel, selecione o comando **definir strip para NeoPixel at pin P0 with 24 leds as RGB (GRB format)** e coloque-o dentro do bloco de acionamento **no iniciar**, Figuras 23.3 e 23.4;

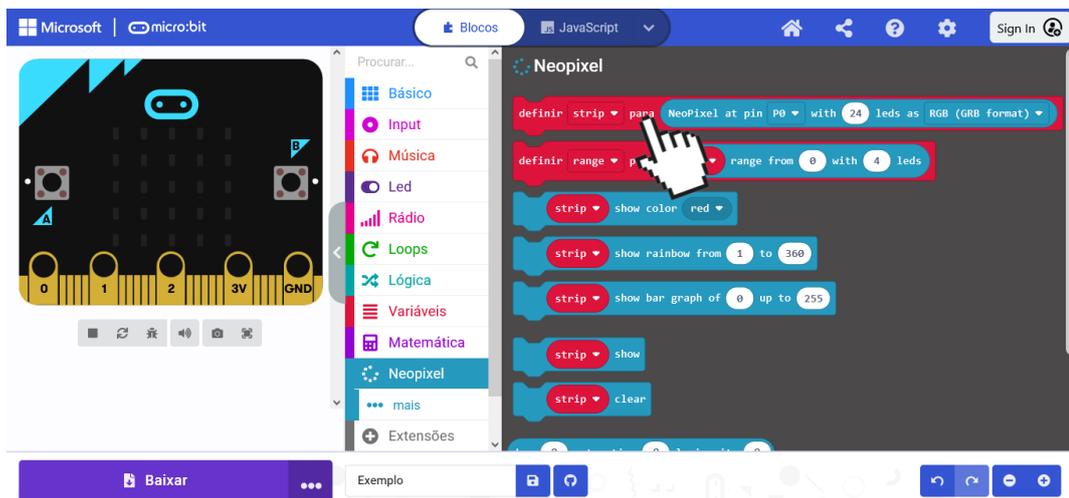


Figura 23.3: Selecionando um comando.



Figura 23.4: Montando um comando.



# Semáforo

3. No comando **definir strip para NeoPixel at pin P0 with 24 leds as RGB (GRB format)**, altere pin **P0** para **P15**, que é o pino padrão para a conexão das luzes do carrinho com a placa, Figura 23.5;

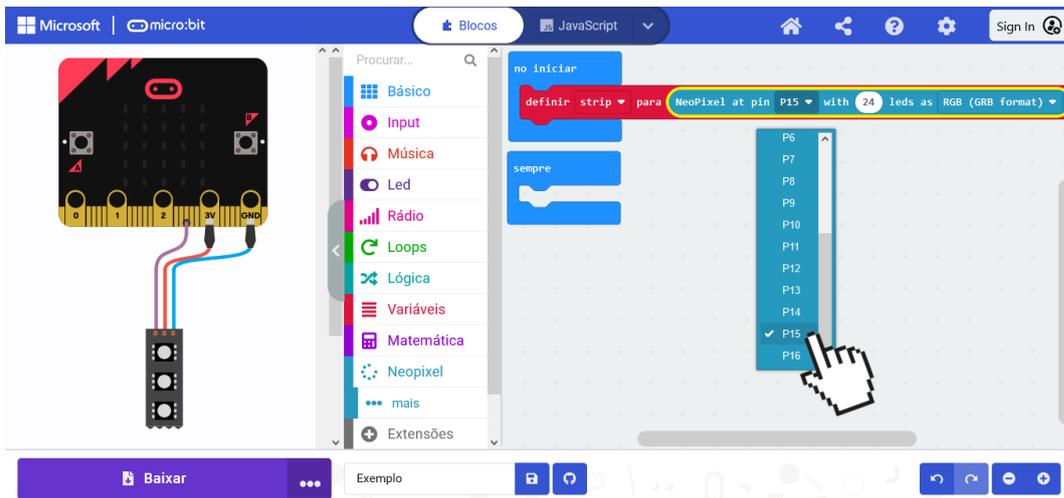


Figura 23.5: Alterando um pino de conexão.

4. Em **Neopixel**, selecione o comando **strip show color red**, Figura 23.6;

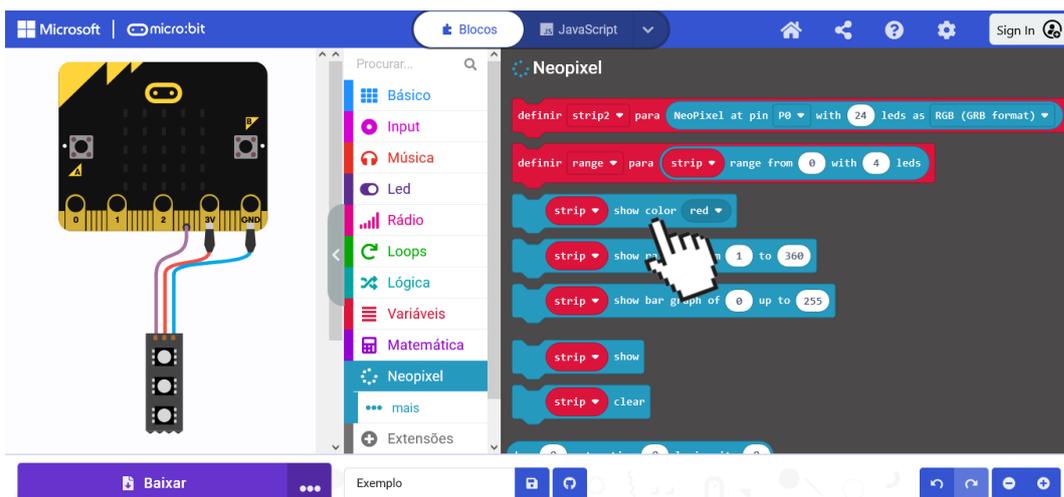


Figura 23.6: Selecionando um comando.

# Semáforo

5. Insira o último comando selecionado no bloco de acionamento **sempre**, Figura 23.7;

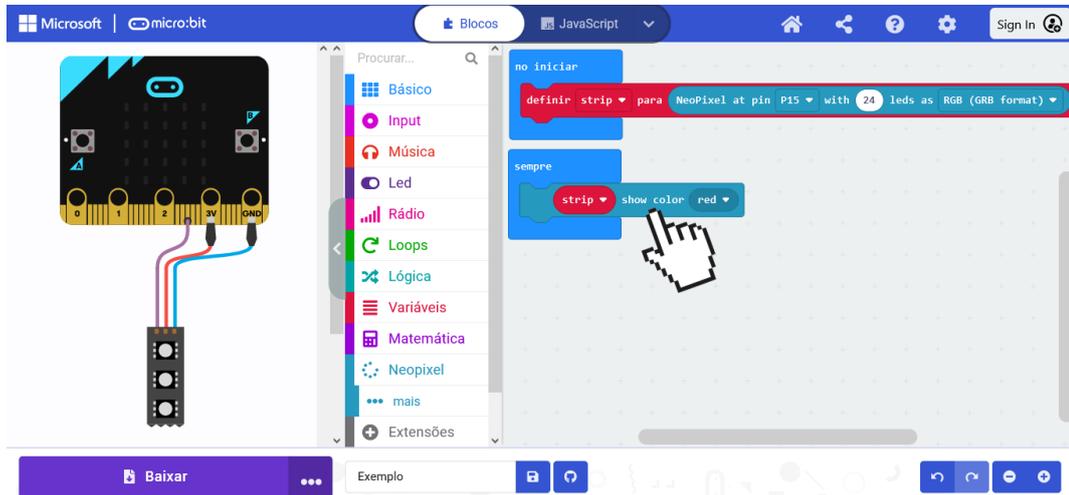


Figura 23.7: Montando um comando.

6. Em **Básico**, selecione um comando de **pausa (ms) 100**, Figura 23.8;

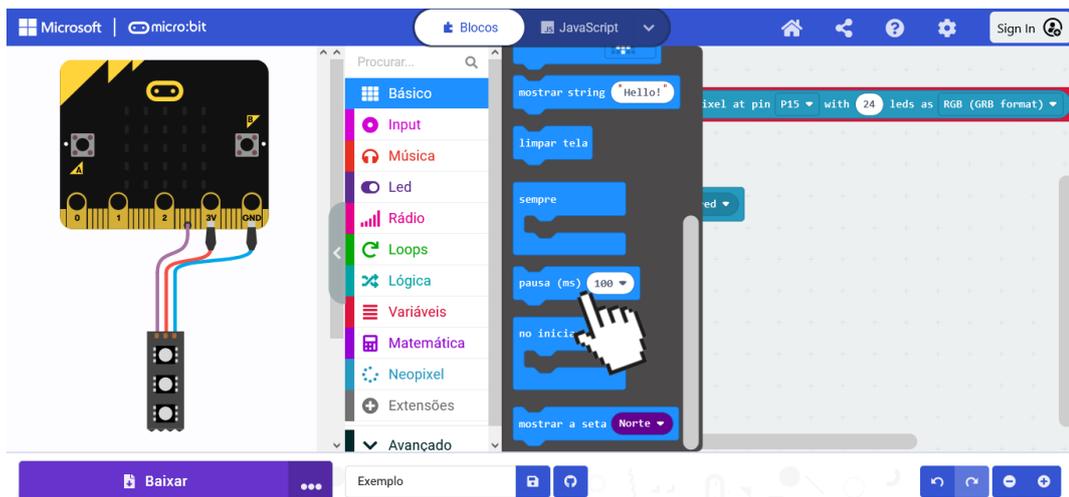


Figura 23.8: Selecionando um comando.



# Semáforo

7. Insira a **pausa** logo abaixo do comando **strip show color red**, Figura 23.9;

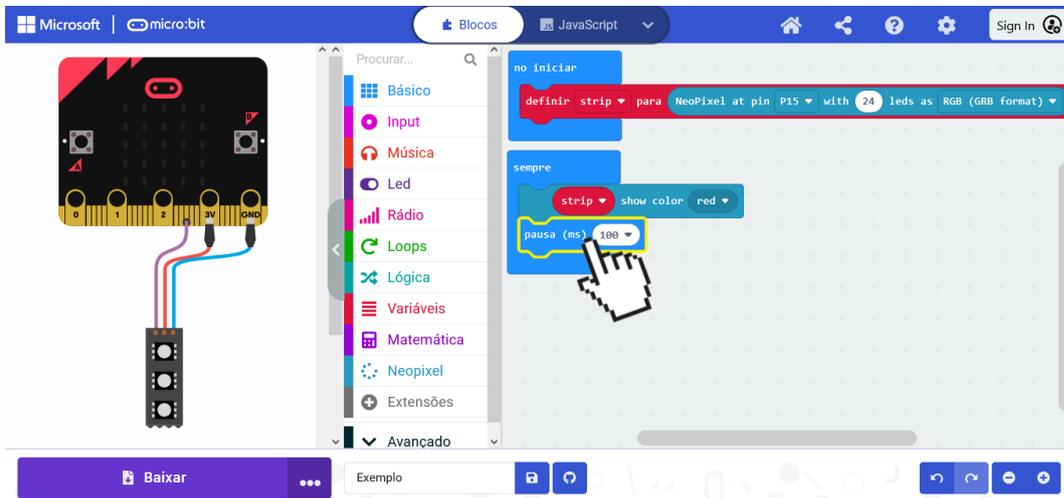


Figura 23.9: Montando um comando.

8. Altere o tempo da pausa de **100 ms** para **5 segundos**, Figura 23.10;

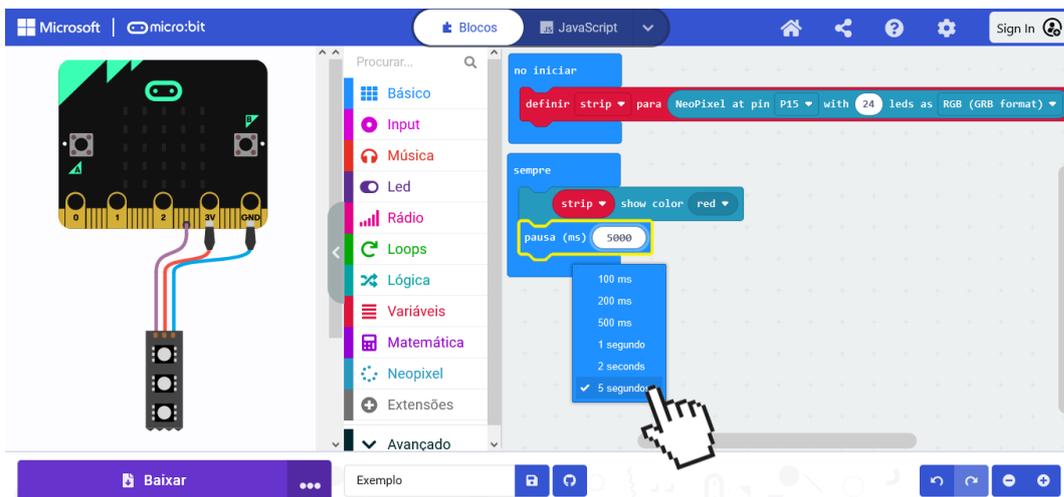


Figura 23.10: Alterando o tempo de uma pausa.

# Semáforo

9. Novamente em **Neopixel**, selecione o comando **strip show color red**, Figura 23.11;

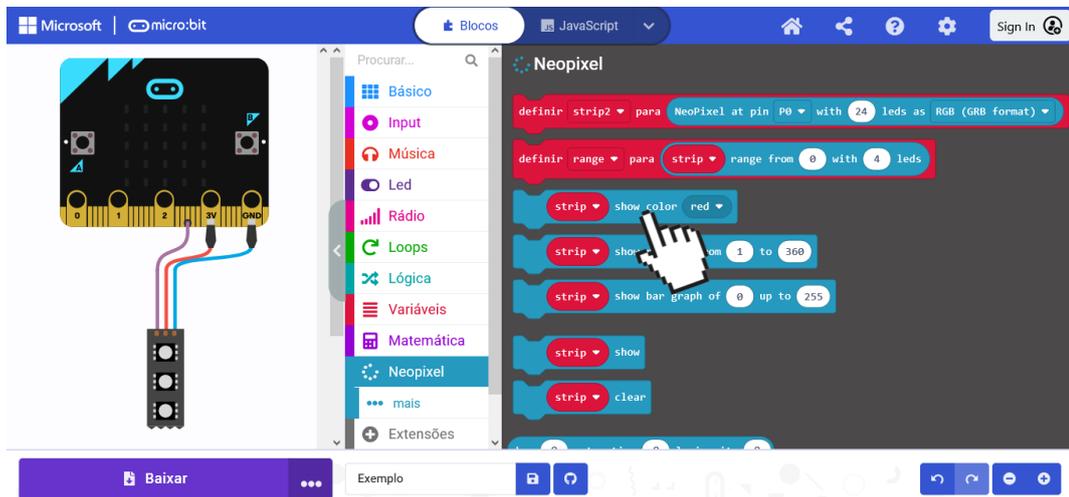


Figura 23.11: Selecionando um comando.

10. Insira o último comando selecionado logo abaixo da **pausa (ms) 5000**, Figura 23.12;



Figura 23.12: Montando um comando.



# Semáforo

- 11.** No último **strip show color red**, altere a cor **red** para **green**, como na Figura 23.13;



Figura 23.13: Alterando uma cor de LED.

- 12.** Em **Básico**, selecione um comando de **pausa (ms) 100**, Figura 23.14;



Figura 23.14: Selecionando uma pausa.

# Semáforo

**13.** Insira a **pausa** logo abaixo do comando **strip show color green**, Figura 23.15;

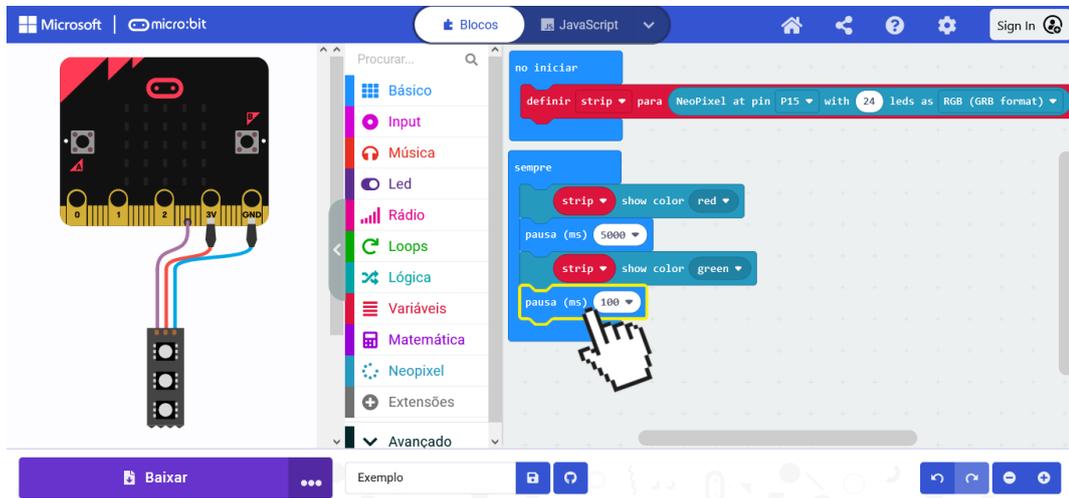


Figura 23.15: Montando uma pausa.

**14.** Altere o tempo da **pausa** de **100** para **4000**, como na Figura 23.16;

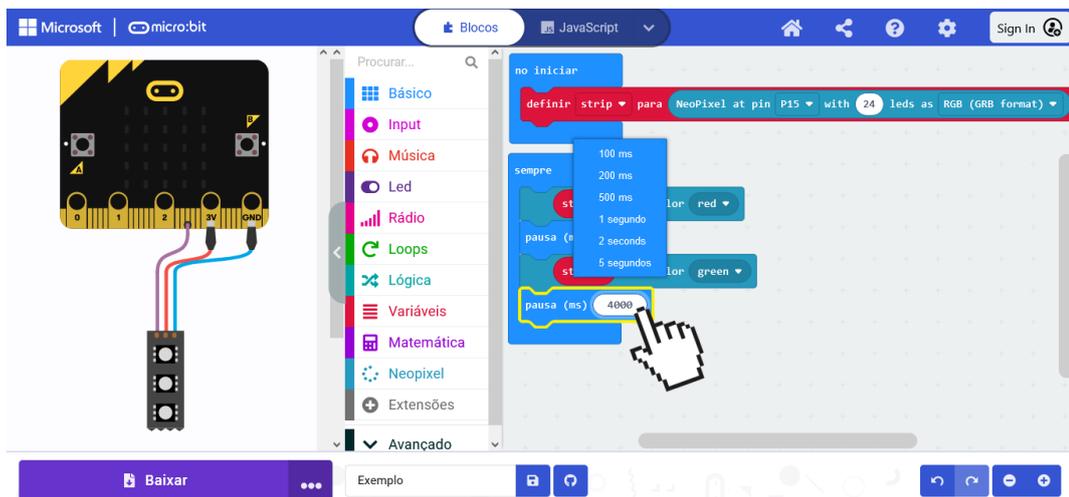


Figura 23.16: Alterando uma pausa.



# Semáforo

15. Em **Neopixel**, selecione o comando **strip show color red** e coloque-o logo abaixo da **pausa ms 4000**, como nas Figuras 23.17 e 23.18;



Figura 23.17: Selecionando um comando.



Figura 23.18: Montando um comando.

# Semáforo

**16.** No último **strip show color red**, altere a cor **red** para **yellow**, Figura 23.19;

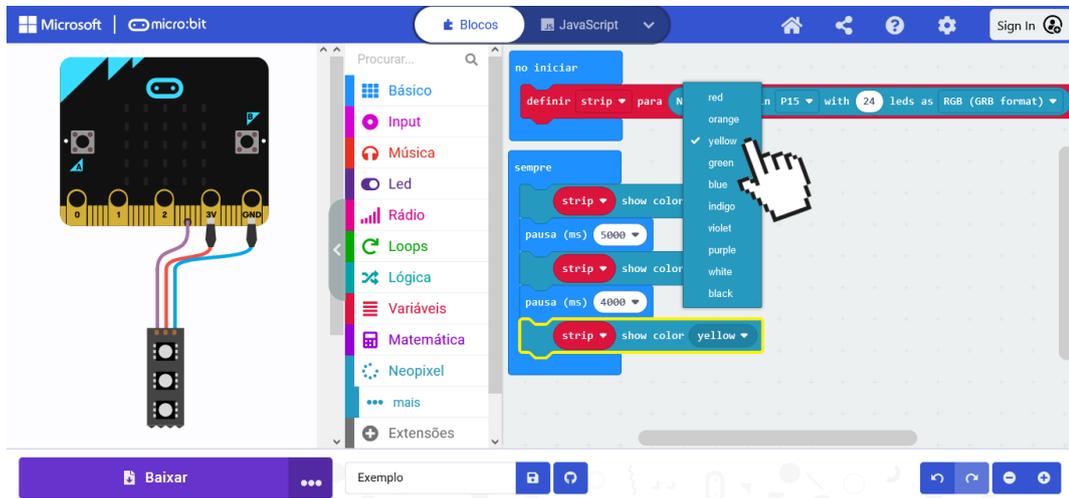


Figura 23.19: Alterando uma cor de LED.

**17.** Em **Básico**, selecione um comando de **pausa (ms) 100**, Figura 23.20;



Figura 23.20: Selecionando uma pausa.



# Semáforo

**18.** Adicione a **pausa** logo abaixo do comando **strip show color yellow**, Figura 23.21;



Figura 23.21: Montando uma pausa.

**19.** Altere a **pausa (ms) 100**, para **1 segundo** Figura 23.22;

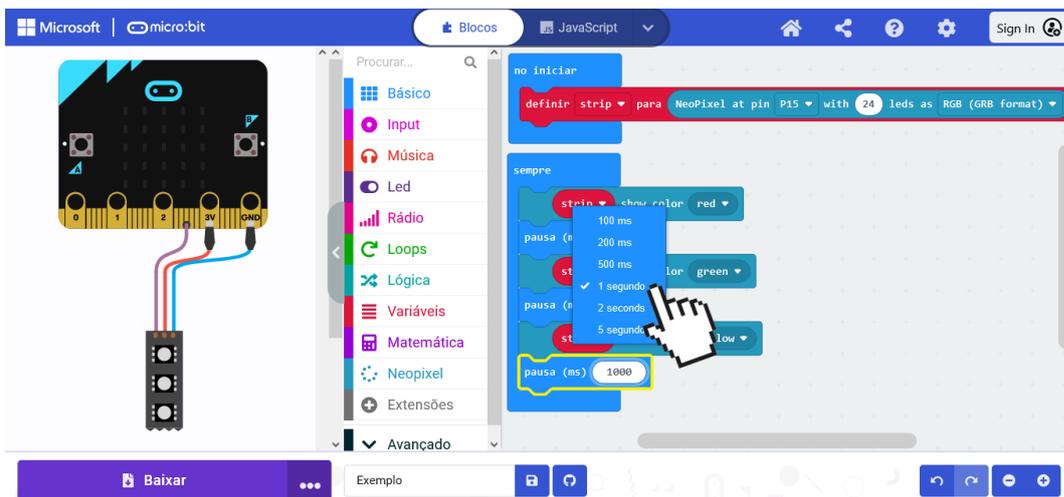


Figura 23.22: Alterando o tempo de uma pausa.

# Semáforo

- 20.** No bloco de acionamento **no iniciar**, altere o número de **24 leds** para **4**, como na Figura 23.23.

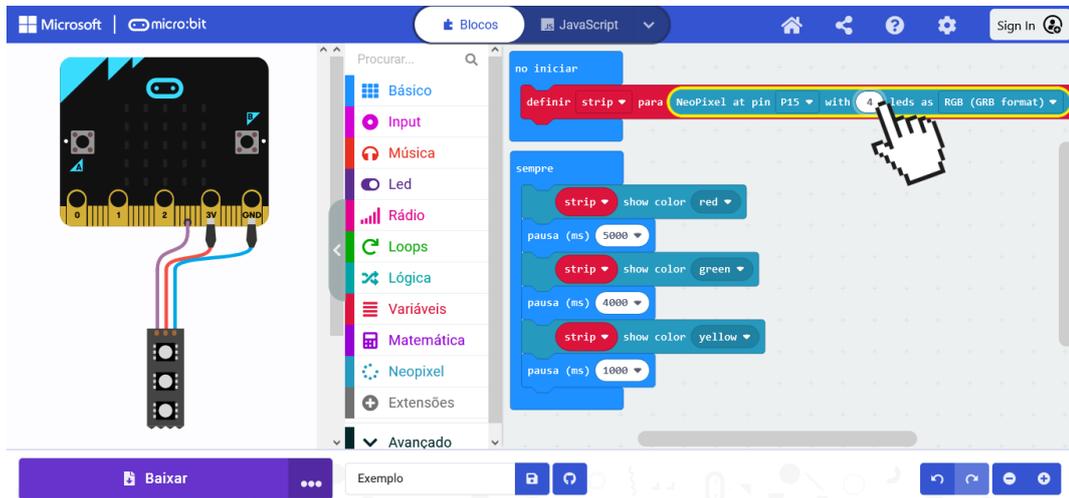


Figura 23.23: Alterando o número de LEDs.

**Baixe o programa para o micro:bit e encaixe a placa no Maqueen para ver os LEDs funcionarem com um semáforo.**

[Clique aqui para acessar o código.](#)





## Semáforo



# Capítulo 24: Carro colorido



Nesta atividade faremos com que nosso carro brilhe com luzes coloridas que mudam de cor com o passar do tempo.



# Carro colorido

## Passo a passo:

1. Clique em **Extensões** para abrir a barra de pesquisa das bibliotecas de funções e busque por **Neopixel**, Figuras 24.1 e 24.2;

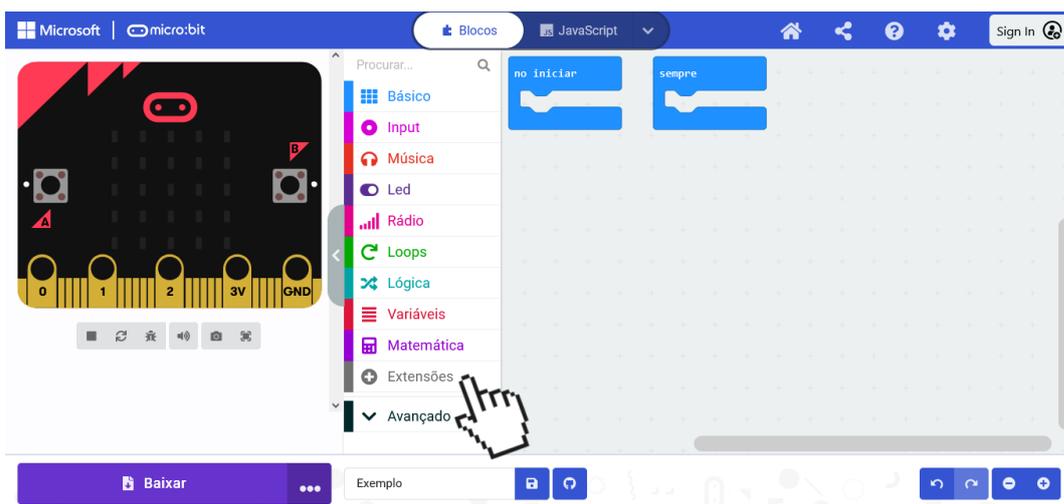


Figura 24.1: Selecionando extensões.

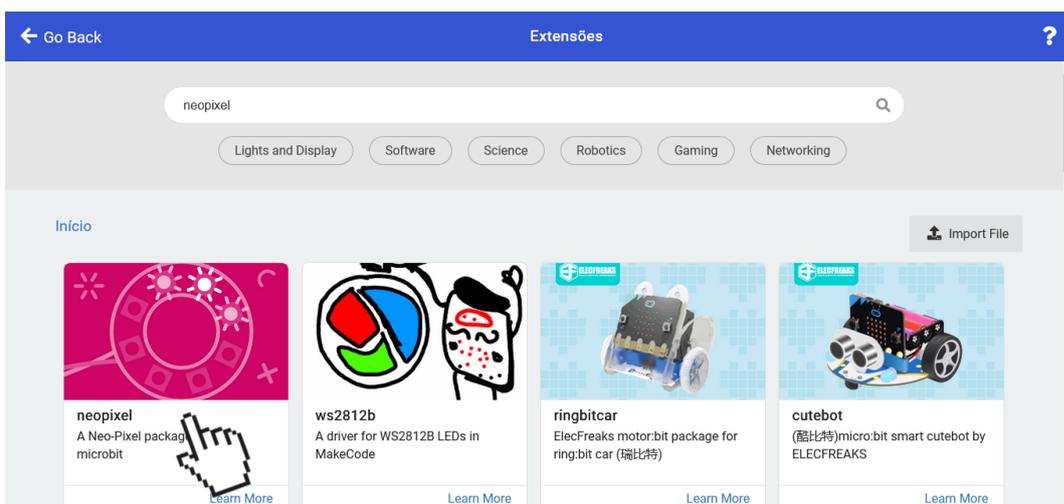


Figura 24.2: Adicionando bibliotecas.

# Carro colorido

2. Na aba **Variáveis**, faça as variáveis de nome **b**, **g** e **r**, como nas Figuras 24.3, 24.4, 24.5, 24.6, 24.7 e 24.8;

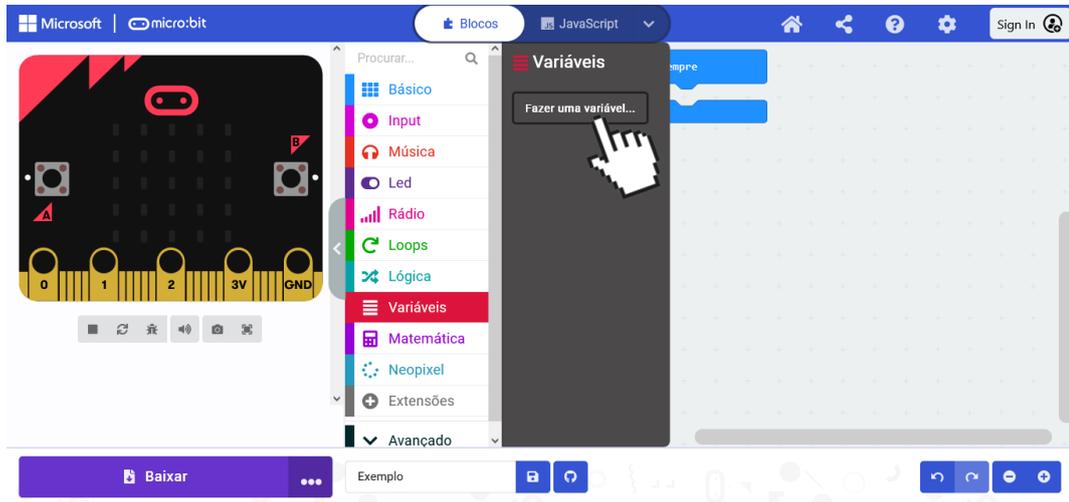


Figura 24.3: Fazendo uma variável.

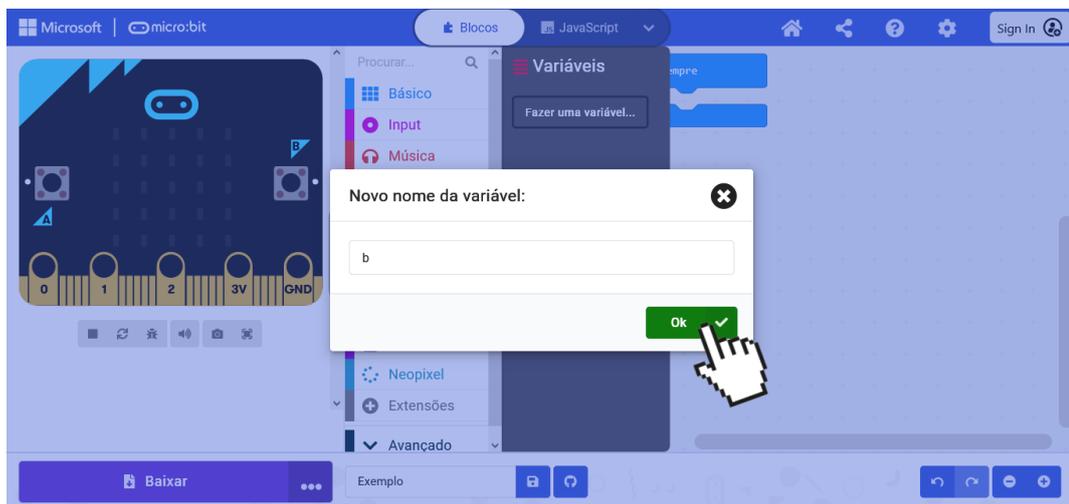


Figura 24.4: Dando nome a uma variável.



# Carro colorido

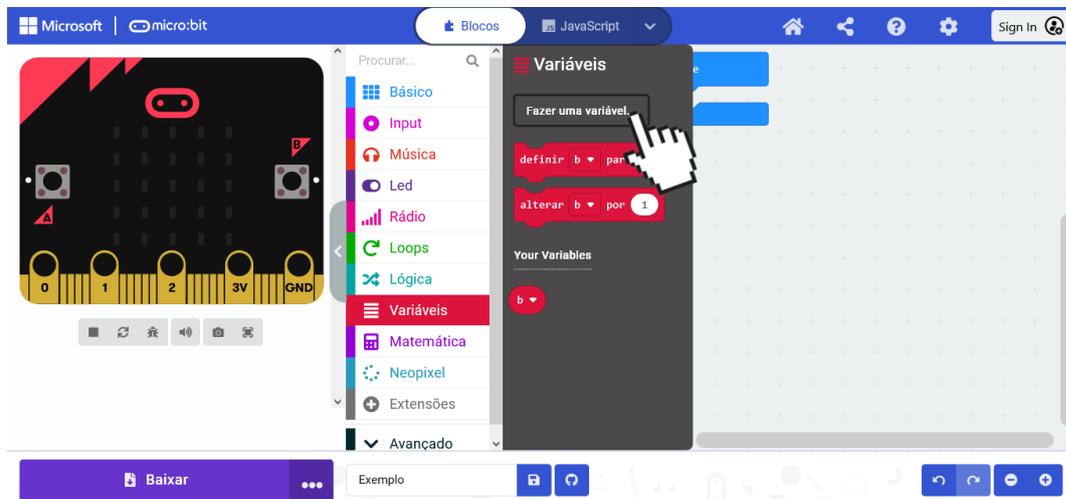


Figura 24.5: Fazendo uma variável.

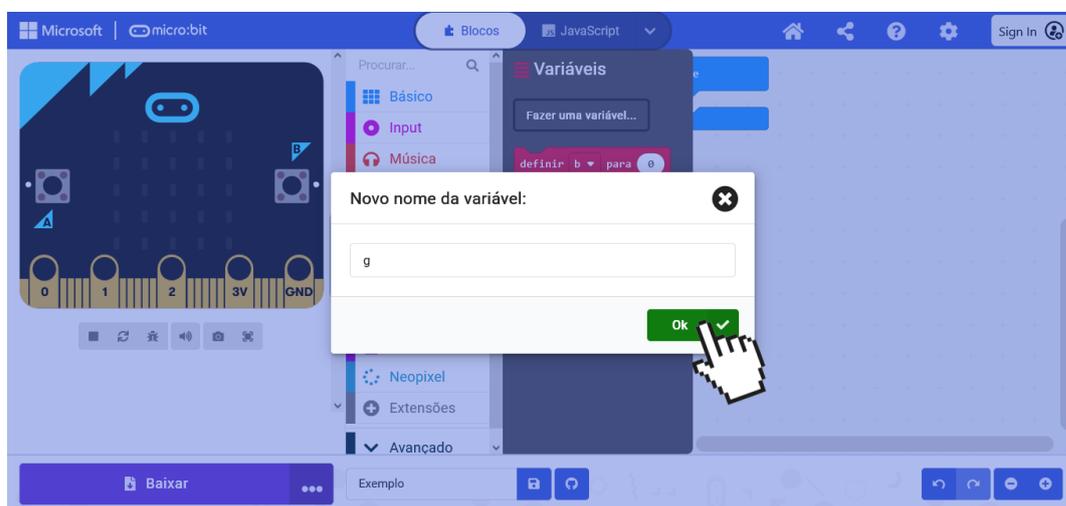


Figura 24.6: Dando nome a uma variável.

# Carro colorido

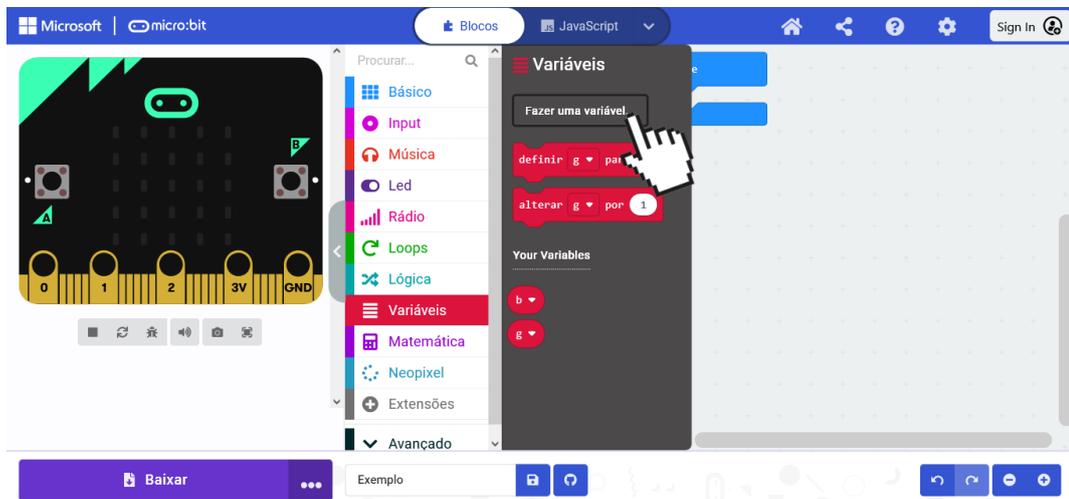


Figura 24.7: Fazendo uma variável.

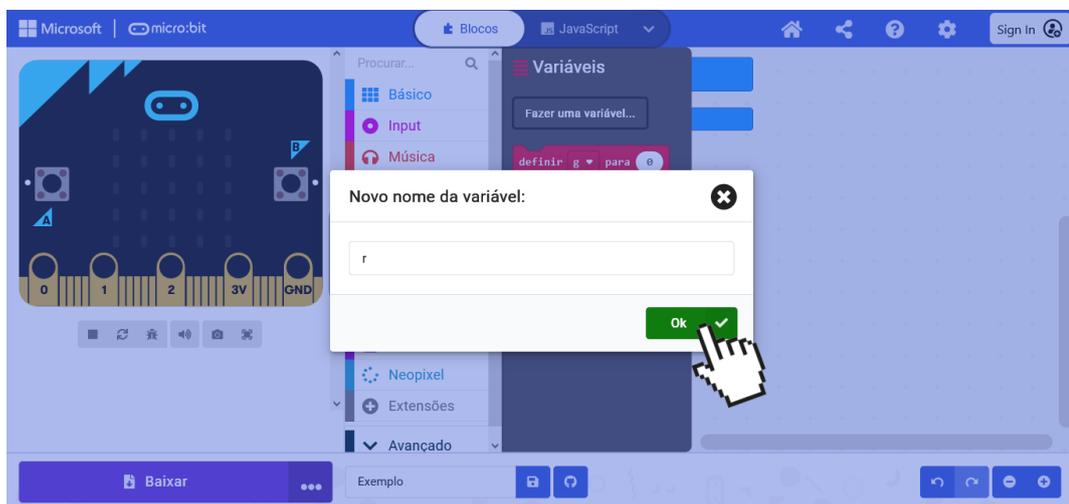


Figura 24.8: Dando nome a uma variável.



# Carro colorido

3. Em **Variáveis**, selecione o comando **definir r para 0**, Figura 24.9;

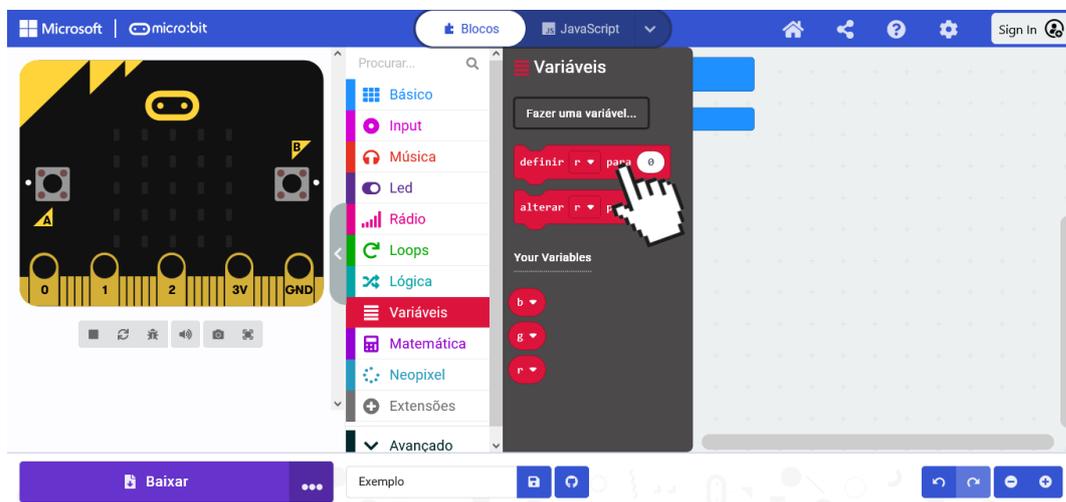


Figura 24.9: Selecionando um comando.

4. Coloque-o dentro do bloco de acionamento **no iniciar**, Figura 24.10;



Figura 24.10: Montando um comando.

# Carro colorido

5. Em **definir r para 0**, altere a variável **r** para **b** e o número **0** para **255**, Figuras 24.11 e 24.12;



Figura 24.11: Selecionando uma variável.

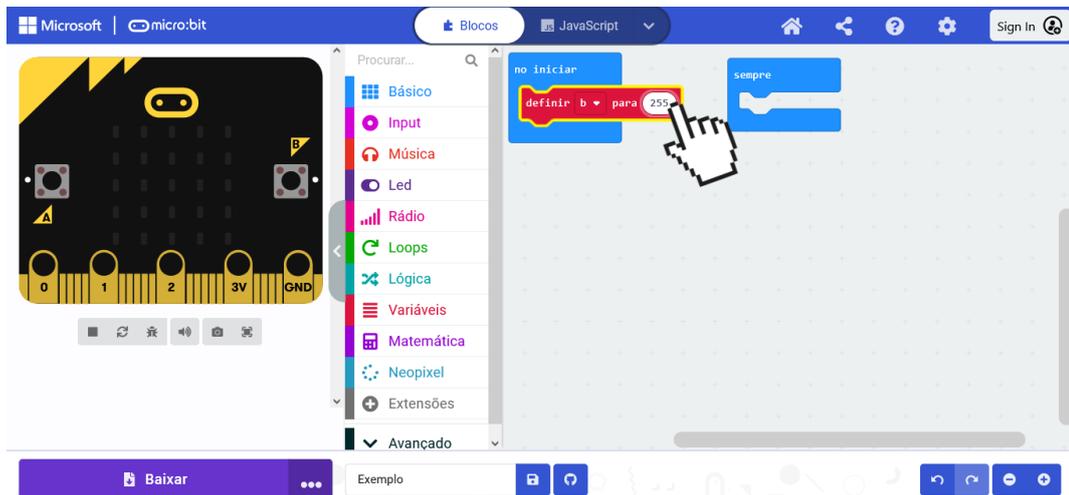


Figura 24.12: Definindo uma variável.



# Carro colorido

6. Em **Variáveis**, selecione o comando **definir r para 0**, Figura 24.13;

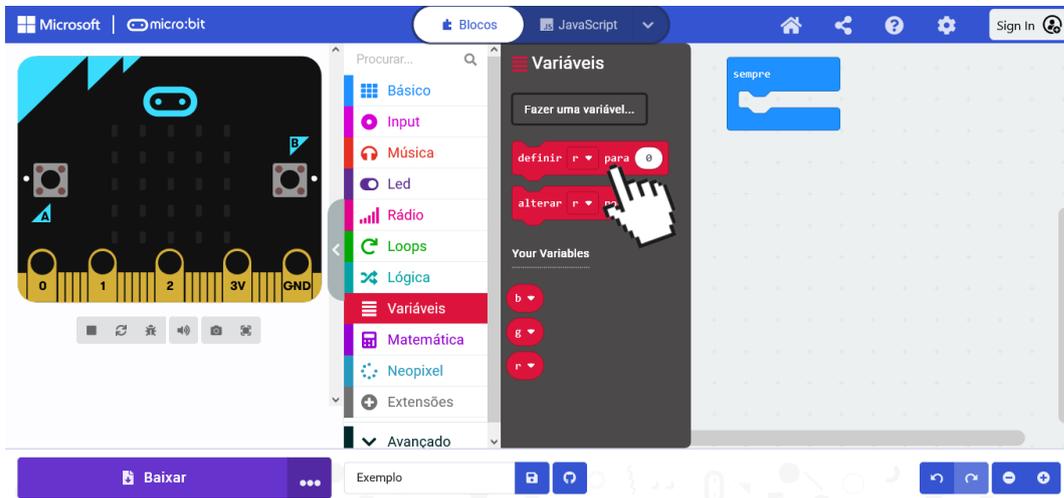


Figura 24.13: Selecionando um comando.

]

7. Insira o último comando selecionado logo abaixo de **definir b para 255**, Figura 24.14;



Figura 24.14: Montando um comando.

# Carro colorido

8. Em **definir r para 0**, altere a variável **r** para **g**, Figura 24.15;



Figura 24.15: Selecionando uma variável.

9. Em **Variáveis**, selecione o comando **definir r para 0**, Figura 24.16;

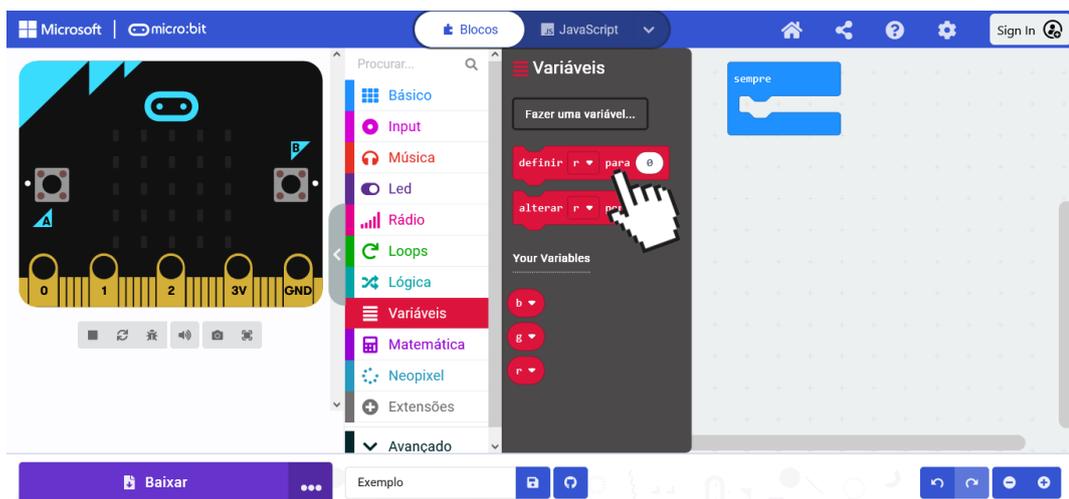


Figura 24.16: Selecionando um comando.



# Carro colorido

10. Coloque o último comando selecionado logo abaixo de **definir g para 0**, Figura 24.17;

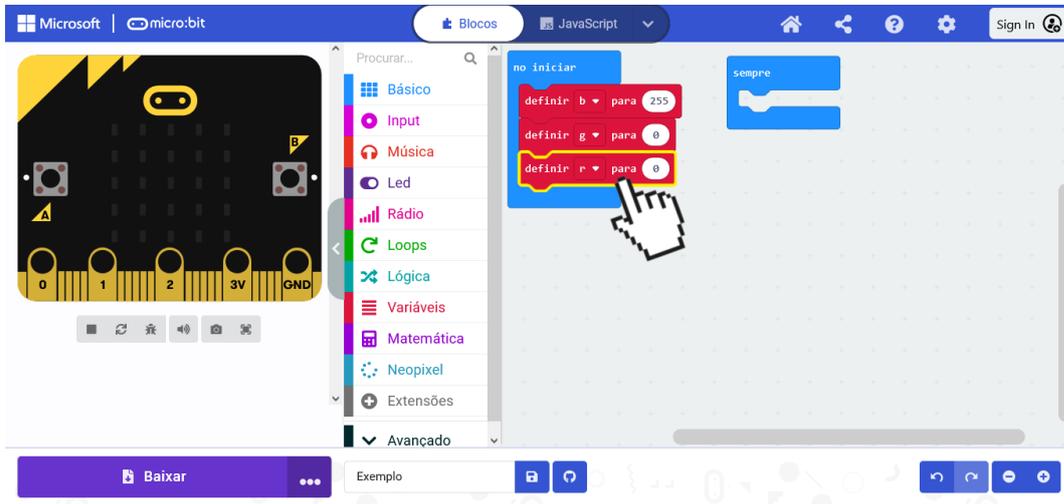


Figura 24.17: Montando um comando.

11. Na aba **Neopixel**, selecione o comando **definir strip para NeoPixel at pin P0 with 24 leds as RGB (GRB format)**, Figura 24.18;

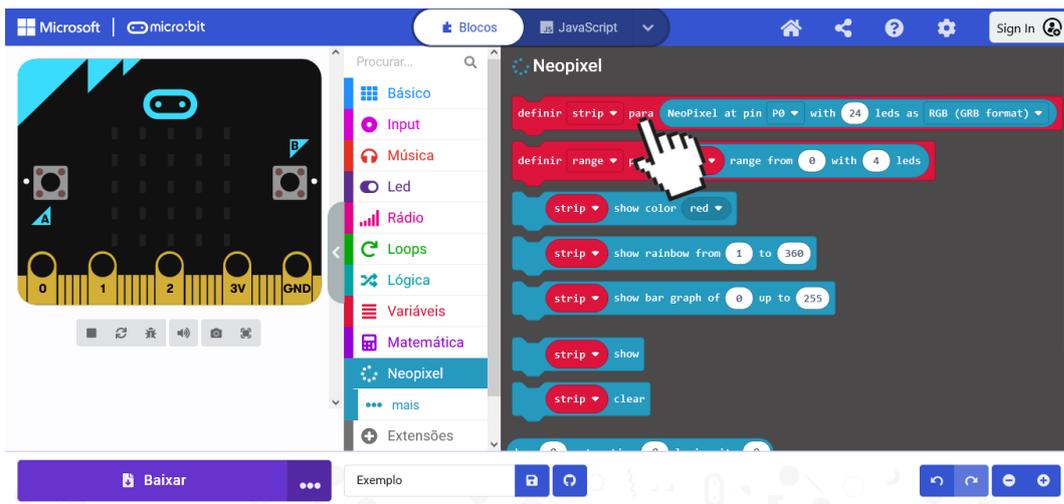


Figura 24.18: Selecionando um comando.

# Carro colorido

- 12.** Insira o último comando selecionado logo abaixo de **definir r para 0**, Figura 24.19;



Figura 24.19: Montando um comando.

- 13.** Altere o pino **P0** para **P15**, Figura 24.20;

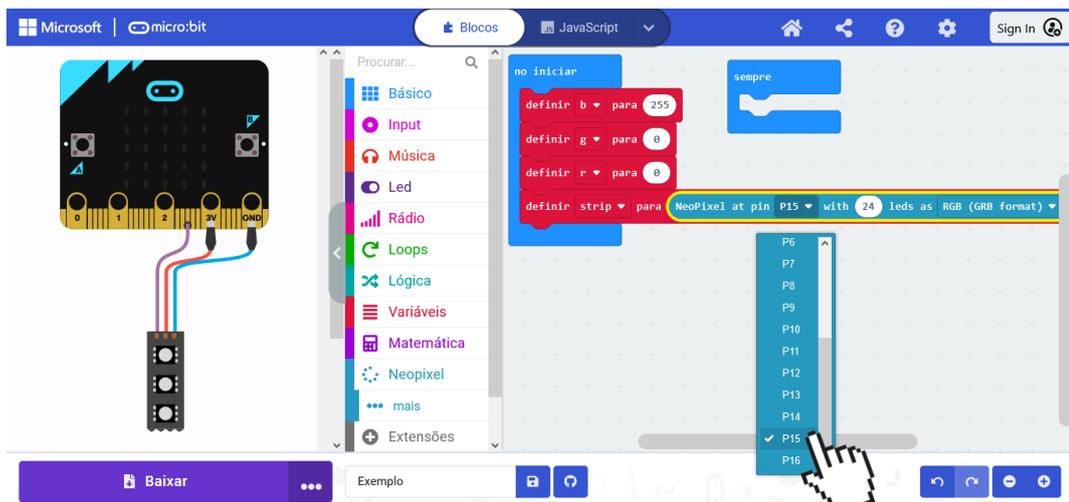


Figura 24.20: Selecionando um pino.



# Carro colorido

14. Na aba **Loops**, selecione o bloco de repetição **para índice de 0 a 4 executar**, Figura 24.21;



Figura 24.21: Selecionando um bloco de repetição.

15. Insira o último bloco selecionado dentro do bloco de acionamento **sempre**, Figura 24.22;



Figura 24.22: Montando um bloco de repetição.

# Carro colorido

**16.** No bloco **para índice de 0 a 4 executar**, altere o número **4** para **255**, Figura 24.23;



Figura 24.23: Alterando um índice.

**17.** Em **Neopixel**, selecione o comando **strip show color red**, Figura 24.24;

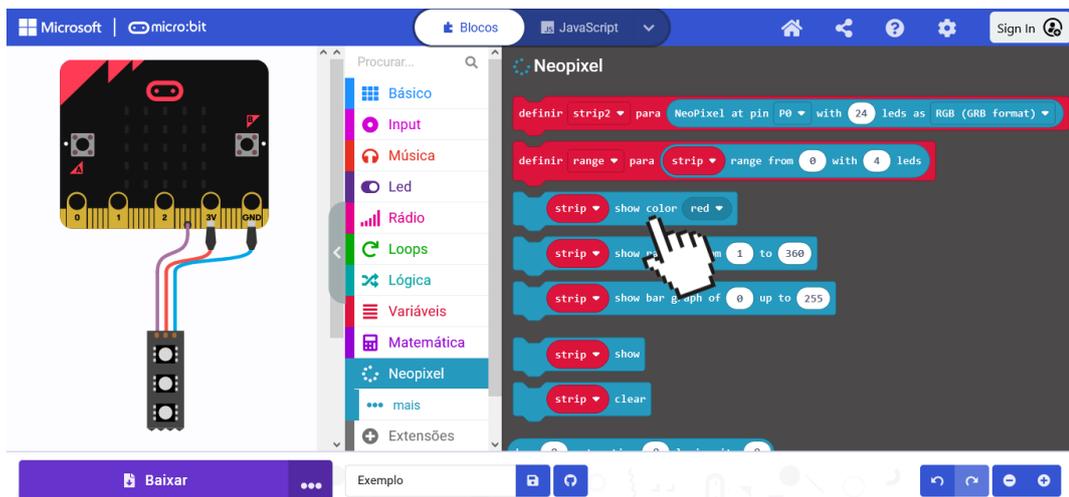


Figura 24.24: Selecionando um comando.



# Carro colorido

18. Insira o último comando selecionado dentro do bloco **para índice de 0 a 255 executar**, Figura 24.25;



Figura 24.25: Montando um comando.

19. Na aba **mais**, logo abaixo de **Neopixel**, selecione o comando **red 255 green 255 blue 255**, Figuras 24.26 e 24.27;



Figura 24.26: Selecionando mais comandos Neopixel.

# Carro colorido

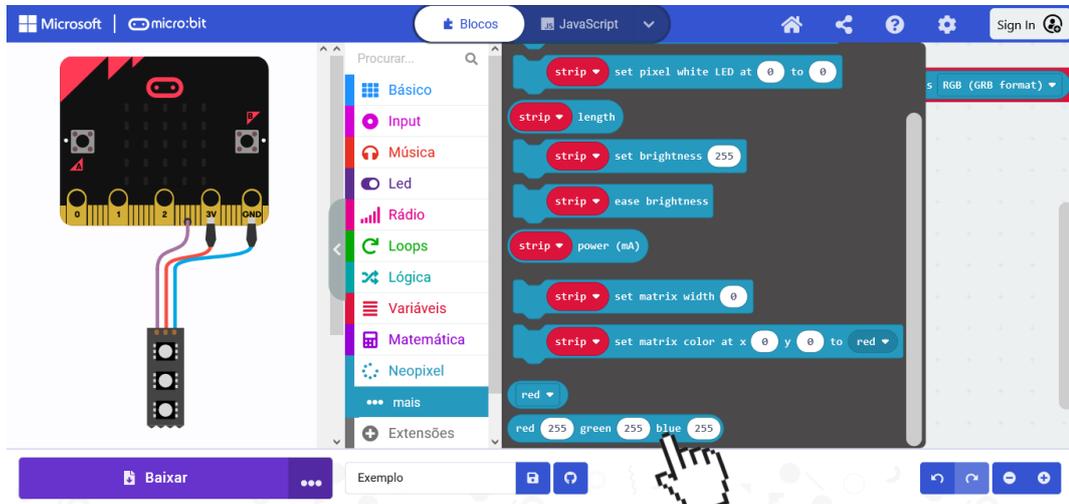


Figura 24.27: Selecionando um comando.

20. Coloque o último comando selecionado no lugar de **red**, no bloco **strip show color red**, Figura 24.28;



Figura 24.28: Montando um comando.



# Carro colorido

**21.** Em **Variáveis**, selecione a variável **r**, Figura 24.29;

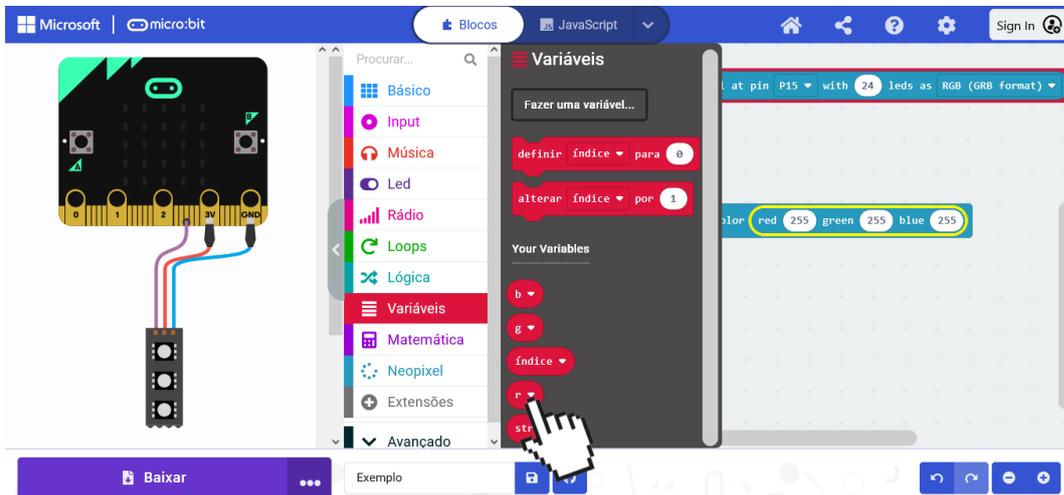


Figura 24.29: Selecionando uma variável.

**22.** Insira a variável selecionada no lugar do número **255**, logo depois de **show color red**, Figura 24.30;



Figura 24.30: Montando um comando com variáveis.

# Carro colorido

**23.** Em **Variáveis**, selecione a variável **g**, Figura 24.31;

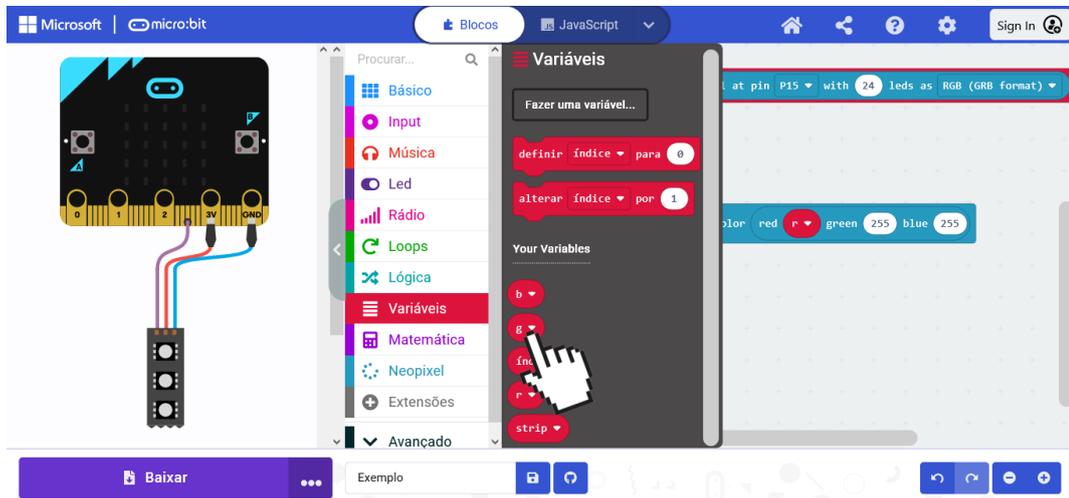


Figura 24.31: Selecionando uma variável.

**24.** Insira a variável selecionada no lugar do número **255**, logo depois da cor **green**, Figura 24.32;



Figura 24.32: Montando um comando com variáveis.



# Carro colorido

**25.** Em **Variáveis**, selecione a variável **b**, Figura 24.33;

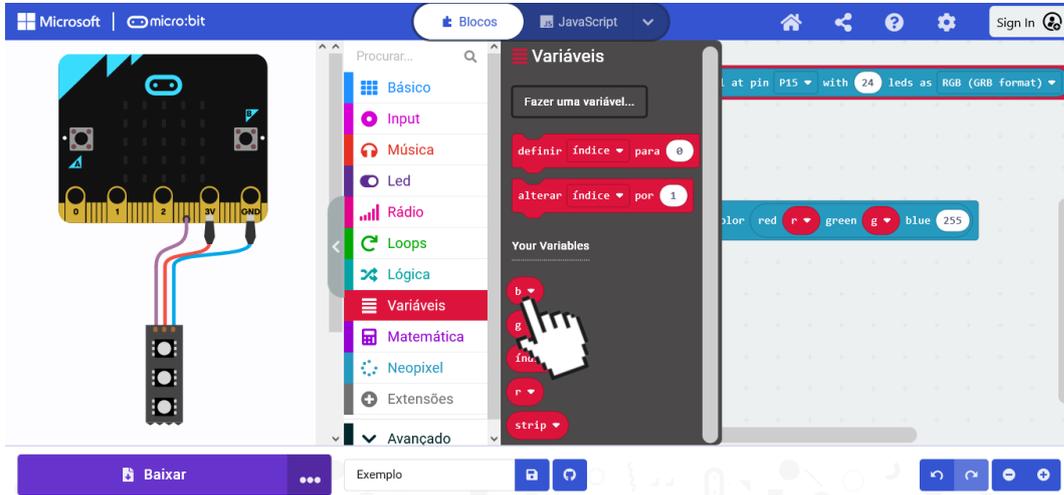


Figura 24.33: Selecionando uma variável.

**26.** Insira a variável selecionada no lugar do número **255**, logo depois da cor **blue**, Figura 24.34;



Figura 24.34: Montando um comando com variáveis.

# Carro colorido

**27.** Em **Variáveis**, selecione o comando **alterar índice por 1**, Figura 24.35;



Figura 24.35: Selecionando um comando.

**28.** Insira o último comando selecionado logo abaixo de **strip show color red r green g blue b**, Figura 24.36;



Figura 24.36: Montando um comando.



# Carro colorido

**29.** Em **alterar índice por 1**, altere o **índice** para a variável **g**, Figura 24.37;



Figura 24.37: Selecionando uma variável.

**30.** Clique sobre o bloco **para índice de 0 a 255 executar** com o botão de abrir janelas (normalmente o botão direito do mouse) e selecione a opção **duplicar**, Figura 24.38;



Figura 24.38: Duplicando um bloco.

# Carro colorido

**31.** Insira o bloco duplicado abaixo do copiado, Figura 24.39;

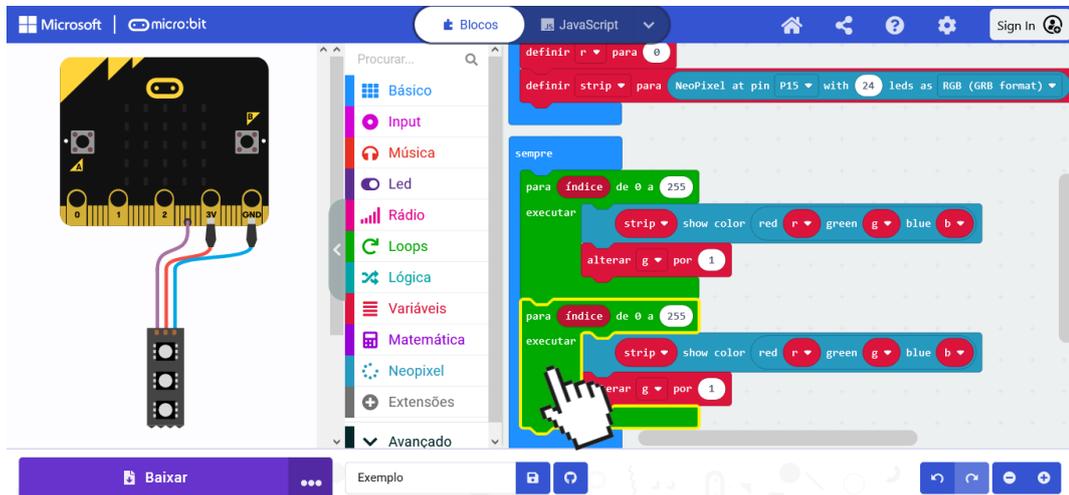


Figura 24.39: Montando um bloco.

**32.** No comando **alterar g por 1**, altere a variável **g** para **b** e o número **1** para **-1**, Figuras 24.40 e 24.41;



Figura 24.40: Selecionando uma variável.



# Carro colorido



Figura 24.41: Alterando uma variável.

**33.** Duplica novamente o bloco **para índice de 0 a 255 executar**, Figura 24.42;



Figura 24.42: Duplicando um bloco.

# Carro colorido

**34.** Insira o bloco duplicado abaixo do copiado, Figura 24.43;



Figura 24.43: Montando um bloco.

**35.** No bloco copiado, altere a variável **b** para **r** e o número **-1** para **1**, Figuras 24.44 e 24.45;

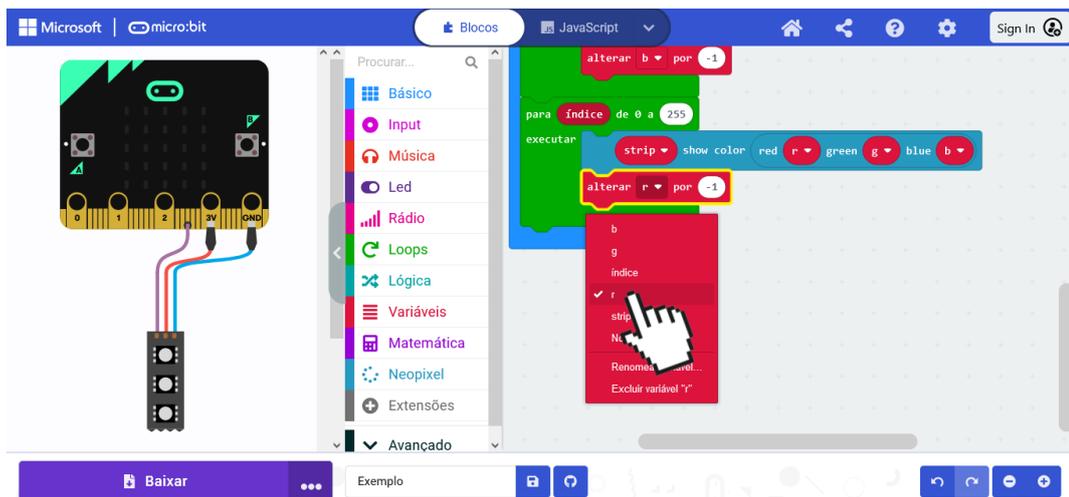


Figura 24.44: Selecionando uma variável.



# Carro colorido



Figura 24.45: Alterando uma variável.

- 36.** Duplicue também o bloco anterior e insira-o abaixo do bloco copiado, Figuras 24.46 e 24.47;



Figura 24.46: Duplicando um bloco.

# Carro colorido



Figura 24.47: Montando um bloco.

- 37.** No novo bloco duplicado, altere a variável **r** para **g** e o número **1** para **-1**, Figuras 24.48 e 24.49;

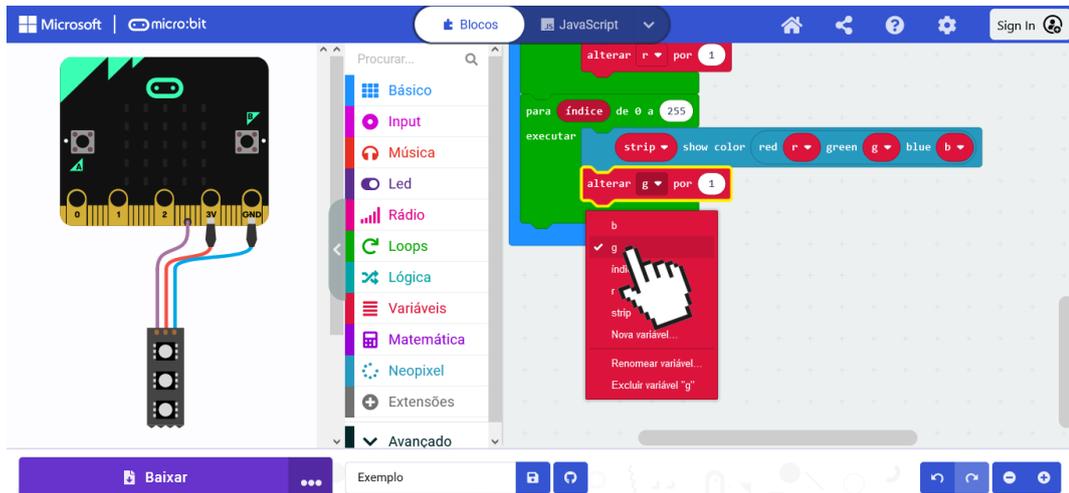
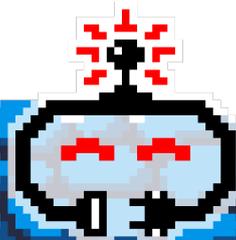


Figura 24.48: Selecionando uma variável.



# Carro colorido



Figura 24.49: Alterando uma variável.

- 38.** Duplica novamente o bloco anterior e coloque-o abaixo do bloco copiado, Figuras 24.50 e 24.51;



Figura 24.50: Duplicando um bloco.

# Carro colorido

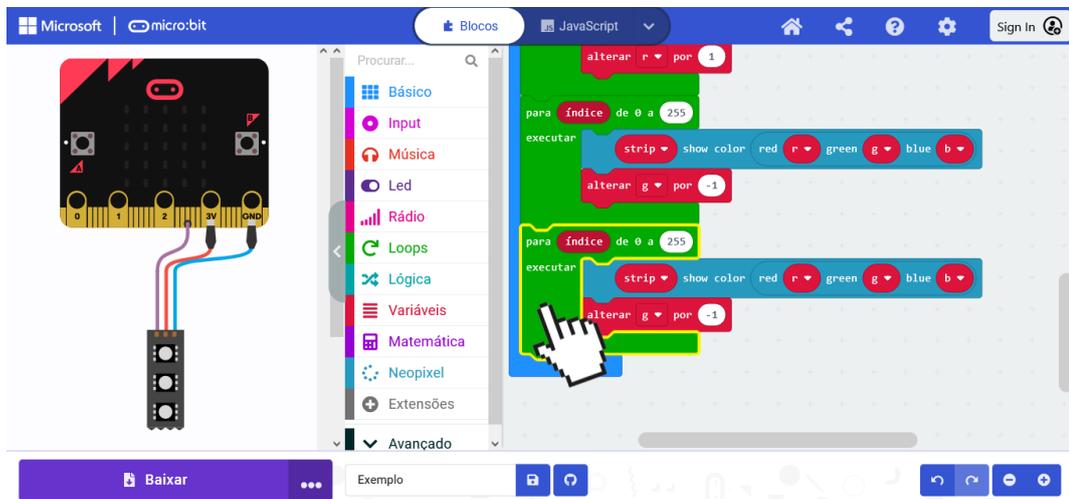


Figura 24.51: Montando um comando.

- 39.** No novo bloco, altere a variável **g** para **b** e o número **-1** para **1**, Figuras 24.52 e 24.53;



Figura 24.52: Selecionando uma variável.



# Carro colorido



Figura 24.53: Alterando uma variável.

- 40.** Duplicue o último bloco montado e insira-o abaixo do bloco copiado, Figura 24.54;



Figura 24.54: Duplicando um bloco.

# Carro colorido

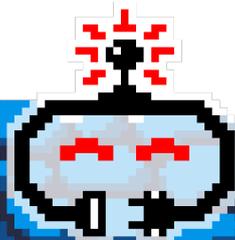
- 41.** No novo bloco, altere a variável **b** para **r** e o número **1** para **-1**, Figuras 24.55 e 24.56;



Figura 24.55: Selecionando uma variável.



Figura 24.56: Alterando uma variável.



# Carro colorido

42. Por último, volte ao comando **definir strip para NeoPixel at pin P15 with 24 leds as RGB (GRB format)** e altere o número de **24 leds** para **4**, Figura 24.57.



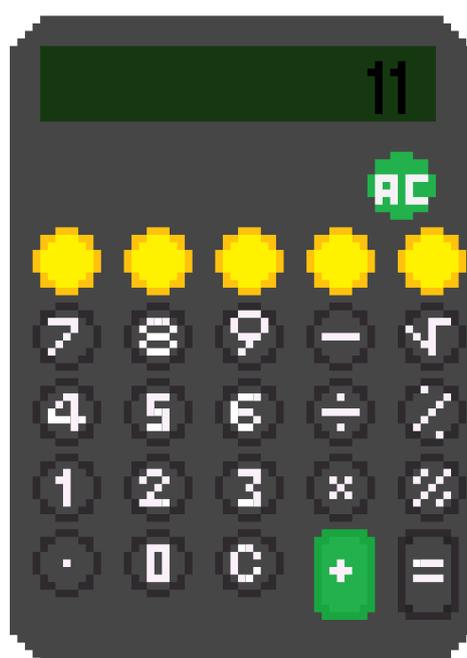
Figura 24.57: Alterando o número de LEDs.

**Baixe o código para o micro:bit e encaixe a placa no carrinho para ver o show de luzes.**

[Clique aqui para acessar o código.](#)



# Capítulo 25: Montando uma calculadora



Você já viu uma calculadora, não é mesmo? Sabe o que a calculadora faz? Adição, subtração, multiplicação e divisão são algumas das operações que podemos realizar com a calculadora. O micro:bit pode ser programado para isso.



# Montando uma calculadora

## Passo a passo:

1. Na aba **Variáveis**, crie duas variáveis, uma de nome **A** e outra **B**, como nas Figuras 25.1, 25.2, 25.3 e 25.4;

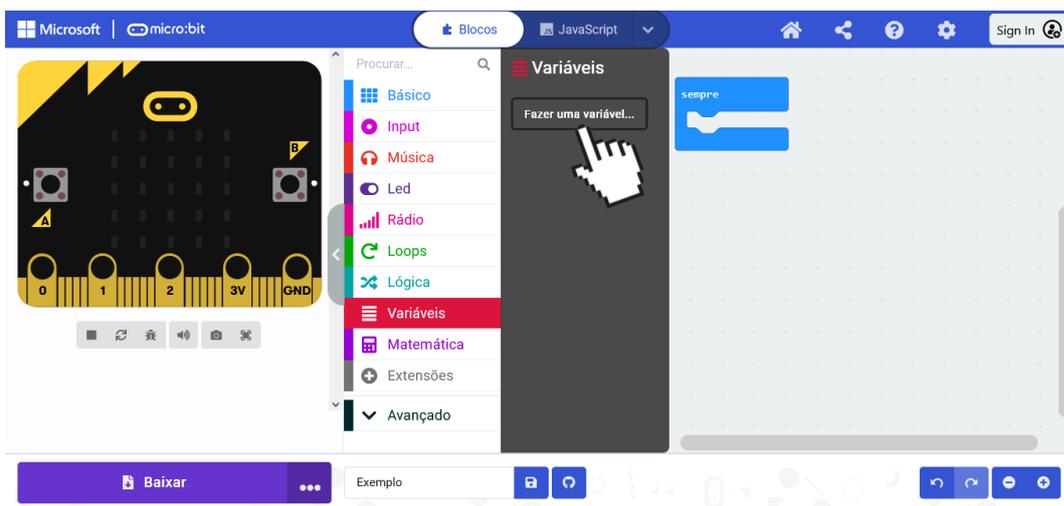


Figura 25.1: Fazendo uma variável.

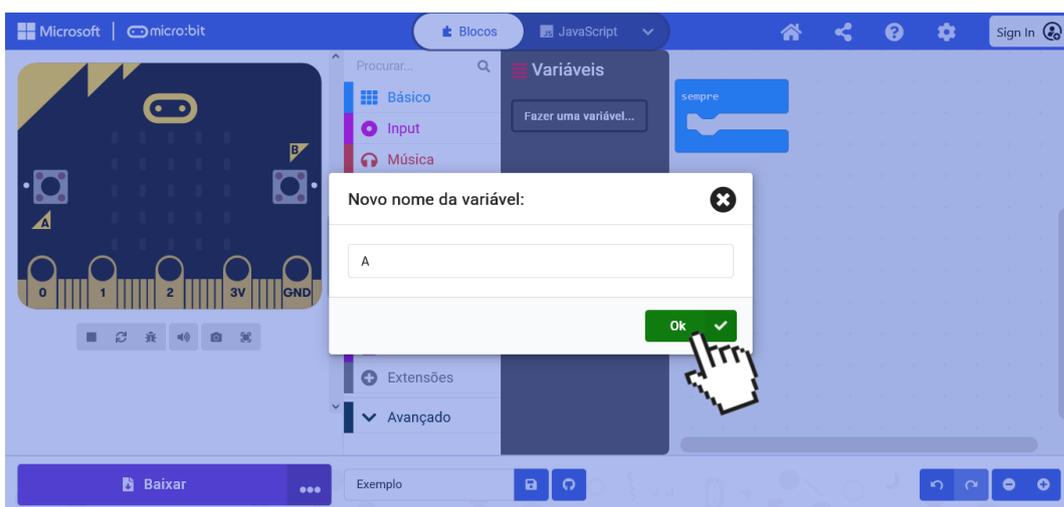


Figura 25.2: Dando nome a uma variável.

# Montando uma calculadora

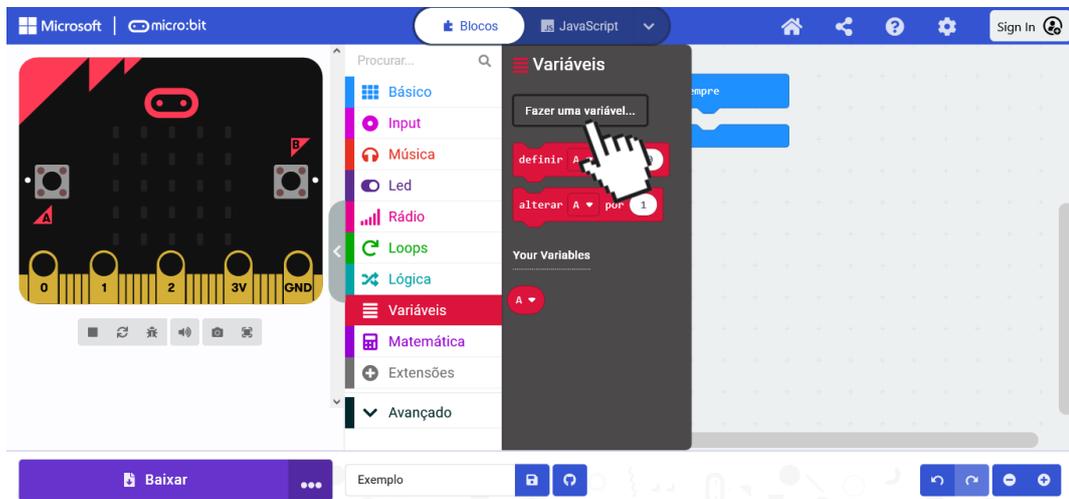


Figura 25.3: Fazendo uma variável.

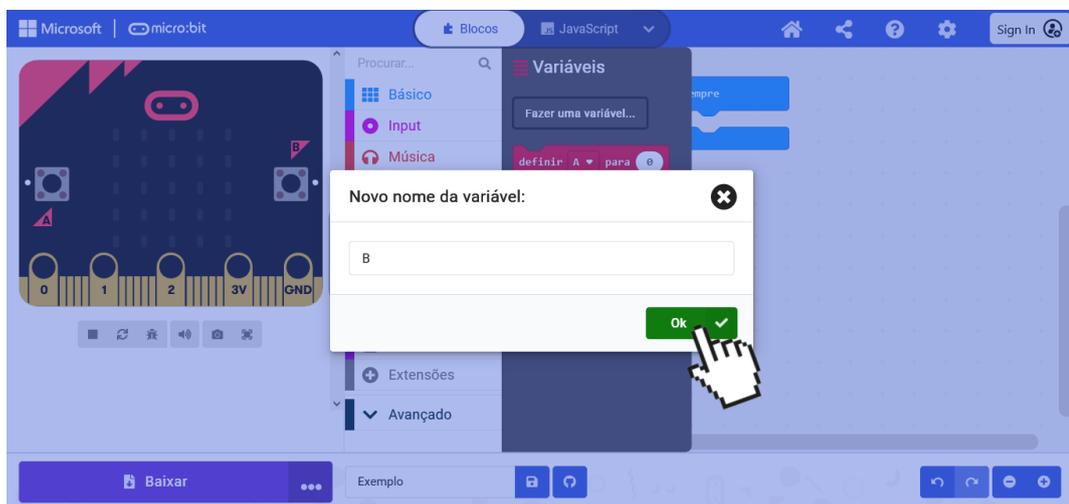


Figura 25.4: Dando nome a uma variável.



# Montando uma calculadora

- Novamente na aba **Variáveis**, selecione duas vezes o comando **definir B para 0** e coloque-os dentro do bloco de acionamento **no iniciar**, Figuras 25.5 e 25.6;

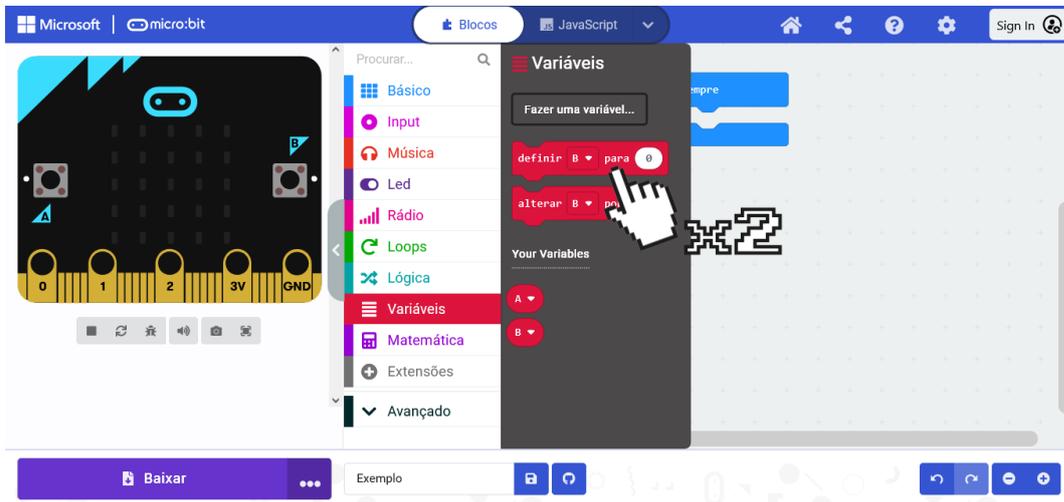


Figura 25.5: Selecionando comandos.



Figura 25.6: Montando comandos.

# Montando uma calculadora

3. Altere a letra **B** do primeiro comando para **A**, como na Figura 25.7. Quando definimos uma variável para zero no bloco **no iniciar**, toda vez que o micro:bit for iniciado os números da calculadora serão iguais a zero;



Figura 25.7: Definindo uma variável.

4. Na aba **Input**, selecione o bloco de acionamento **no botão A pressionado**, Figuras 25.8 e 25.9;

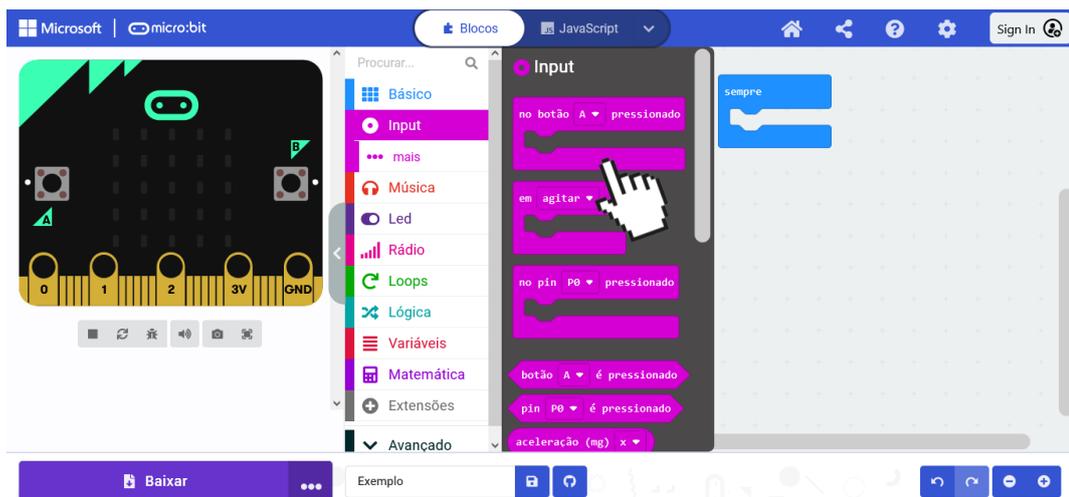


Figura 25.8: Selecionando um bloco.



# Montando uma calculadora

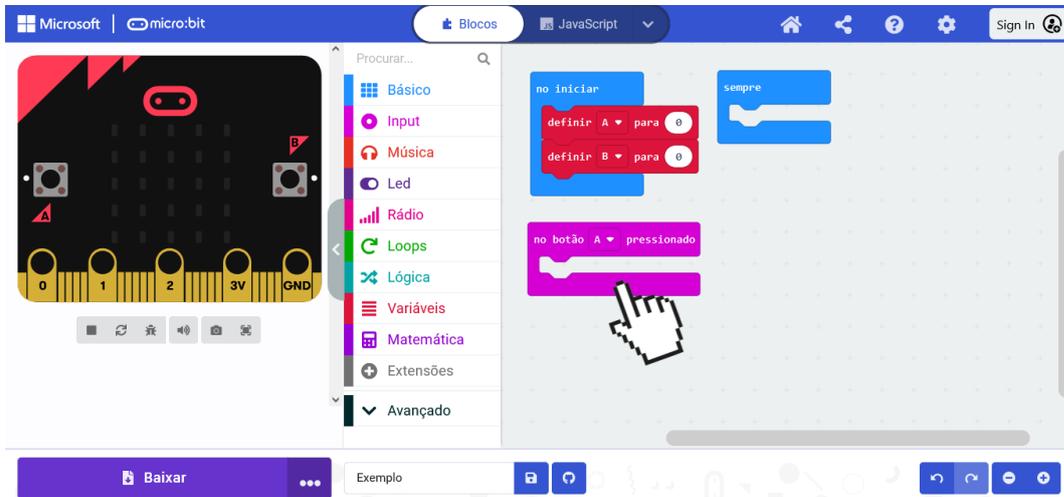


Figura 25.9: Montando um bloco.

5. Selecione o comando **alterar B por 1**, da aba **Variáveis**, e coloque-o dentro do bloco **no botão A pressionado**, Figuras 25.10 e 25.11;

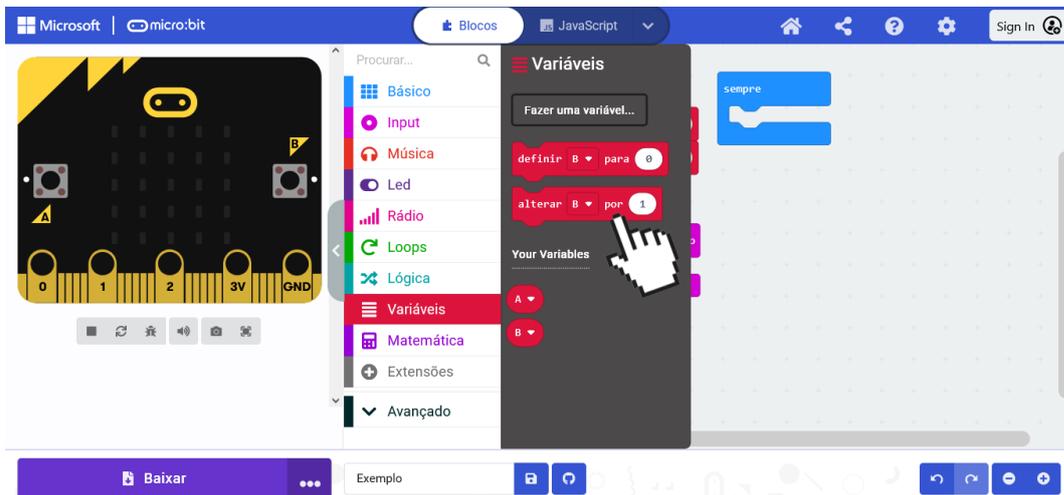


Figura 25.10: Selecionando um comando.

# Montando uma calculadora



Figura 25.11: Montando um comando.

6. Altere a variável **B** para **A**, como na Figura 25.12;



Figura 25.12: Alterando uma variável.



# Montando uma calculadora

7. Na aba **Básico**, selecione o comando **mostrar número 0** e coloque-o abaixo, de **alterar A por 1**, dentro do bloco de acionamento **no botão A pressionado**, Figuras 25.13 e 25.14;

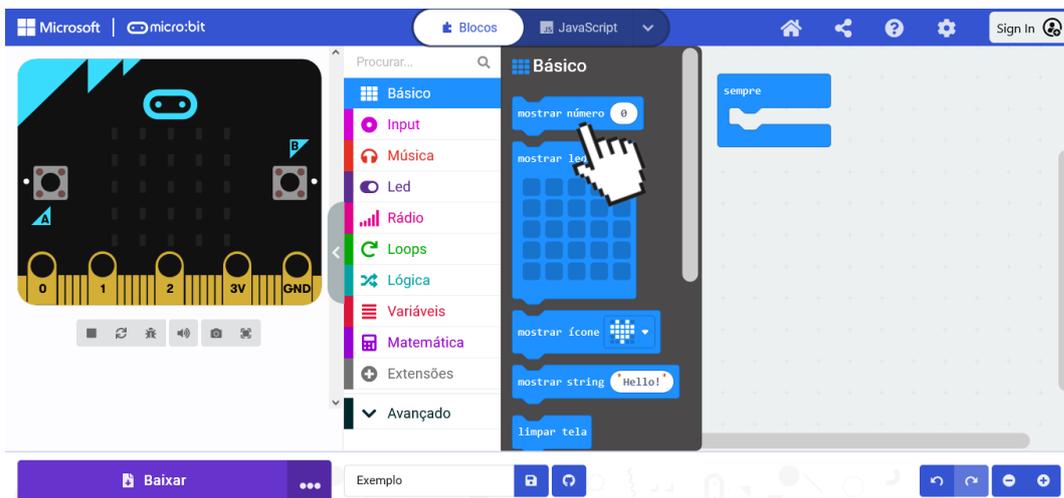


Figura 25.13: Selecionando um comando.

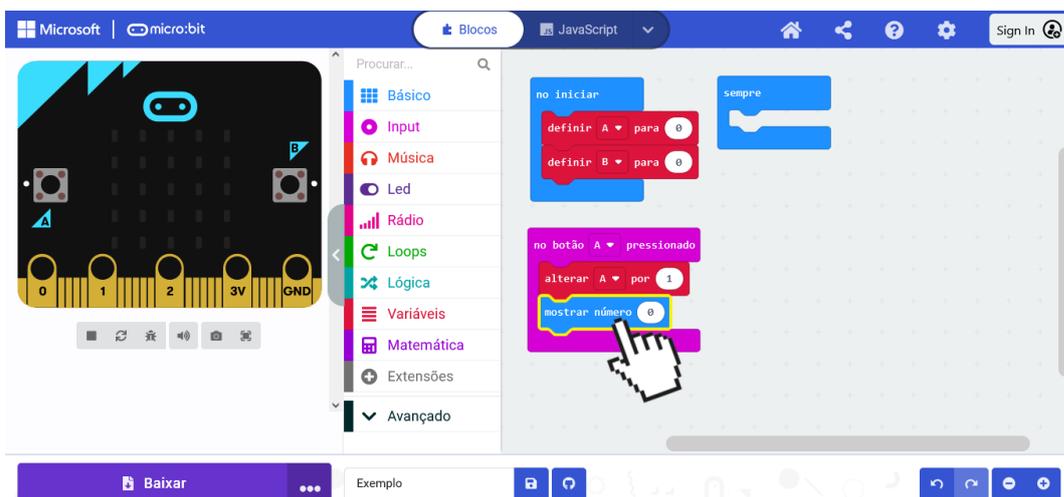


Figura 25.14: Montando um comando.

# Montando uma calculadora

8. Na aba **Variáveis**, selecione a variável **A** e coloque-a no lugar do número **0** no comando **mostrar número 0**, Figuras 25.15e 25.16;



Figura 25.15: Selecionando uma variável.



Figura 25.16: Montando um comando.



# Montando uma calculadora

9. Na aba **Básico**, selecione uma **pausa (ms) 100** e coloque-a abaixo do comando **mostrar número A**, Figuras 25.17 e 25.18;

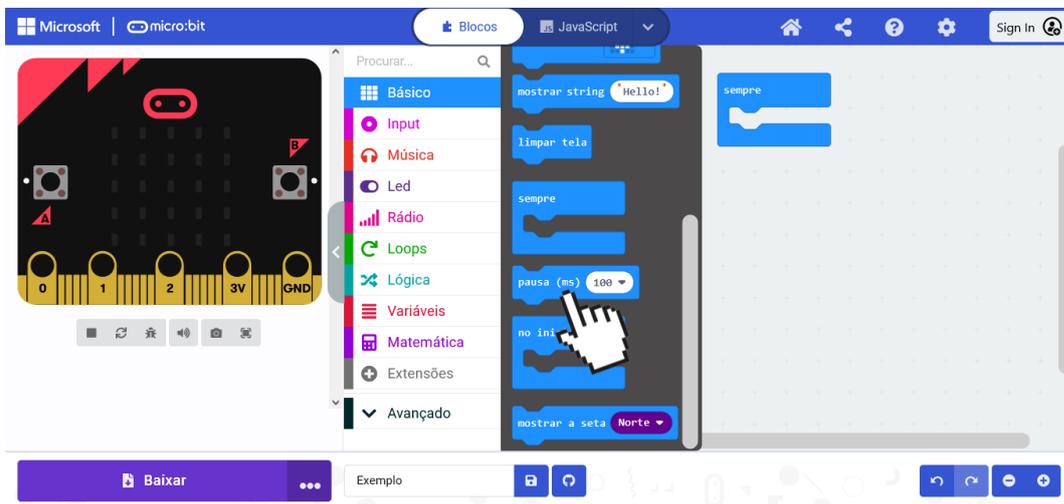


Figura 25.17: Selecionando uma pausa.



Figura 25.18: Montando um comando de pausa.

# Montando uma calculadora

**10.** Altere o tempo da **pausa** de **100 ms** para **500 ms**, Figura 25.19;



Figura 25.19: Alterando o tempo de uma pausa.

**11.** Selecione o comando **limpar tela** e coloque-o abaixo da **pausa** dentro do bloco **no botão A pressionado**, Figuras 25.20 e 25.21;

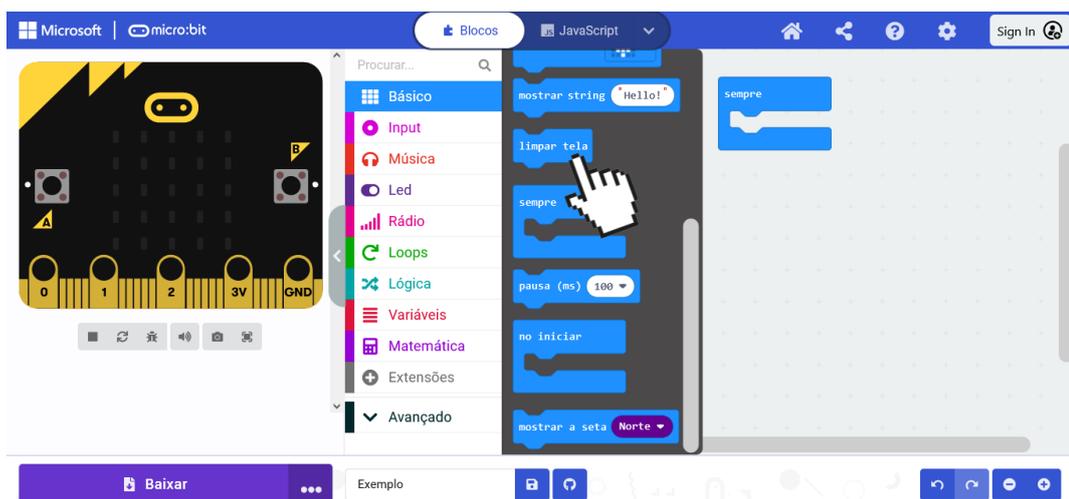
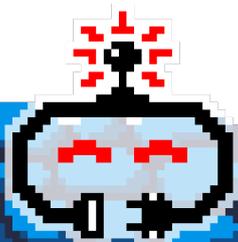


Figura 25.20: Selecionando um comando.



# Montando uma calculadora

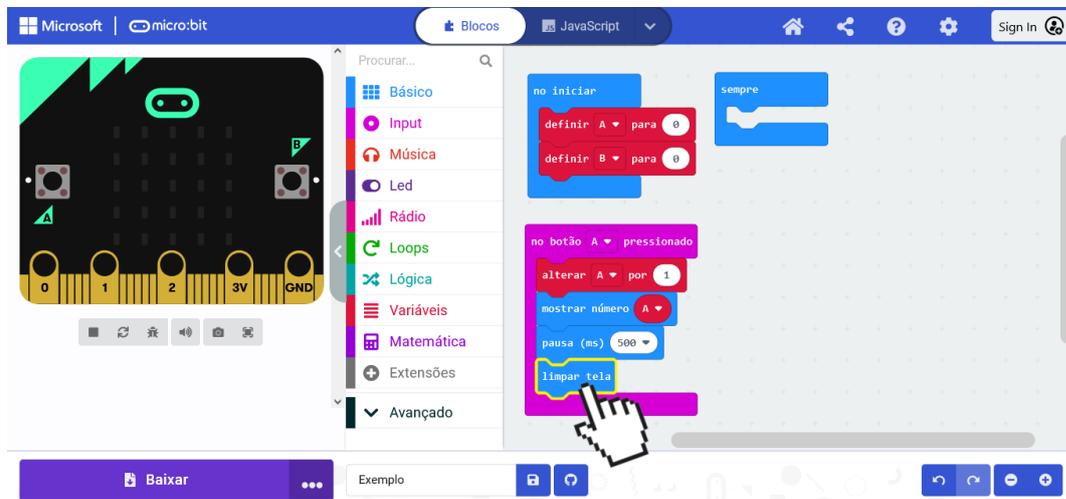


Figura 25.21: Montando um comando.

12. Repita os passos de 4 a 11 para montar o comando **no botão B pressionado** mas, desta vez, selecione a variável **B** em vez da **A**. Veja como deve ficar na Figura 25.22;

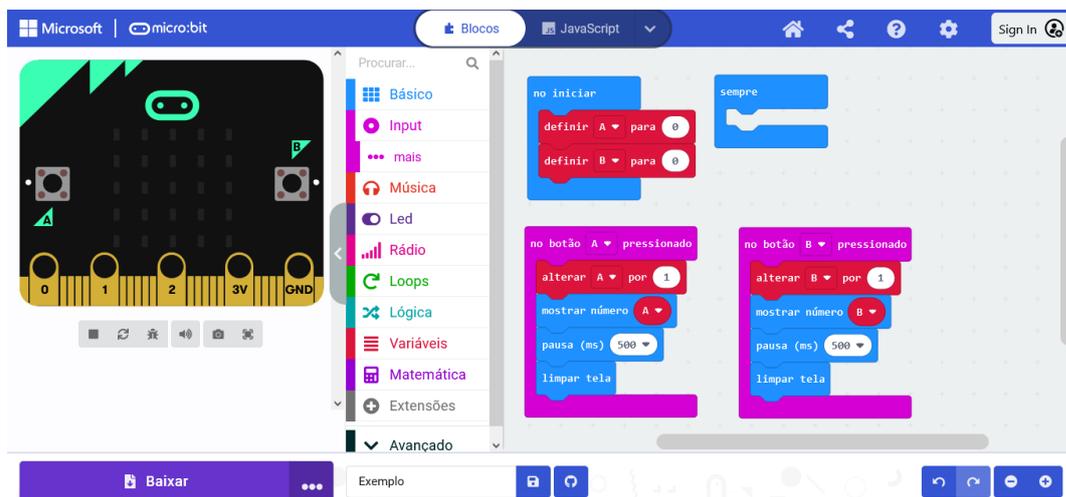


Figura 25.22: Montagem do bloco para o botão B.

# Montando uma calculadora

13. Na aba **Input**, selecione o bloco **no botão A pressionado** e altere o botão **A** para **A+B**, Figuras 25.23 e 25.24;



Figura 25.23: Selecionando um bloco.

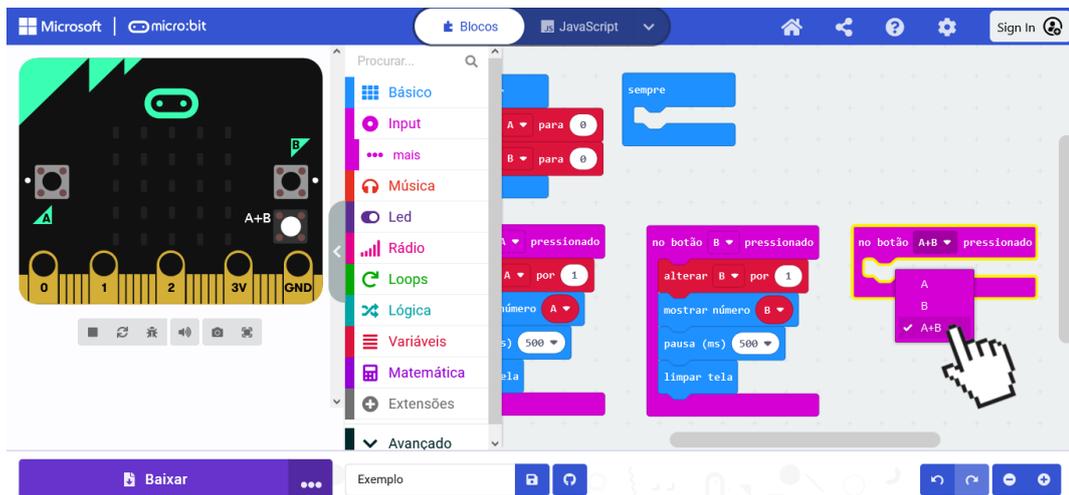


Figura 25.24: Alterando um botão.



# Montando uma calculadora

14. Em **Básico**, selecione o comando **mostrar número 0** e insira-o no bloco **no botão A+B pressionado**, Figuras 25.25 e 25.26;

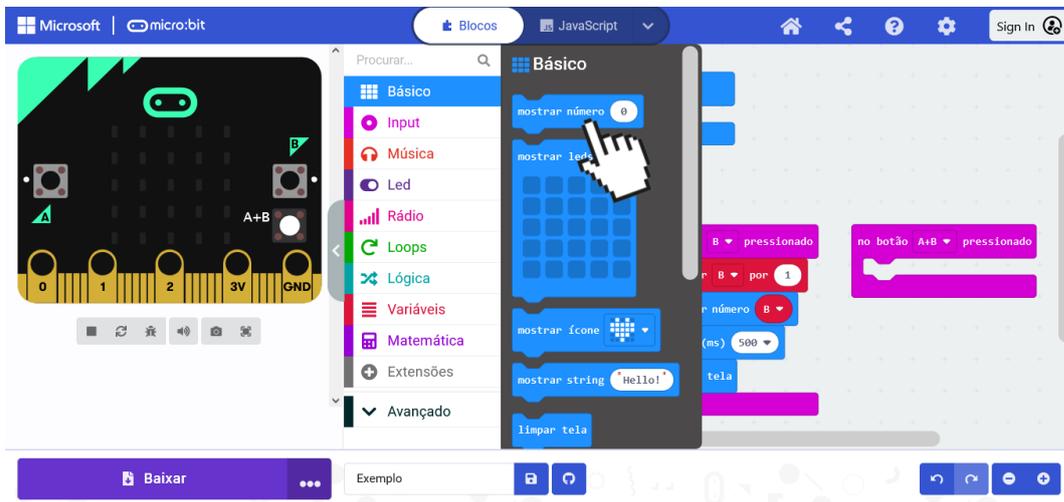


Figura 25.25: Selecionando um comando.

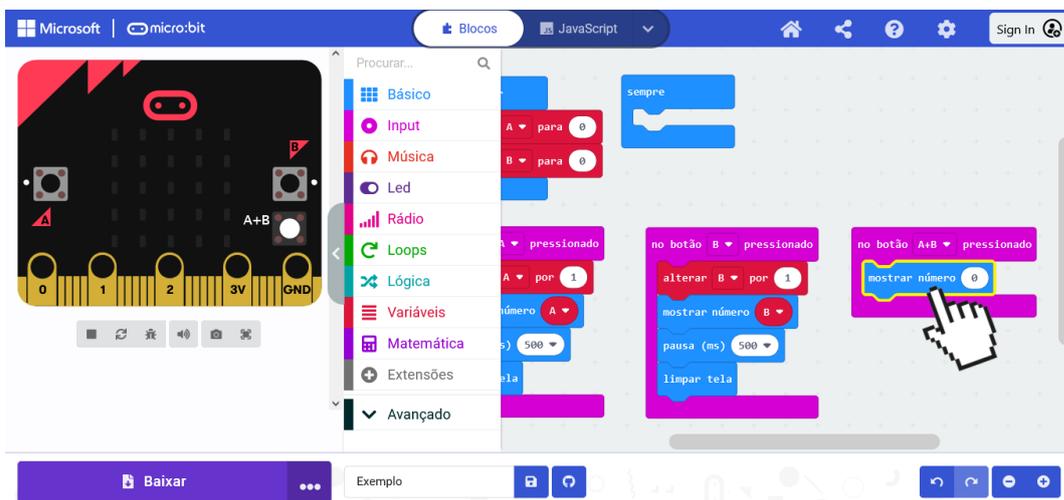


Figura 25.26: Montando um comando.

# Montando uma calculadora

15. Na aba **Matemática**, selecione a operação **0 + 0** e coloque-a no lugar do número **0** em **mostrar número 0**, Figuras 25.27 e 25.28;



Figura 25.27: Selecionando um comando.

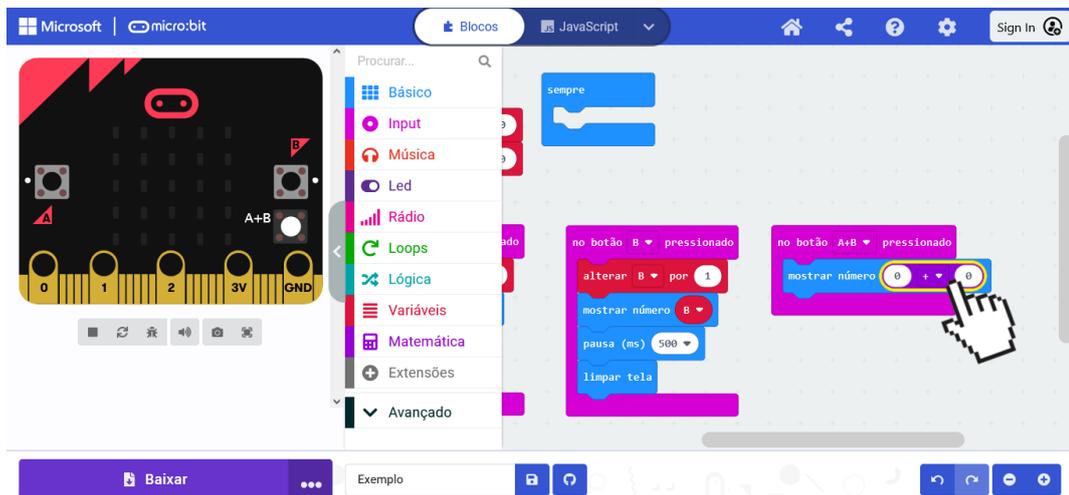


Figura 25.28: Montando um comando.



# Montando uma calculadora

16. Em **Variáveis**, selecione as variáveis **A** e **B** e coloque-as no lugar de cada **0** no comando da operação **0 + 0**, como nas Figuras 25.29, 25.30, 25.31 e 25.32;

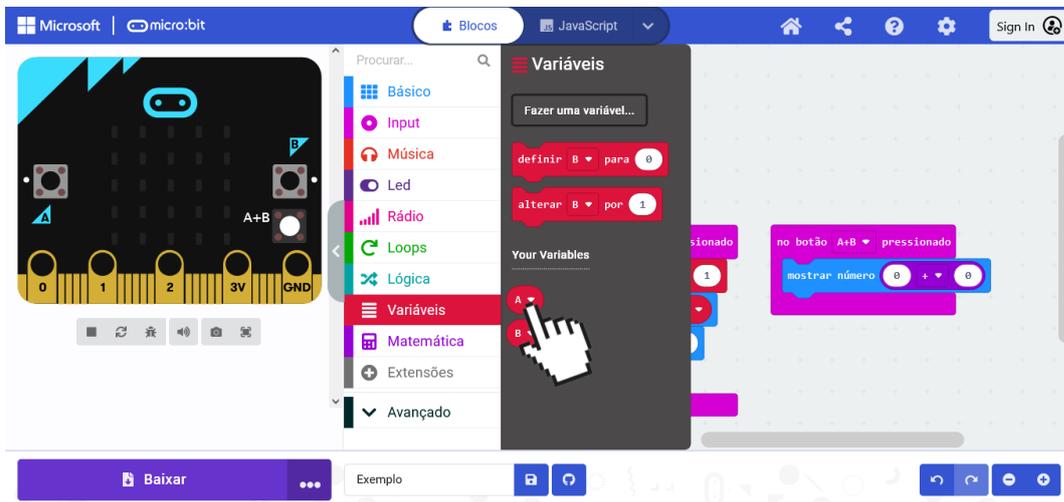


Figura 25.29: Selecionando variáveis.

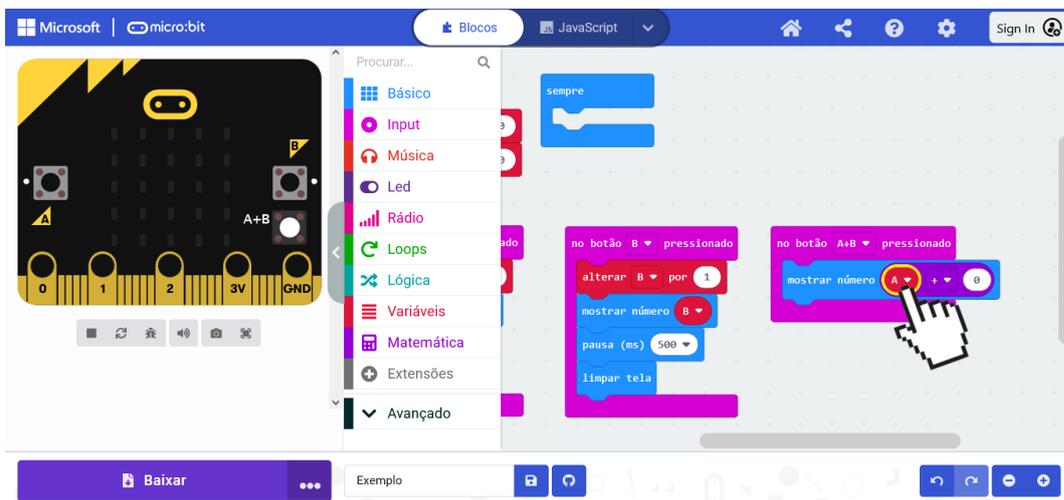


Figura 25.30: Montando um comando com variáveis.

# Montando uma calculadora

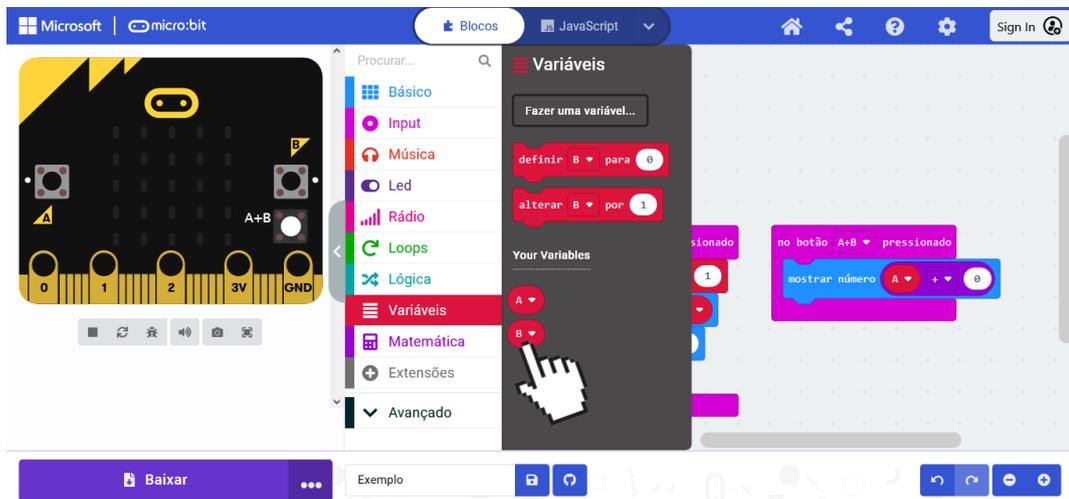
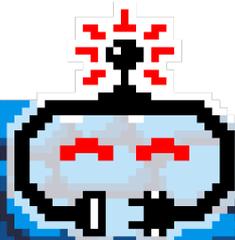


Figura 25.31: Selecionando variáveis.



Figura 25.32: Montando comando com variáveis.



# Montando uma calculadora

17. Na aba **Variáveis**, selecione dois blocos **definir B para 0** e coloque-os logo abaixo de **mostrar número A + B**, Figuras 25.33 e 25.34;

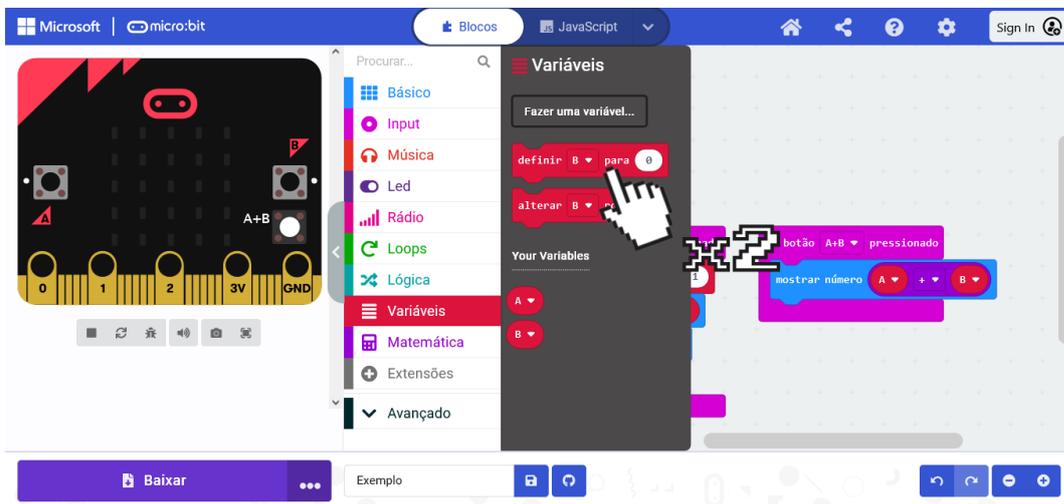


Figura 25.33: Selecionando comandos.

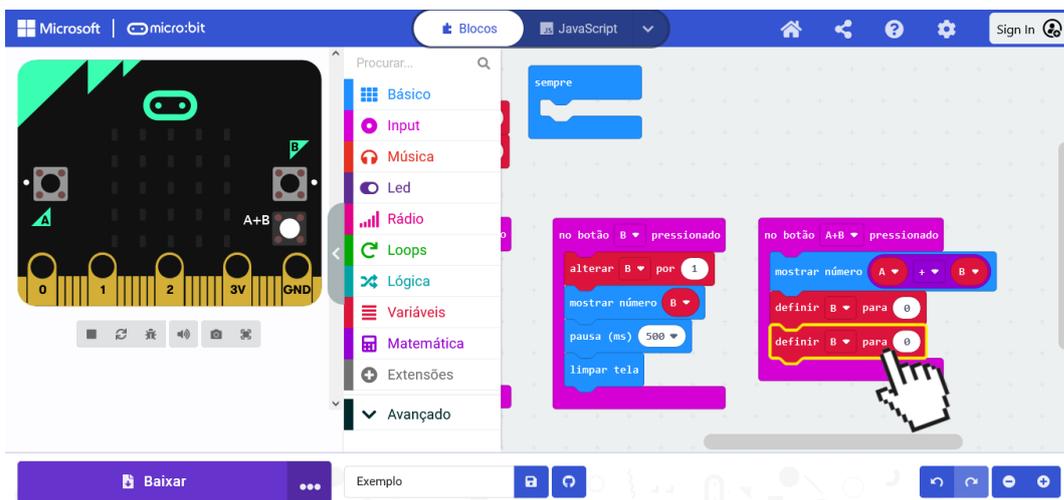


Figura 25.34: Montando comandos.

# Montando uma calculadora

**18.** Altere uma das variáveis de **B** para **A**, como na Figura 25.35.

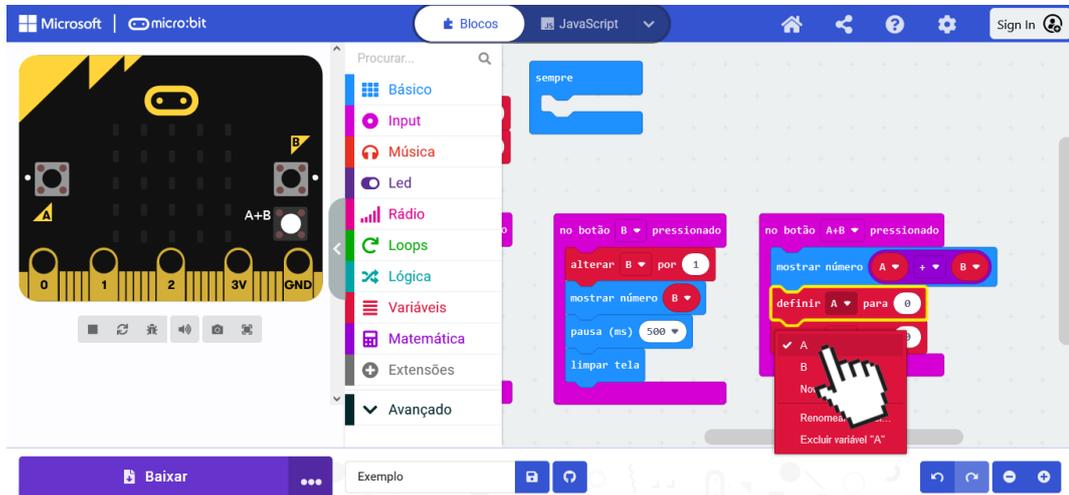


Figura 25.35: Alterando uma variável.

E a calculadora está pronta para ser usada.

Para escolher quais números quer somar, basta apertar os botões **A** e **B** até aparecer nos LEDs do micro:bit os números desejados. Toda vez que apertar **A+B**, obtêm-se o resultado da soma.

## MICRO:DICA

Para as demais operações matemáticas, a montagem do código é o mesmo. Basta que no **passo 15**, em vez de selecionar a adição  $0 + 0$ , escolha a operação  $0 - 0$  para subtração,  $0 \times 0$  para multiplicação e  $0 / 0$  para divisão.



[Clique aqui para acessar o código.](#)

