

# ROBÓTICA

Primeiros Passos

Módulo 2



AULA 17

## Olhos Piscantes

Diretoria de Tecnologia e Inovação

**GOVERNADOR DO ESTADO DO PARANÁ**

Carlos Massa Ratinho Júnior

**SECRETÁRIO DE ESTADO DA EDUCAÇÃO**

Roni Miranda Vieira

**DIRETOR DE TECNOLOGIA E INOVAÇÃO**

Claudio Aparecido de Oliveira

**COORDENADOR DE TECNOLOGIAS EDUCACIONAIS**

Marcelo Gasparin

**Produção de Conteúdo**

Andrea da Silva Castagini Padilha

Cleiton Rosa

**Revisão Textual**

Orlando de Macedo Junior

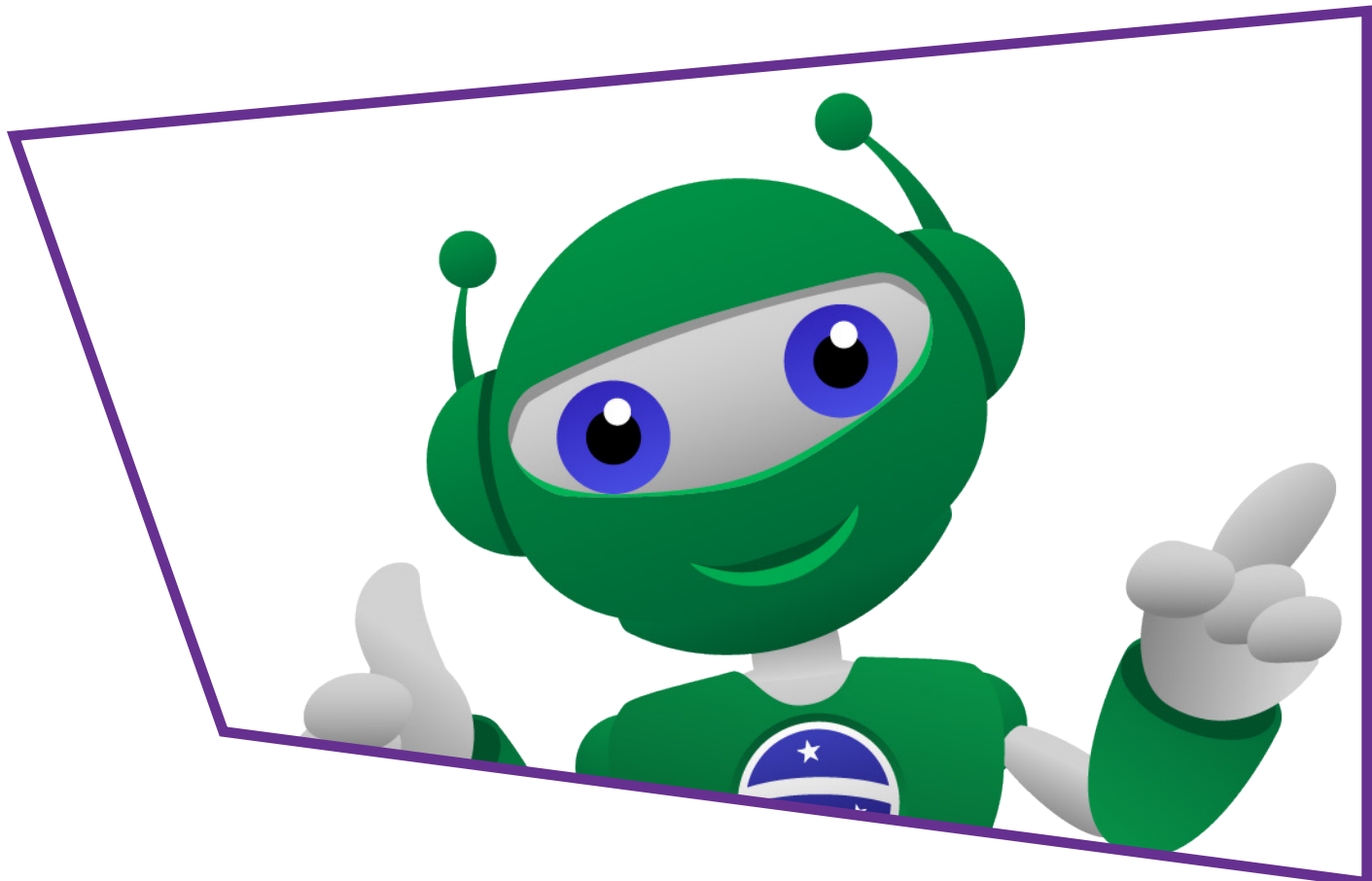
**Projeto Gráfico e Diagramação**

Edna do Rocio Becker

2023

# SUMÁRIO

Introdução	2
Objetivos desta aula	2
Competências gerais previstas na BNCC	3
Habilidades do século XXI a serem desenvolvidas	4
Roteiro da aula	5
1. Contextualização	5
2. Conteúdo	6
3. Feedback	32
Referências	32





## Introdução

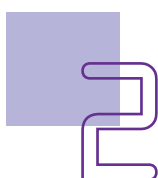
Já ouviu a frase “Os olhos são o espelho da alma”? Ou então, “É melhor tapar os olhos?”. Os olhos não são apenas os órgãos de sentido que apreendem visualmente tudo que está ao seu redor, mas também se expressam. Podemos sinalizar diferentes emoções nos olhos, dor, atenção, fuga, medo, sedução etc.

A isso, chama-se de linguagem não verbal. Nesta aula, você vai aprender a programar a matriz de LEDs 8x8 para simular exemplos de linguagem não verbal expressa pelos olhos. Assim, poderá dar mais expressividade aos seus projetos robóticos! Vamos lá?



## Objetivos desta aula

- Promover a movimentação de caracterizações com o uso de uma matriz de LEDs 8x8;
- Utilizar a programação para trocar alterações de matriz de LEDs 8x8, com o movimento de abrir e fechar os olhos.





## Competências gerais previstas na BNCC

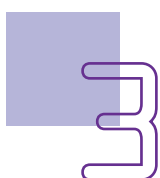
**[CG02]** - Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas.

**[CG04]** - Utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital –, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo.

**[CG05]** - Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

**[CG09]** - Exercitar a empatia, o diálogo, a resolução de conflitos e a cooperação, fazendo-se respeitar e promovendo o respeito ao outro e aos direitos humanos, com acolhimento e valorização da diversidade de indivíduos e de grupos sociais, seus saberes, identidades, culturas e potencialidades, sem preconceitos de qualquer natureza.

**[CG10]** - Agir pessoal e coletivamente com autonomia, responsabilidade, flexibilidade, resiliência e determinação, tomando decisões com base em princípios éticos, democráticos, inclusivos, sustentáveis e solidários.





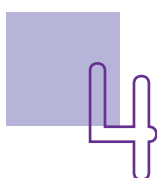
## Habilidades do século XXI a serem desenvolvidas

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



## Lista de Materiais

- 1 placa Arduino Uno R3;
- 1 cabo USB;
- 1 placa protoboard;
- 2 módulos matriz de LEDs 8x8;
- 5 jumpers macho-macho;
- 5 jumpers macho-fêmea;
- Notebook;
- Software Arduino IDE.





## Roteiro da Aula

### 1. Contextualização

Um dos quesitos que dá a sensação de humanidade aos robôs é a possibilidade de expressão não verbal, e entre elas, a presença de olhos e sua animação. Segundo uma canção do cantor e compositor Chico Buarque, que começa assim: “Olhos nos olhos, quero ver o que você diz”.

Então, nesta aula, você vai utilizar componentes do seu kit de robótica para programar diferentes expressões do olhar do seu robô. Para isso, utilizará duas matrizes de LED 8x8.



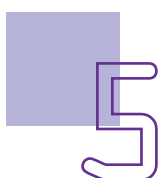
### Para saber mais

Para se inspirar, que tal assistir a esse trecho da animação “Wall-E” (2008), na qual a personagem E.V.E. se expressa alterando a imagem dos seus olhos de robô.

Wall-E, as expressões não verbais dos robôs (E.V.E.)



Disponível em: <https://youtu.be/Cnv-iFhdlqE> . Acesso em: 26 jan. de 2023.



## 2. Conteúdo

Nesta aula, utilizaremos o módulo matriz de LEDs 8x8. Como já visto na aula anterior, este componente é um conjunto de LEDs organizado em uma matriz de 8 colunas e 8 linhas, ou seja, 64 LEDs no total. Ele é controlado por um circuito integrado chamado MAX7219 ou MAX7221, que permite controlar individualmente cada LED por comandos enviados por um microcontrolador, como o Arduino.

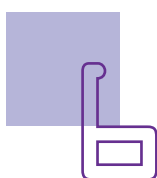
Com esse componente, você pode criar exibição de informações, como letras, números, emojis, entre outros. Aqui, vamos utilizar dois componentes para simular o movimento de olhos em um robô.

A programação é realizada com a ajuda de bibliotecas prontas do módulo matriz de LEDs 8x8.

### Curiosidade

Por que piscamos?

Você sabia que piscamos de 15 a 20 vezes por minuto? É verdade, e isso, sem pensar, porque o ato de piscar é involuntário, da mesma forma que a respiração. Por ser um reflexo automático do corpo, o ato de piscar ajuda a lubrificar e proteger os olhos, pois quando piscamos há a liberação de uma pequena quantidade de líquido lacrimal, que limpa e hidrata os olhos. Quando piscamos ajudamos a manter a visão clara e nítida. Além disso, piscar também pode ser uma forma de se expressar, indicando surpresa, medo, desconforto ou até indicar que está brincando (com uma piscadela).





### 3. Montagem e programação

Separe os componentes da lista de materiais. O sucesso desse projeto está na pinagem correta dos jumpers nos pontos da matriz de LEDs 8x8. Para isso, verifique o mapeamento dos pinos entre esse componente e o Arduino (quadro 1).

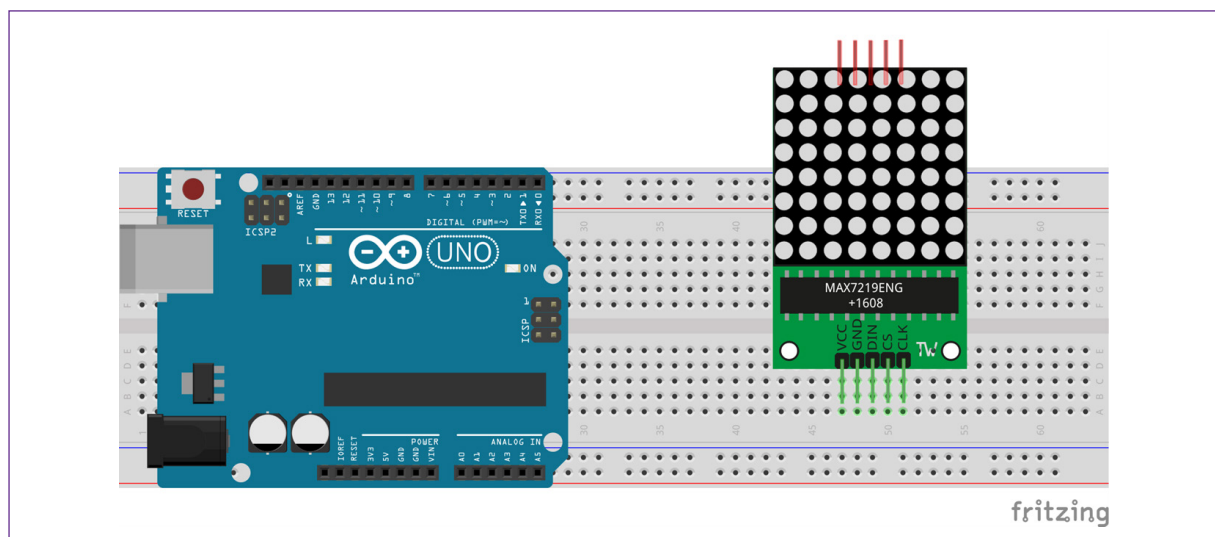


Quadro 1 - Mapeamento dos pinos na placa Arduino

LIGAÇÕES	
Arduino	Módulo
5V	VCC
GND	GND
Pino 8	DIN
Pino 9	CS
Pino 10	CLK

Vamos começar a montagem encaixando a matriz de LEDs na protoboard (figura 1).

Figura 1 - Encaixe da matriz de LEDs na protoboard



Fonte: Fritzing, 2023.

Encaixe a segunda matriz de LEDs na protoboard (figura 2).

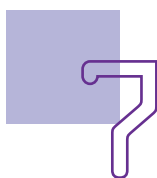
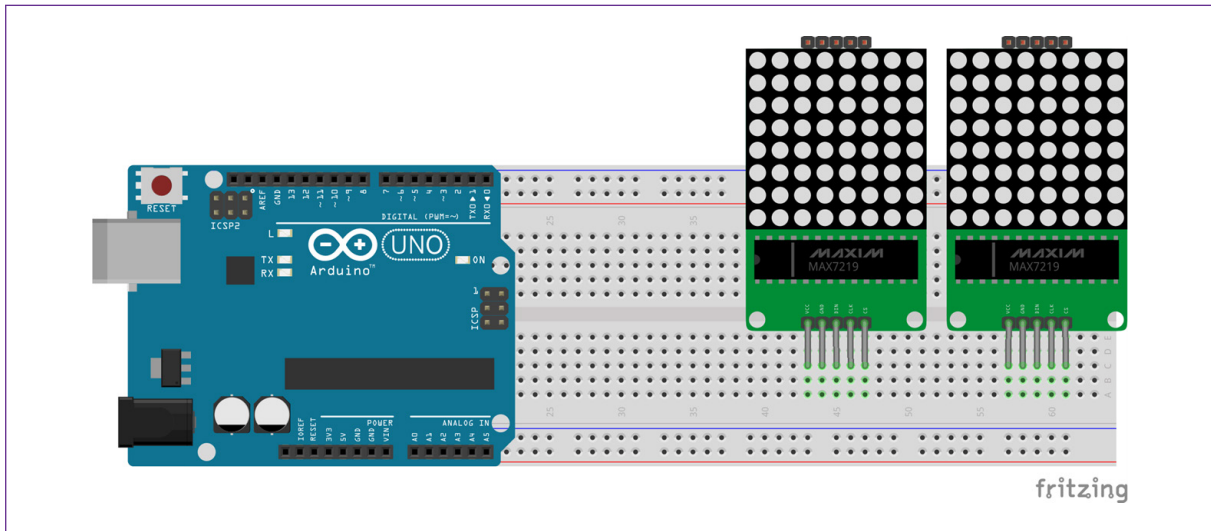


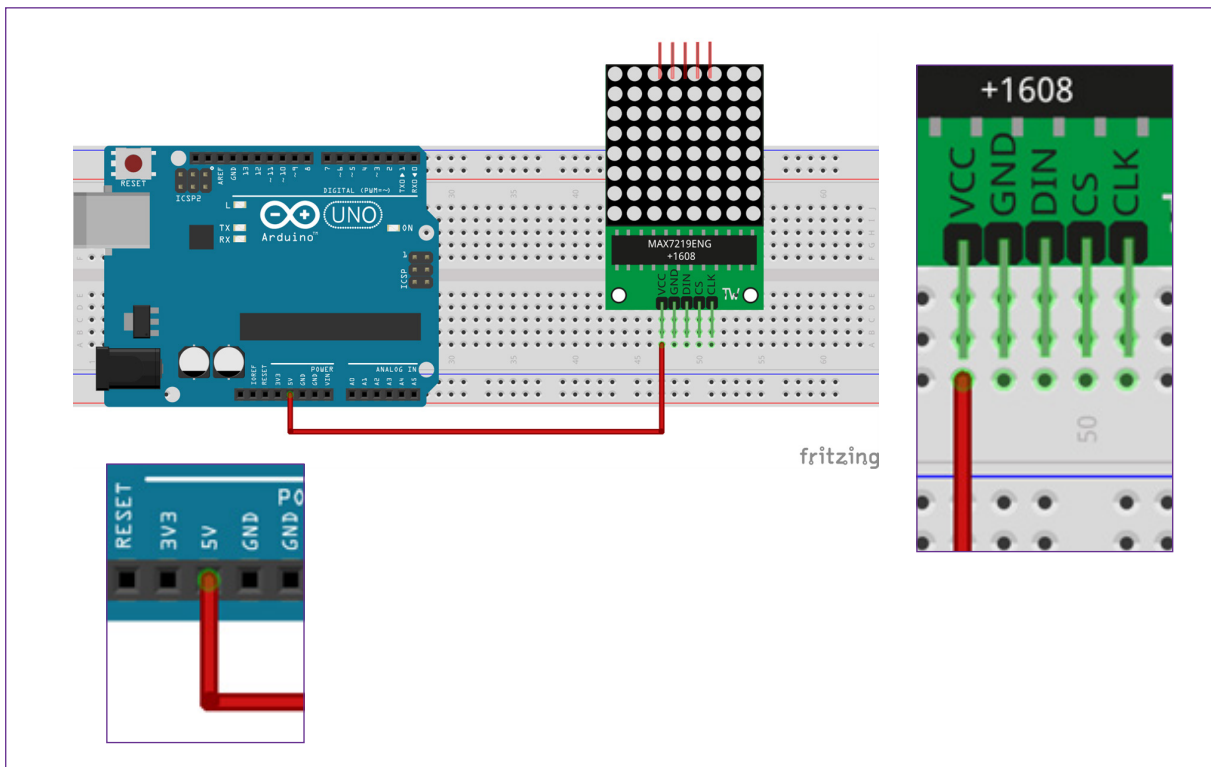
Figura 2 - Encaixe da segunda matriz de LEDs na protoboard



Fonte: Fritzing, 2023.

Conecte com um jumper, o **pino 5V** do Arduino com o **pino VCC** na matriz de LEDs.

Figura 3 - Conexão do pino 5V com o pino VCC na matriz de LEDs

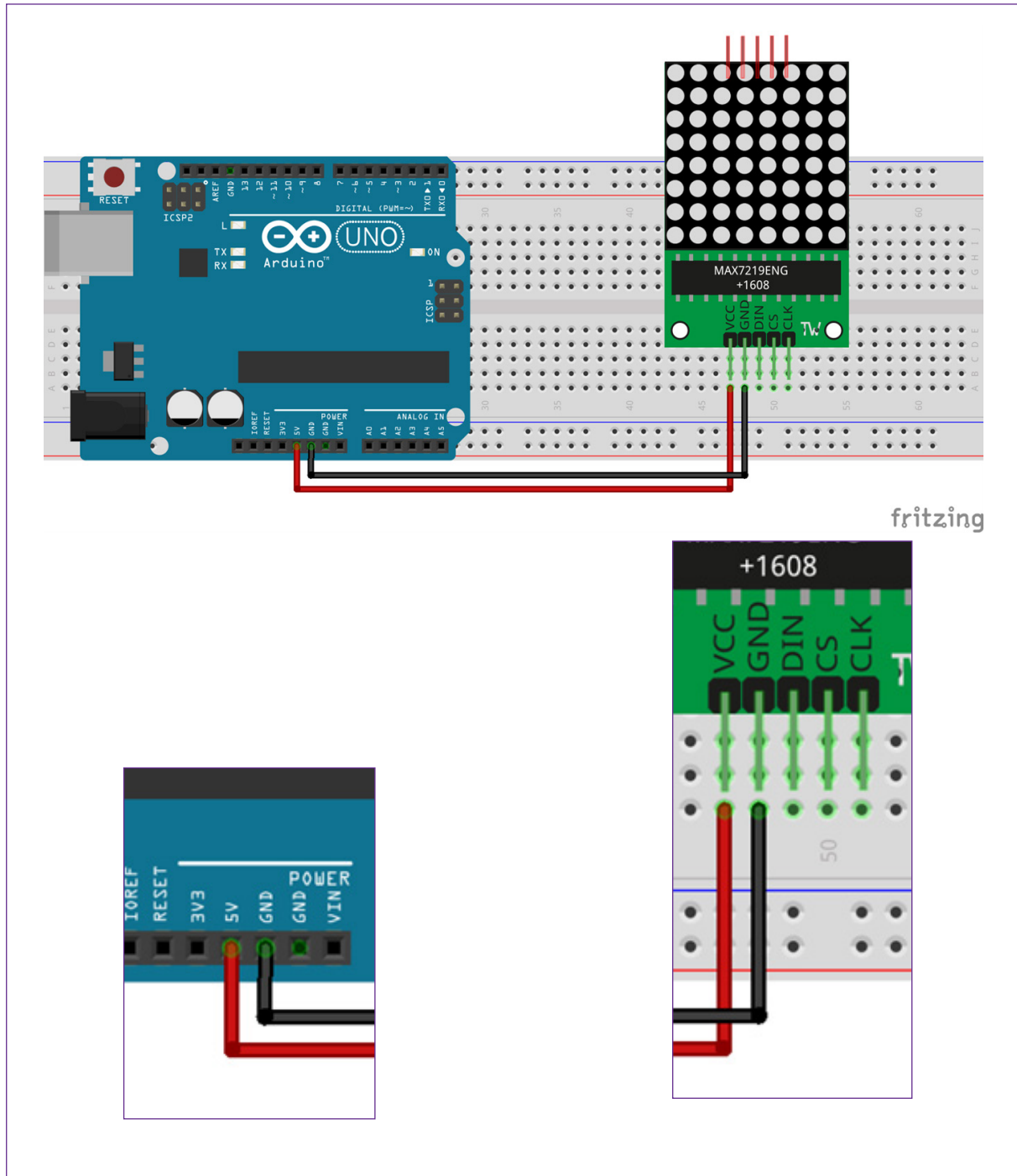


Fonte: Fritzing, 2023.



Conecte o segundo jumper, interligando o **pino GND** do Arduino com o **pino GND** na matriz de LEDs (figura 4).

Figura 4 - Conexão do pino GND do Arduino ao pino GND da matriz de LEDs

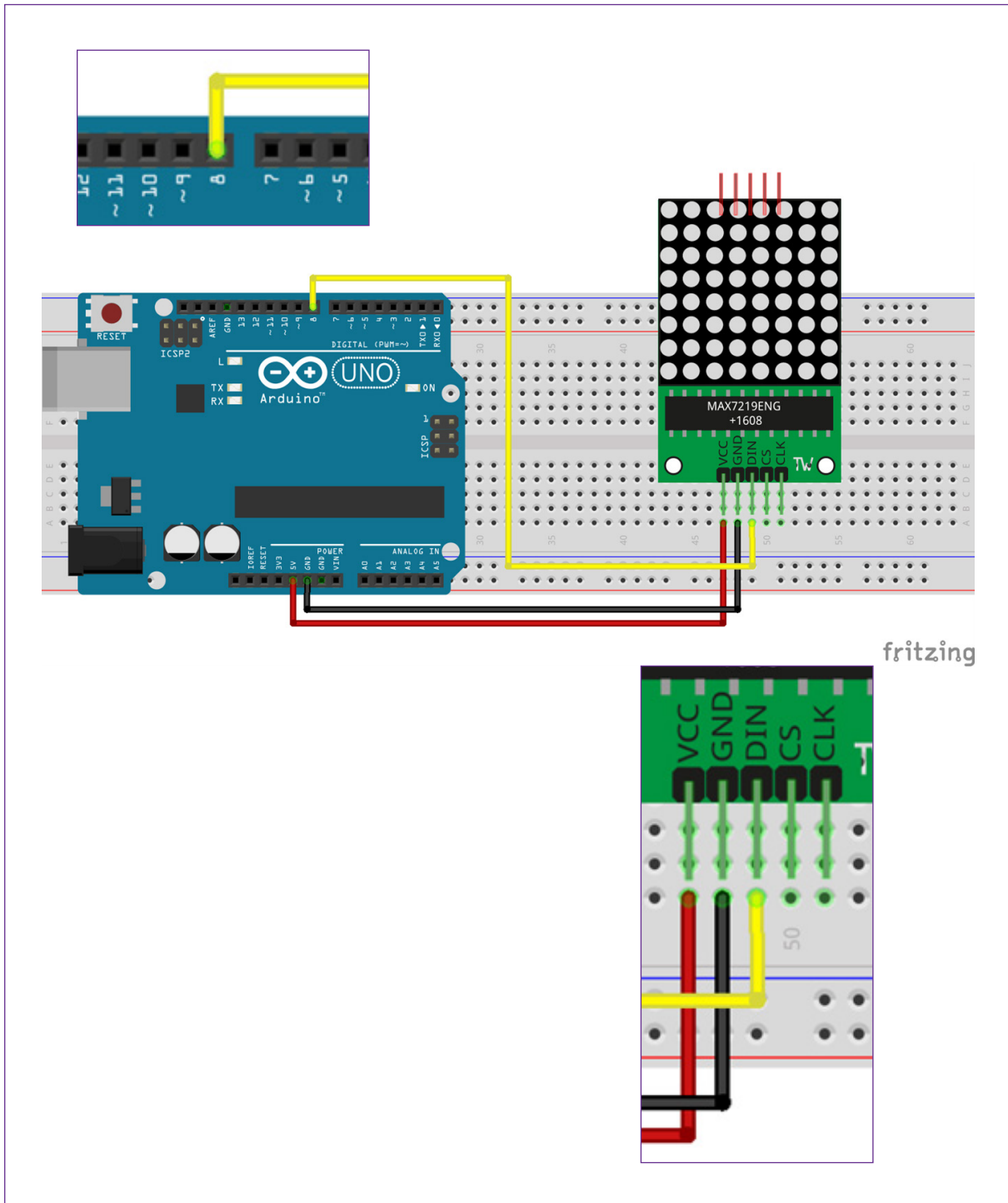


Fonte: Fritzing, 2023.



Utilizando outro Jumper, interligue o **pino 8** do Arduino com o **pino DIN** da matriz de LEDs, jumper amarelo (figura 5).

Figura 5 - Conexão do pino 8 do Arduino ao pino DIN da matriz de LEDs

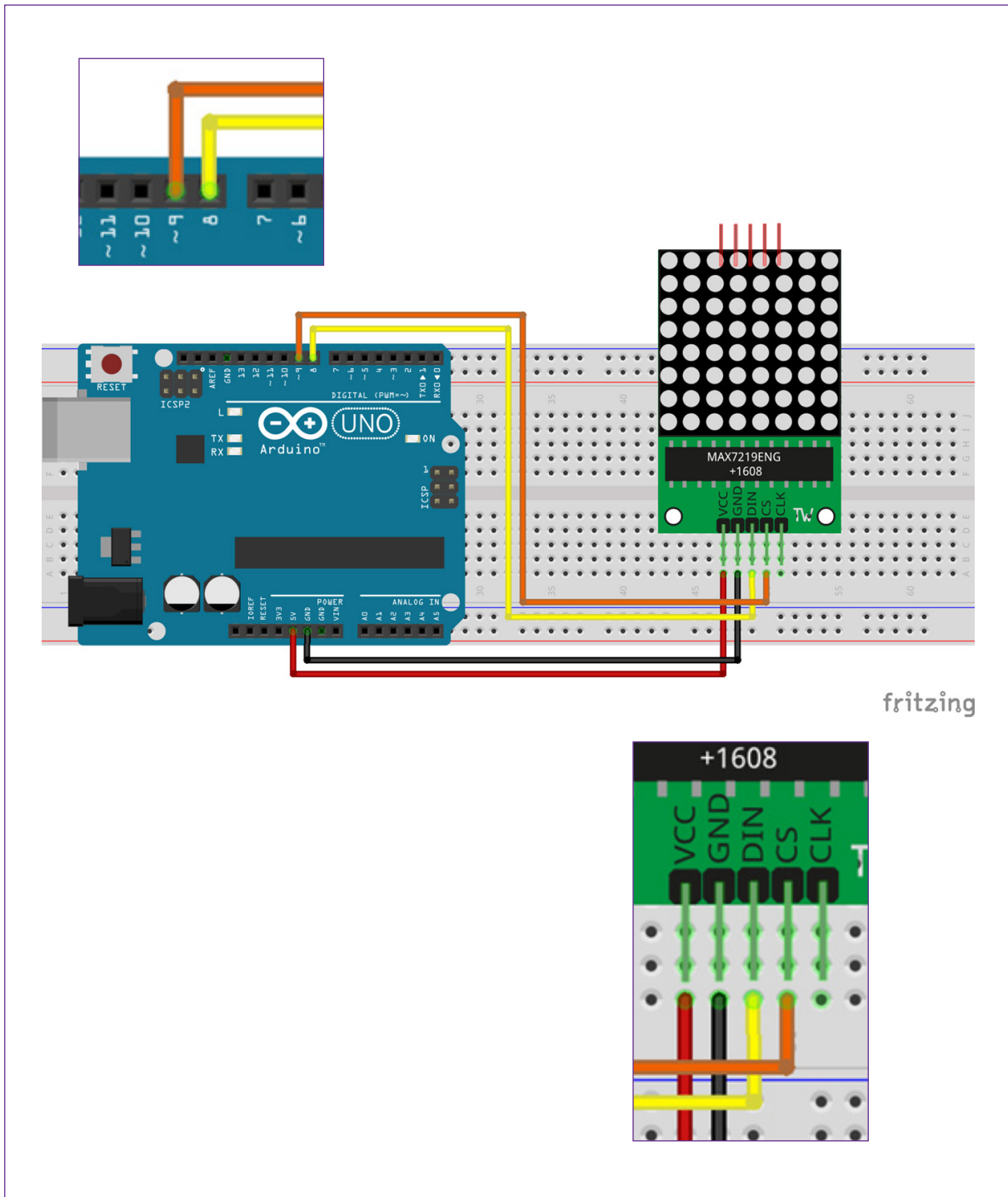


Fonte: Fritzing, 2023.

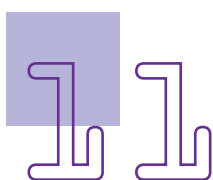


Agora, insira o quarto jumper, interligando o **pino CS** da matriz de LEDs com o **pino 9** do Arduino, jumper laranja (figura 6).

Figura 6 - Conexão do pino 9 do Arduino no pino CS da matriz de LEDs

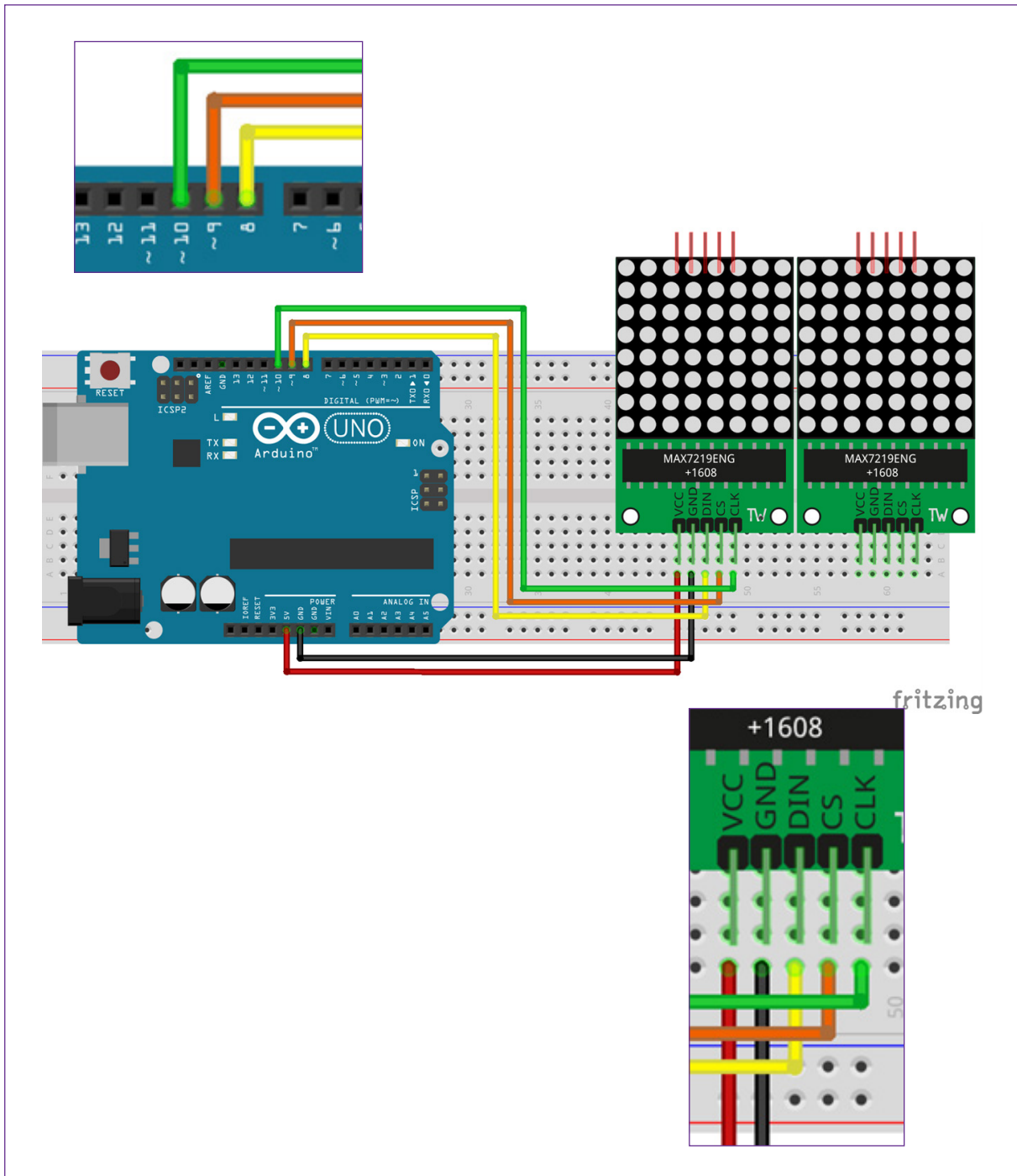


Fonte: Fritzing, 2023.



O último pino da matriz de LEDs, o **pino CLK**, deve ser interligado com um jumper no **pino 10** do Arduino, jumper verde (figura 7).

Figura 7 - Conexão do pino 10 do Arduino no pino CLK da matriz de LEDs

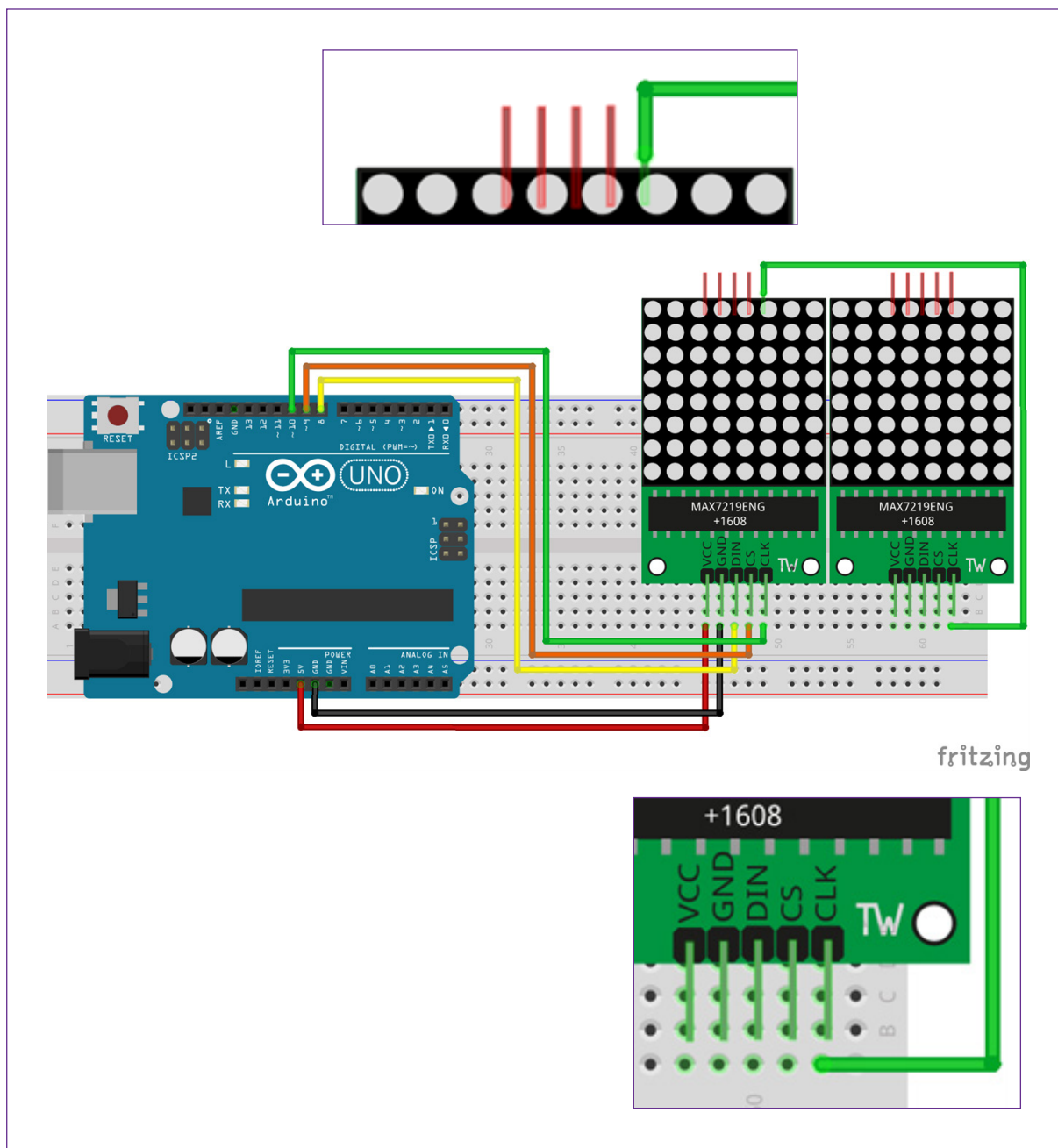


Fonte: Fritzing, 2023.



Agora, faremos a interligação entre as duas matrizes de LED, para isso você precisará de um jumper macho-fêmea. O conector fêmea do jumper, deve ser inserido no pino CLK da matriz 1 (a que está ligada ao Arduino), e o terminal macho, na protoboard, atrás da segunda matriz de LEDs, paralelo ao pino CLK (figura 8).

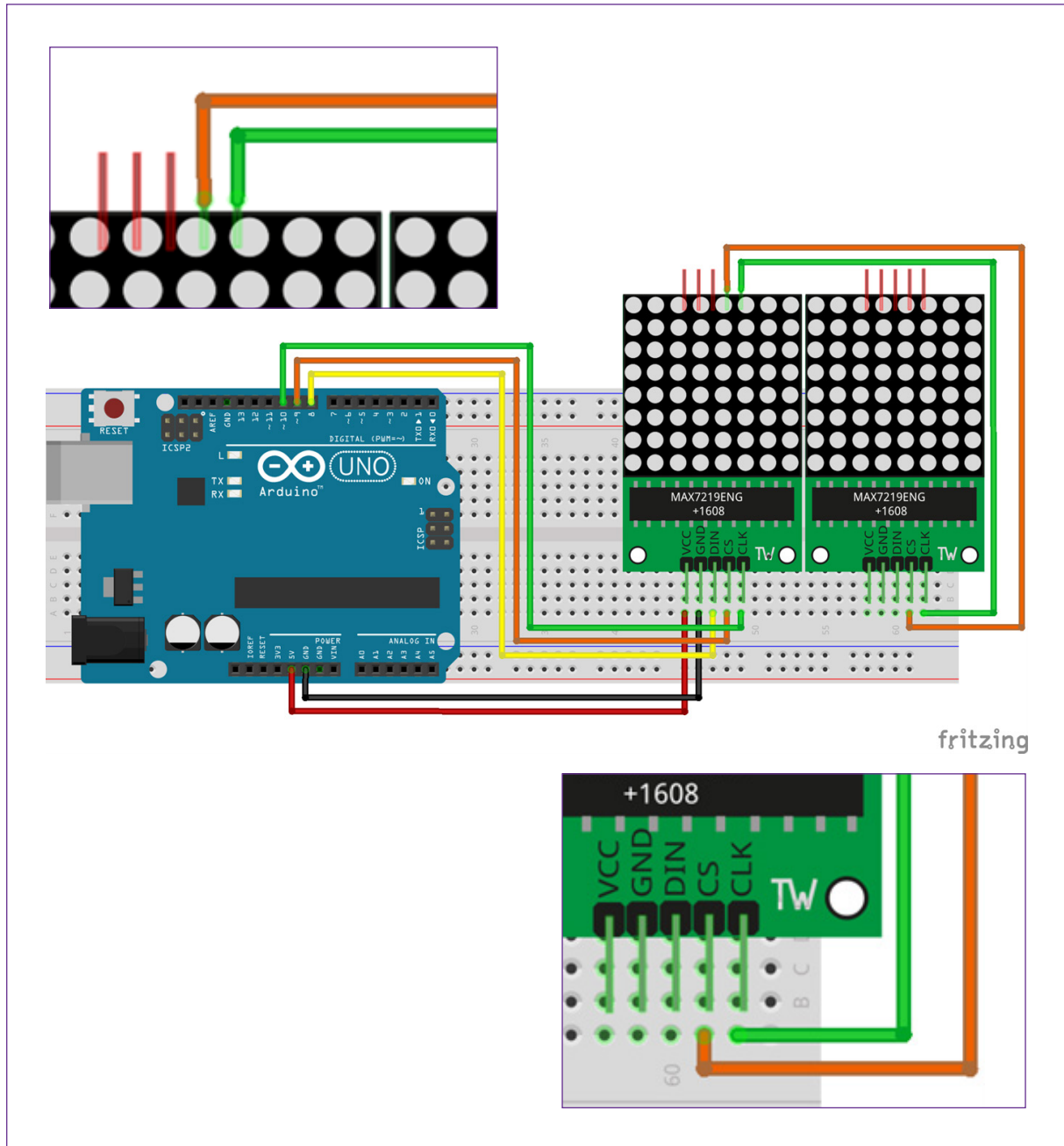
Figura 8 - Conexão do pino CLK entre as matrizes de LED



Fonte: Fritzing, 2023.

Continue a interligar as matrizes, com um segundo jumper macho-fêmea. Nesse momento, conecte o pino CS superior da primeira matriz (pelo terminal fêmea) a outra ponta (terminal macho) na protoboard, na lacuna relacionada ao pino CS da segunda matriz de LEDs (figura 9).

Figura 9 - Conexão do pino CS entre as matrizes de LED



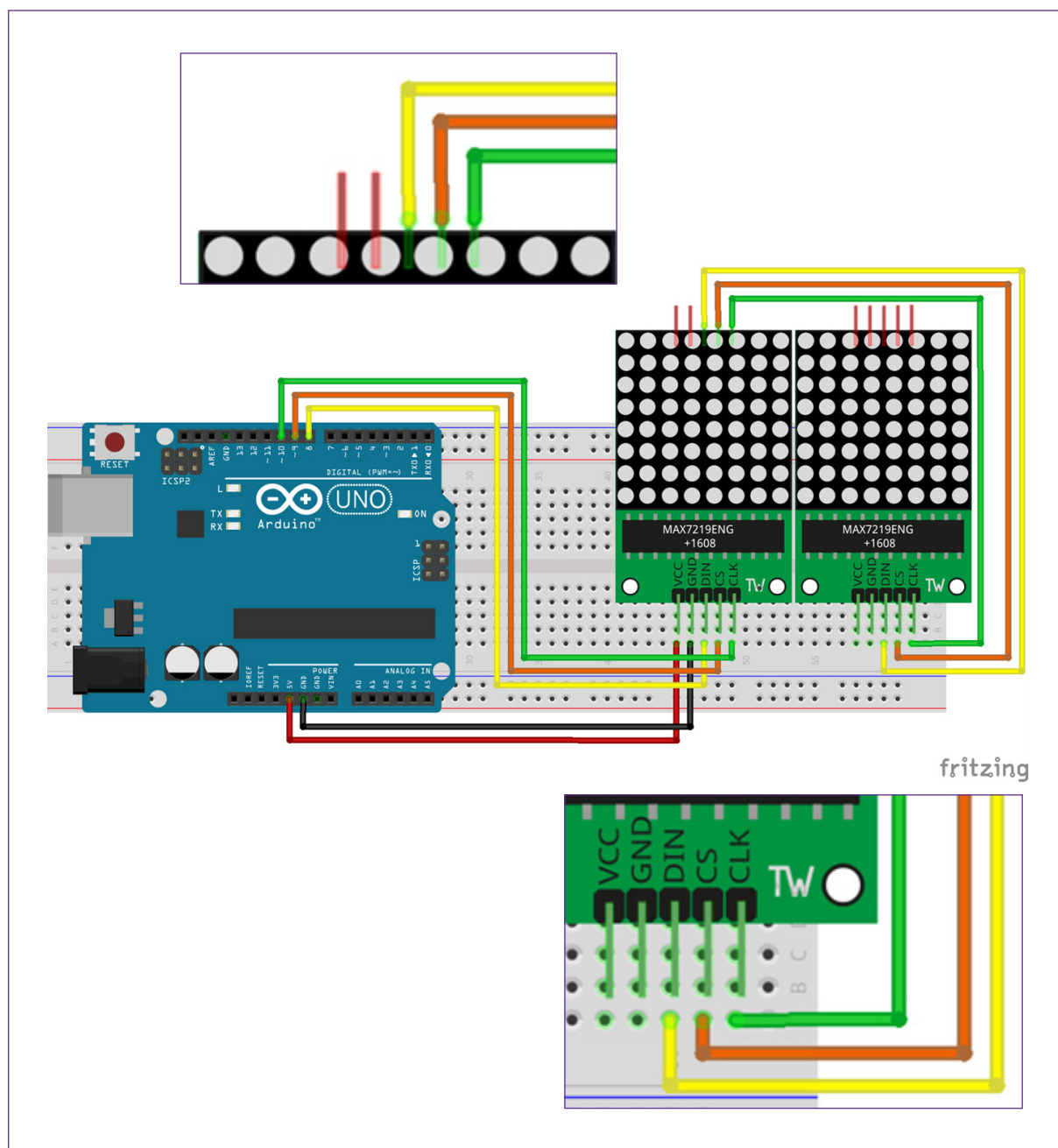
Fonte: Fritzing, 2023.





Pegue um terceiro jumper macho-fêmea para interligar os pinos DIN das matrizes de LED. Da mesma forma que a pinagem anterior, você conectará a ponta fêmea no pino DIN superior da primeira matriz de LEDs e o terminal macho na protoboard, no furo relativo ao pino DIN da segunda matriz de LEDs.

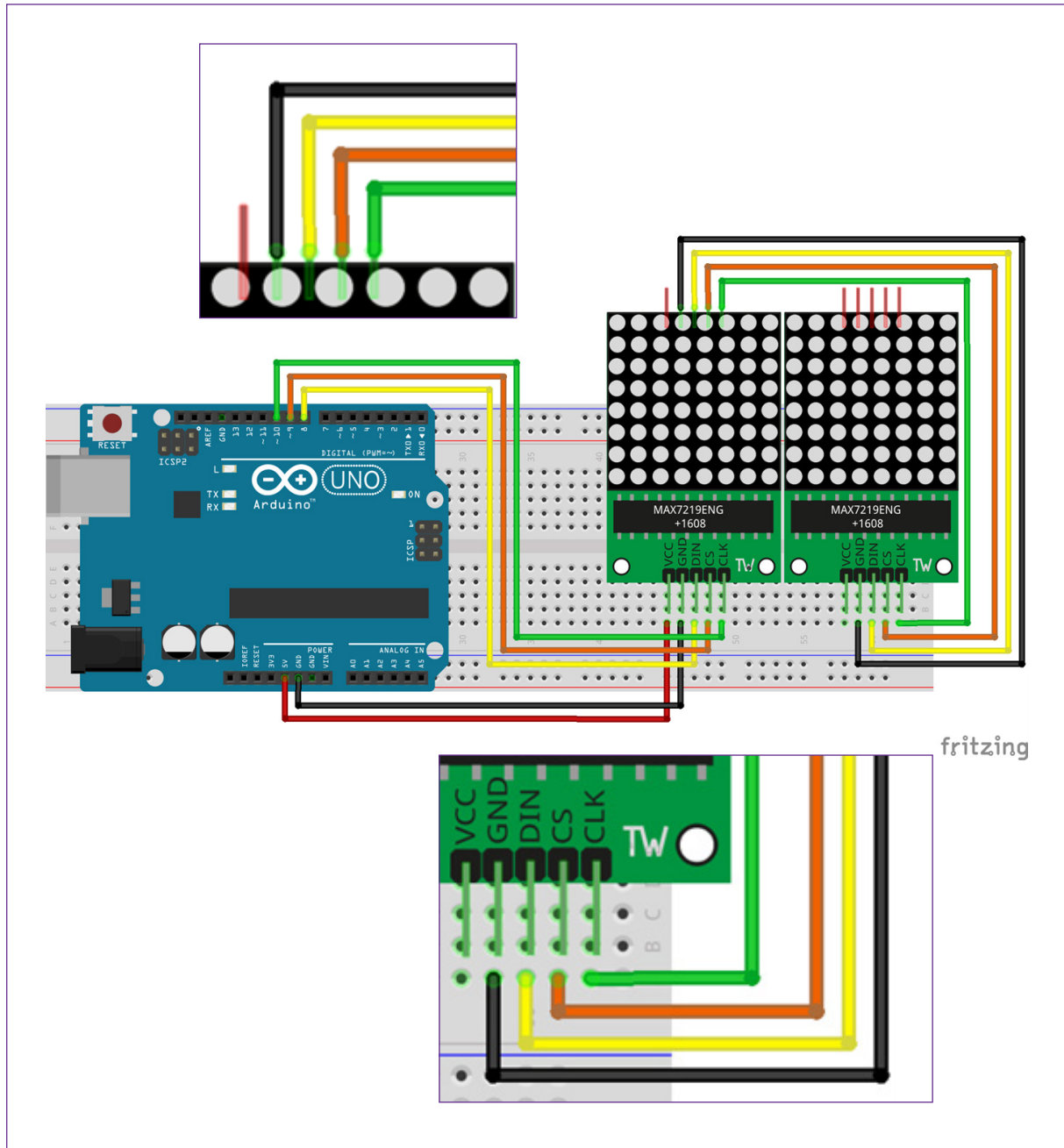
Figura 10 - Conexão do pino DIN entre as matrizes de LED



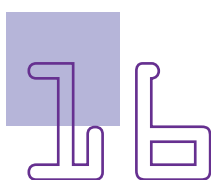
Fonte: Fritzing, 2023.

Pegue um quarto jumper macho-fêmea para interligar os pinos GND das matrizes de LED. Da mesma forma que a pinagem anterior, você conectará a ponta fêmea no pino GND superior da primeira matriz de LEDs e o terminal macho na protoboard, no furo relativo ao pino GND da segunda matriz de LEDs (figura 11).

Figura 11 - Conexão do pino GND entre as matrizes de LED

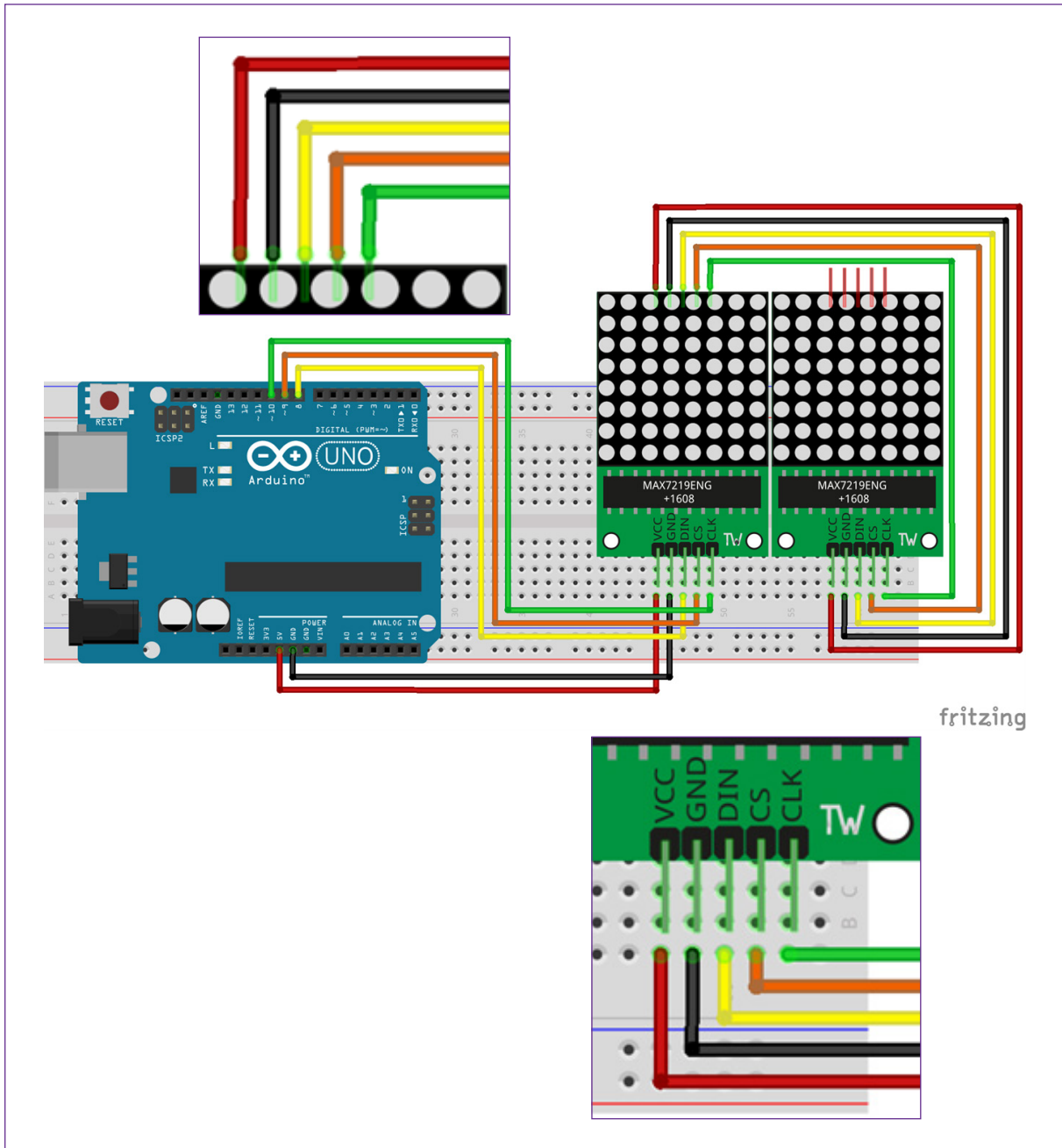


Fonte: Fritzing, 2023.

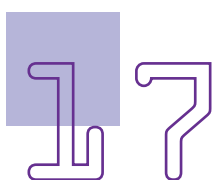


Pegue o último jumper macho-fêmea para interligar os pinos VCC das matrizes de LED. Da mesma forma que a pinagem anterior, você conectará a ponta fêmea no pino VCC superior da primeira matriz de LEDs e o terminal macho na protoboard, no furo relativo ao pino VCC da segunda matriz de LEDs (figura 12).

Figura 12 - Conexão do pino VCC entre as matrizes de LED



Fonte: Fritzing, 2023.



**Obs:** é importante verificar a documentação do seu módulo matriz LED 8x8 para confirmar quais pinos correspondem a quais portas, pois pode haver variações entre os diferentes modelos e fabricantes.

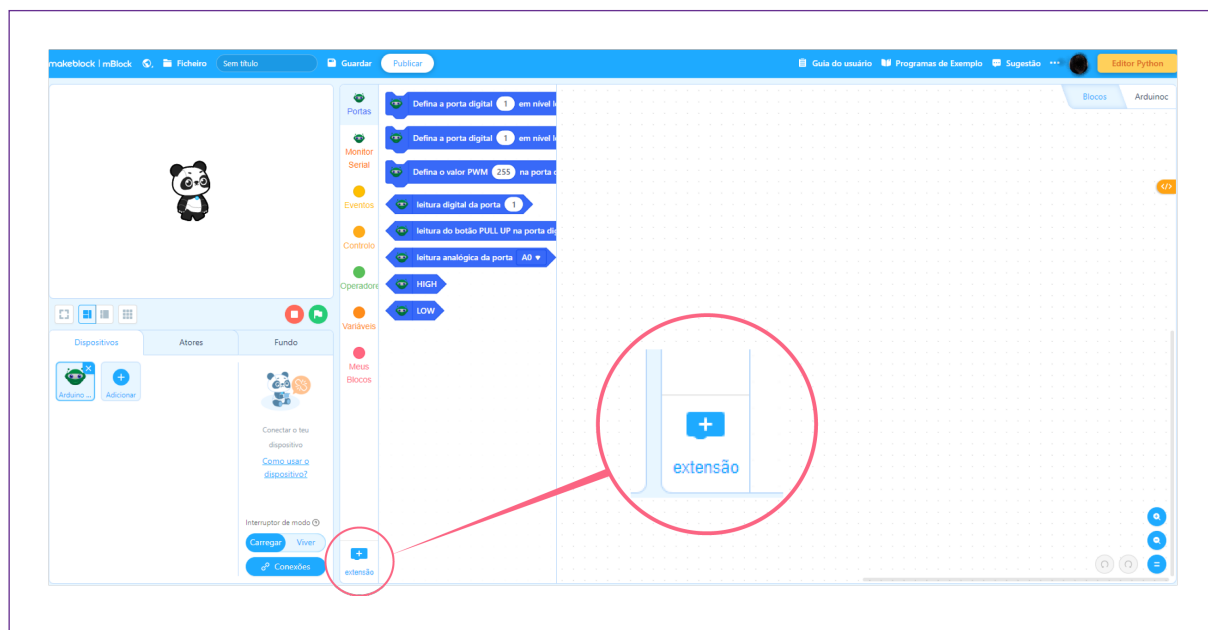
Cuidado para não inverter as conexões de alimentação, pois isso pode danificar o módulo ou o Arduino.

Agora podemos ir para a programação.

## Programação

Para programar seu protótipo de olhos piscantes, você precisa primeiro instalar uma extensão no mBlock. Para isso, abra o mBlock 5.0 e localize, na página inicial, o botão **Extensão**.

Figura 13 - Página inicial do mBlock e detalhe do botão Extensão



Fonte: mBlock 5.0, 2023.

Você será direcionado para a página de **Extensão** do mBlock. Para este projeto, você utilizará a extensão módulo **RP - Matriz de LEDs**. Clique em **Adicionar**.

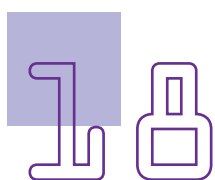
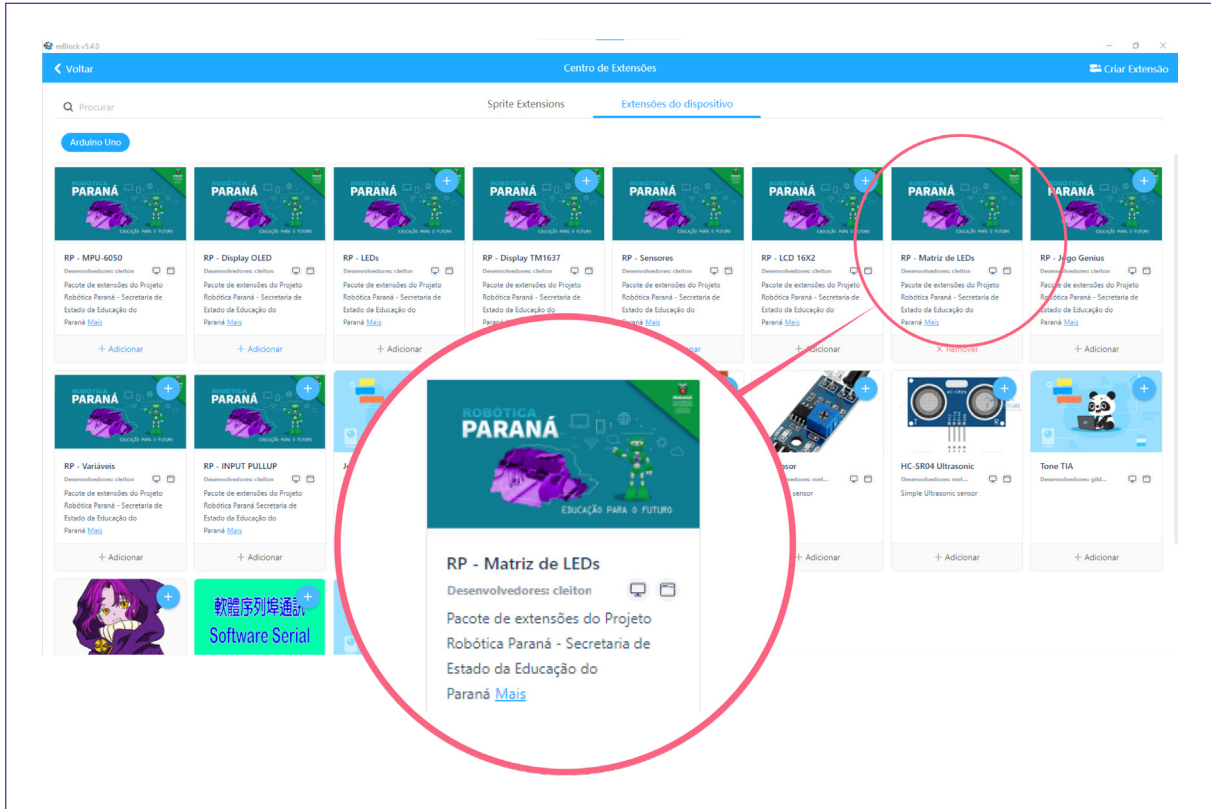


Figura 14 - Página de extensões e detalhe da extensão RP - Matriz de LEDs

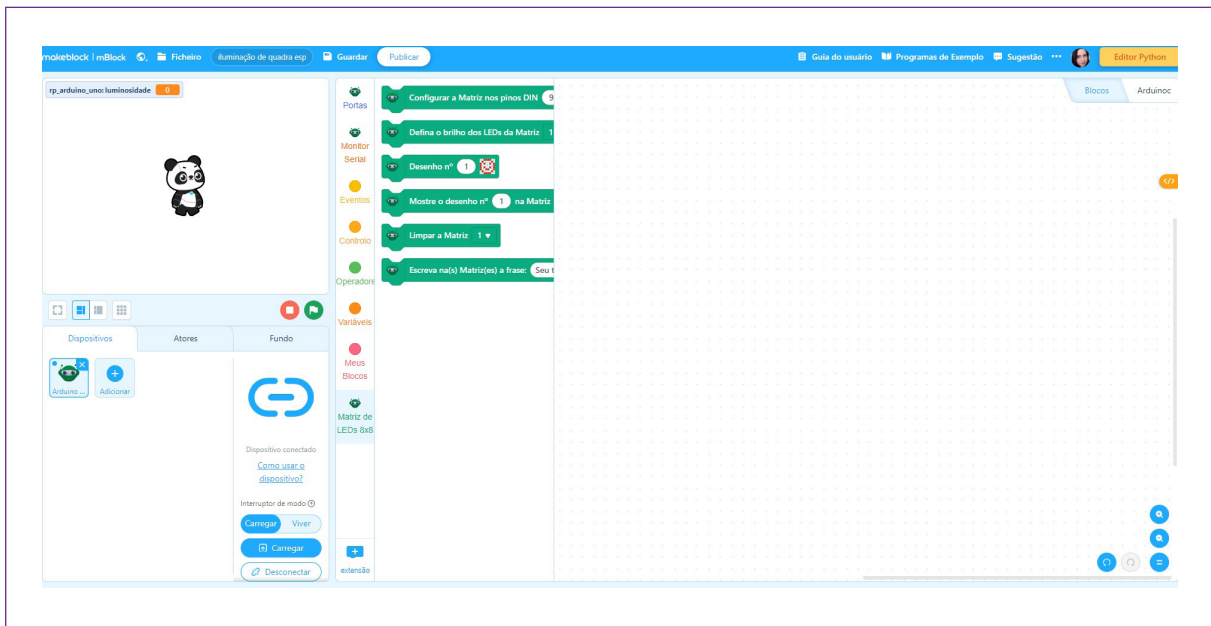


Fonte: mBlock 5.0, 2023.



Agora, você verá que há no mBlock uma nova categoria de comandos (Matriz de LEDs 8x8), a última presente na aba de blocos.

Figura 15 – Blocos da extensão RP - Matriz de LEDs 8x8.

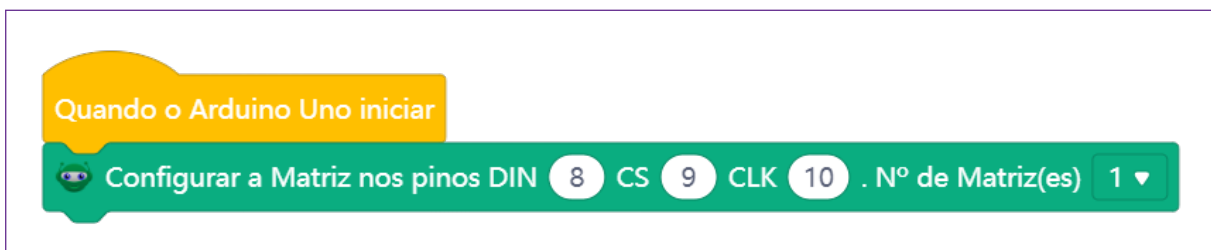


Fonte: mBlock 5.0, 2023.

Tudo pronto para começar a programar? Então, aí vamos nós! Primeiro selecione o bloco de **Evento** <quando o Arduino Uno começar>.

Agora, na categoria **Matriz de LEDs 8x8**, selecione o bloco <Configurar a Matriz nos pinos DIN...>, posicionando-o abaixo do primeiro bloco, como mostra a imagem a seguir.

Figura 16 - Blocos iniciais Matriz de LEDs.



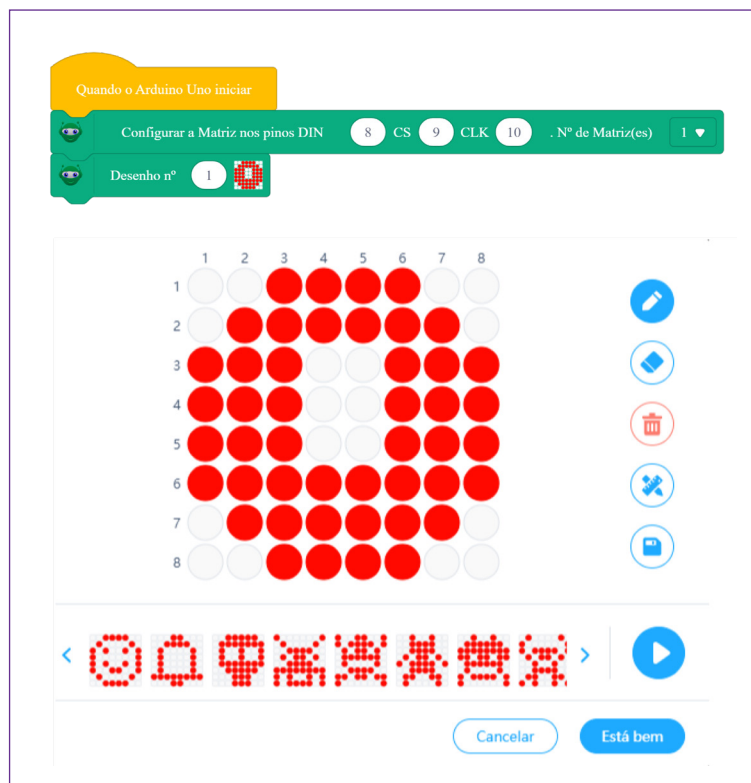
Fonte: mBlock 5.0, 2023.



Tenha especial atenção na inserção do número dos pinos DIN, CS e CLK, que devem obedecer a montagem feita com Arduino.

O próximo bloco que você vai inserir é o de <Desenho nº 1>. Você deve incluir este bloco na programação clicar sobre a carinha feita de pontos vermelhos. Você agora vai desenhar o olho aberto do robô. Para isso, selecione o ícone do lápis, e na matriz de LEDs representada no bloco do mBlock, vá pintando as bolinhas de maneira a criar o olho aberto. Ao finalizar seu desenho, clique em **Está bem**. Esse será o desenho 1, do olho aberto do robô (figura 17).

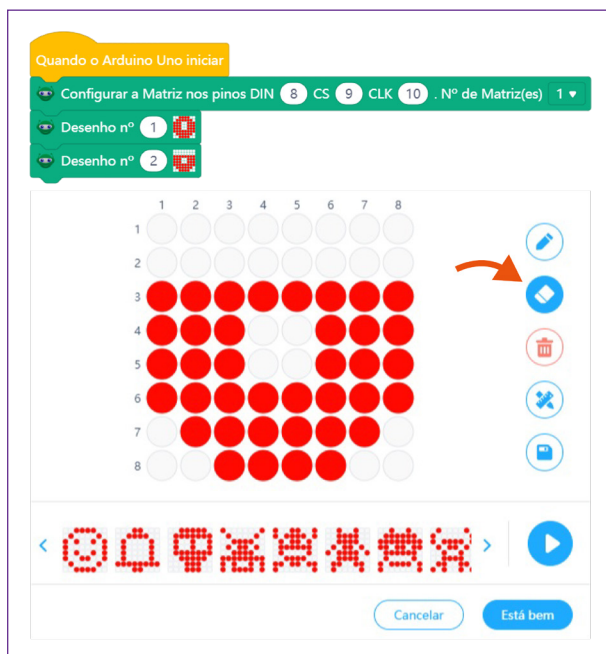
Figura 17 - Janela para edição de desenhos na matriz de LEDs.



Fonte: mBlock 5.0, 2023.

Para criar a impressão que o robô está piscando, você precisará duplicar esse bloco e editar o desenho na matriz diversas vezes.

Figura 18 – Segundo desenho do olho na matriz de LEDs.

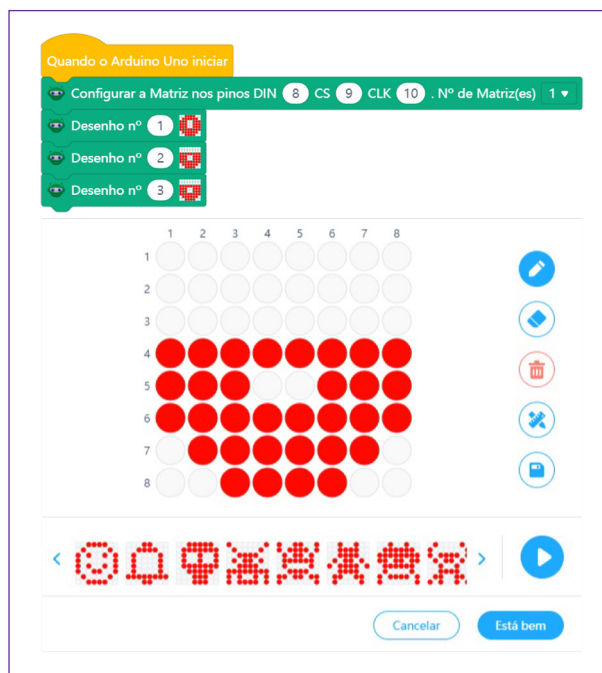


Fonte: mBlock 5.0, 2023.

Duplicate o último bloco, altere o número do desenho para **2** e clique sobre a representação da matriz de LEDs. Edite o desenho do olho, apagando (com a borracha, abaixo do ícone do lápis) os pontos, diminuindo o olho do robô. Ao finalizar seu desenho, clique em **Está bem**.

A seguir, duplique novamente, altere o número do desenho para **3** e repita os passos para editar o desenho, fechando mais um pouco o olho do robô. Sempre que terminar de editar seu desenho, clique em **Está bem**.

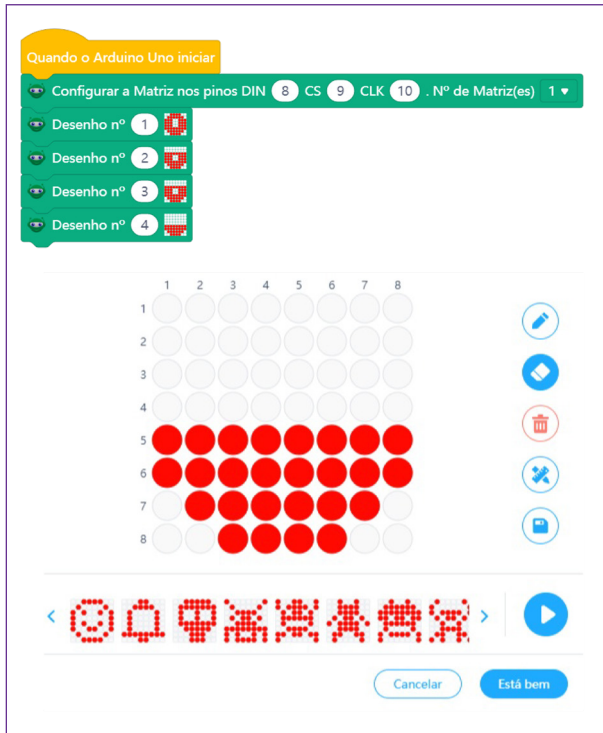
Figura 19 - Terceiro desenho do olho na matriz de LEDs.



Fonte: mBlock 5.0, 2023.



Figura 20 - Quarto desenho do olho na matriz de LEDs.

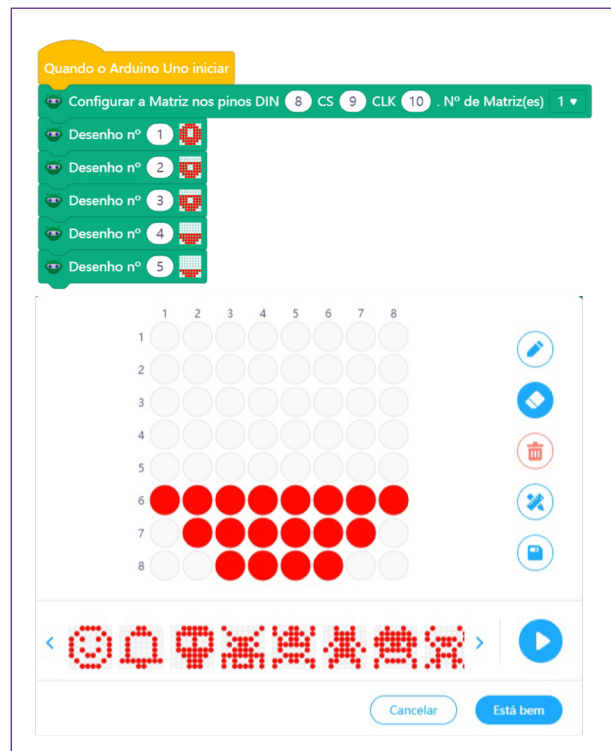


Fonte: mBlock 5.0, 2023.

Execute a quinta duplicação, alterando para <Desenho nº 5> e editando o desenho na matriz de LEDs (figura 21).

Realize a duplicação do último bloco, altere para <Desenho nº 4> e clique na matriz de LEDs, editando o olho (figura 20).

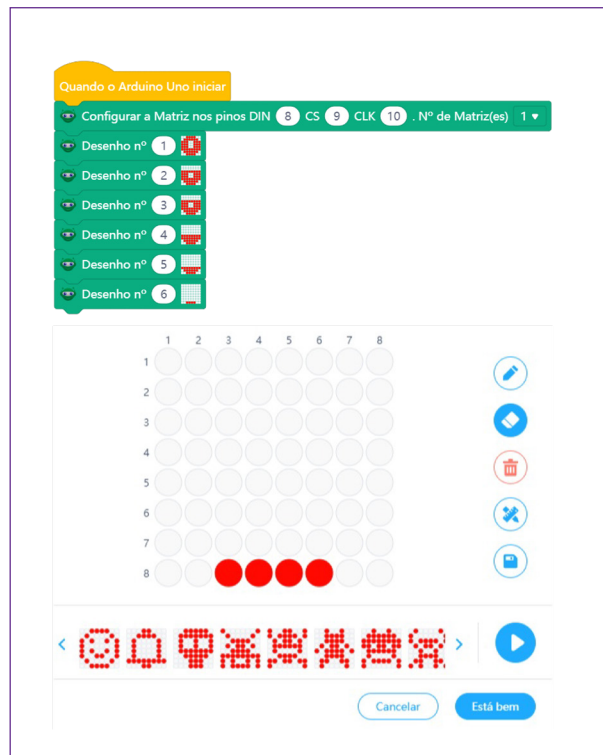
Figura 21 - Quinto desenho do olho na matriz de LEDs.



Fonte: mBlock 5.0, 2023.

Agora vamos fazer a última parte do fechamento do olho: duplique novamente o bloco <Desenho nº> e altere seu número para 6. Ao clicar para editar a matriz de LEDs, deixe apenas um traço (figura 22).

Figura 22 - Sexto desenho do olho na matriz de LEDs.



Fonte: mBlock 5.0, 2023.

Você acabou de desenhar na matriz de LEDs 8x8 representada no mBlock todas as fases dos olhos do robô. Agora, para criar a animação dos olhos, você utilizará o bloco <Repetir para sempre> presente na categoria **Controle**.



O código ficará como a figura a seguir.

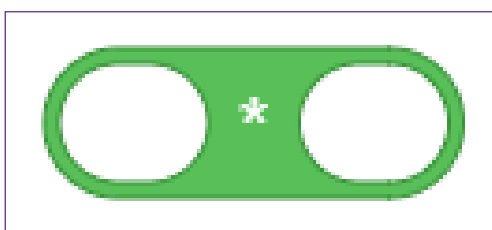
Figura 23 - Programação da matriz.



Já configuramos a matriz nos pinos e desenhamos as formas na matriz de LEDs. Agora, dentro do bloco <Repetir para sempre> insira o bloco <Mostre o desenho nº 1 na Matriz 1>.

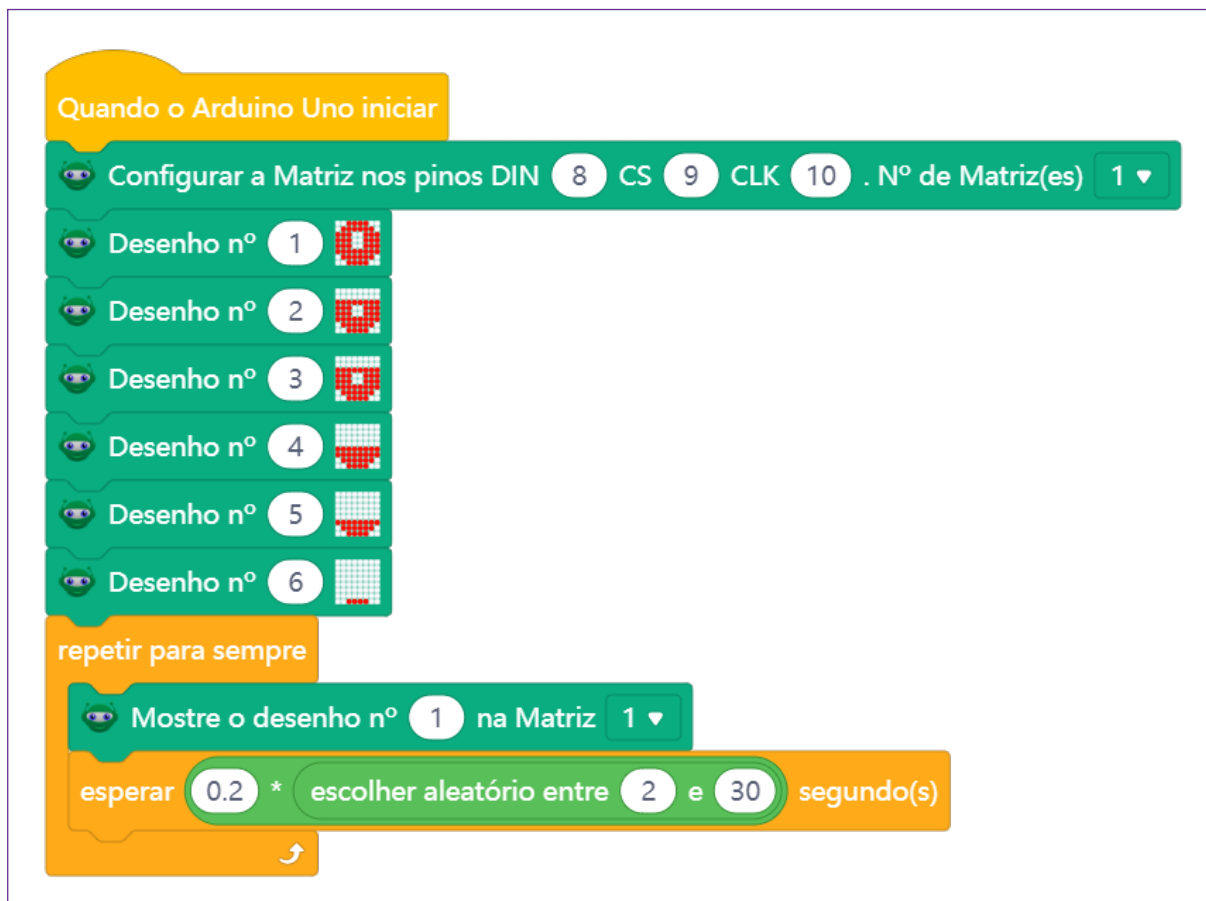
Busque na categoria **Controle** o bloco <esperar 1 segundo(s)> e insira na programação. Dentro do espaço da quantidade de segundos, você irá adicionar um outro bloco, presente na categoria **Operadores**, com o sinal de multiplicação (\*).

Figura 24 - Bloco multiplicação



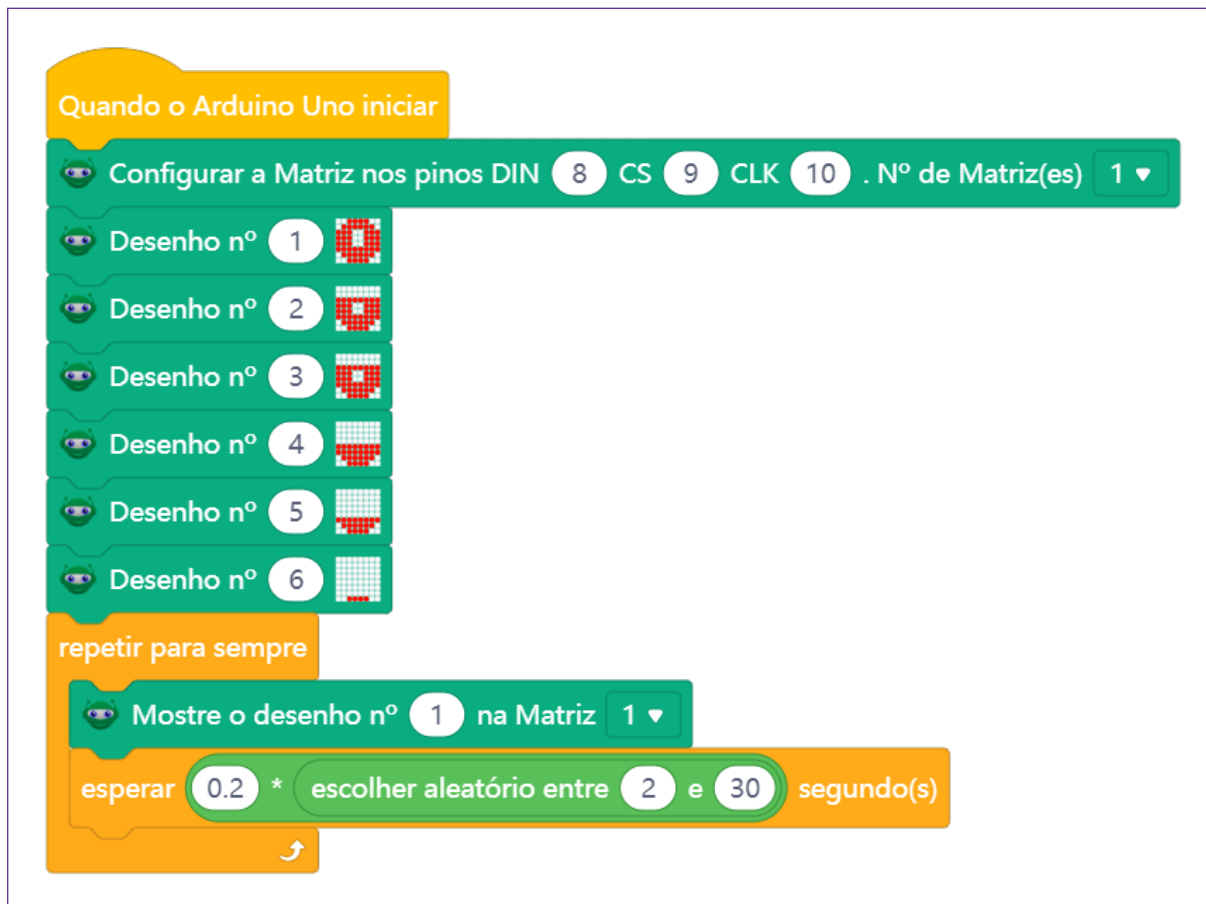
Selecione o bloco de multiplicação e o arraste para dentro do espaço no bloco <esperar 1 segundo(s)> (figura 25).

Figura 25 - Programação da matriz, com temporização.



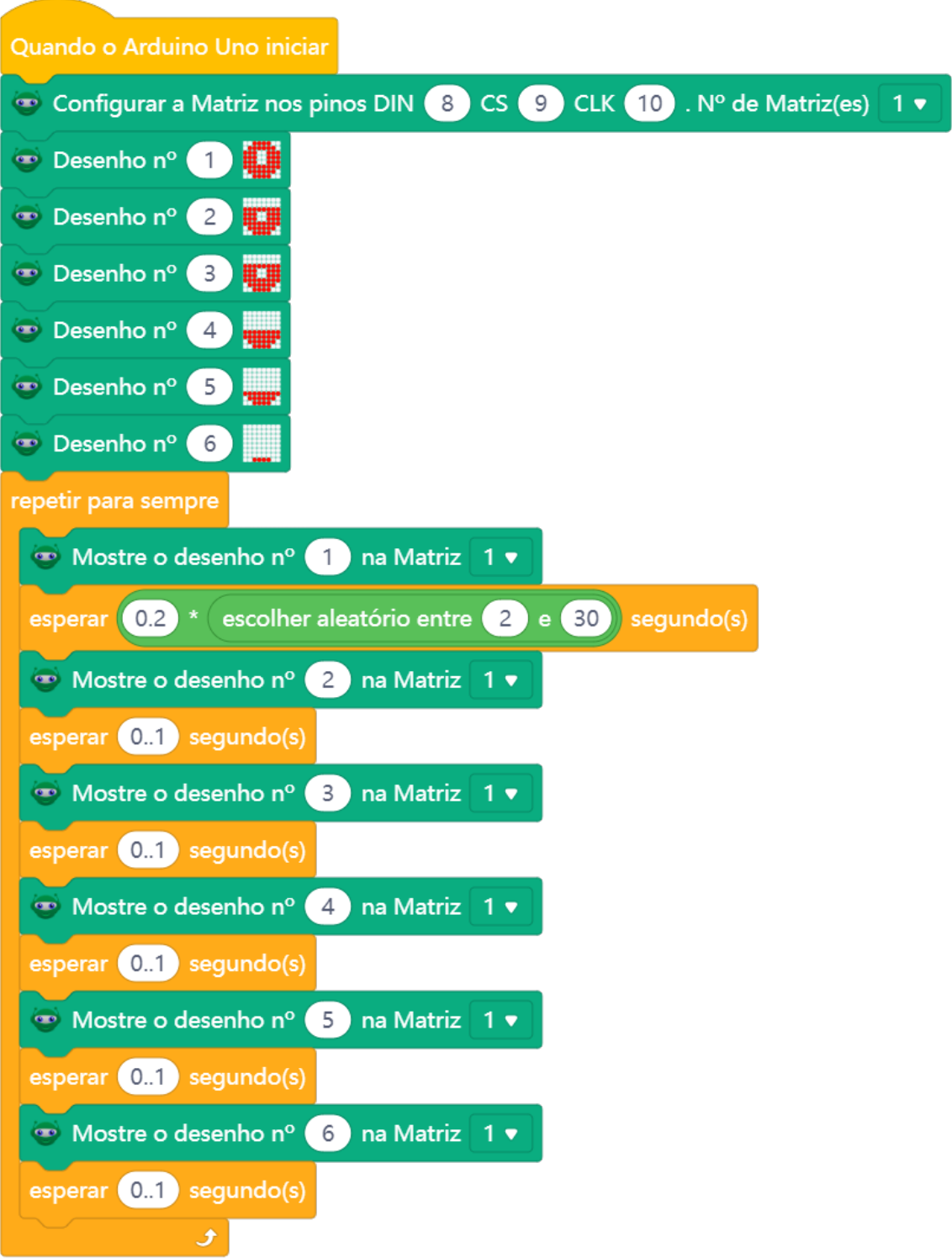
No primeiro espaço do bloco de multiplicação, insira **0.2** e, novamente na categoria **Operadores**, selecione o bloco <escolher aleatório entre 1 e 10> para inseri-lo no segundo espaço do bloco de multiplicação. Altere os valores para 2 e 30, (figura 26).

Figura 26 - Programação da matriz, com a temporização da troca de imagem









Agora, na categoria **RP-Matriz de LEDs 8x8**, selecione o bloco <Mostre o desenho nº 1 na Matriz 1>. E novamente, insira um novo bloco de <esperar 1 segundo(s)>. Altere o número para **0.01**. Estes dois últimos blocos, <Mostre o desenho...> e <esperar 0.01 segundo(s)>, serão repetidos diversas vezes. Cada vez que duplicar os blocos, altere o nº do desenho em ordem crescente. Não altere o valor dos segundos. Veja na figura o código como fica (Aqui temos <Mostre o desenho nº 1> <esperar 0.01 segundo(s)> e depois <Mostre o desenho nº 2> e assim por diante) (figura 27).

Figura 27 - Programação da matriz.



Quando o Arduino Uno iniciar

- Configurar a Matriz nos pinos DIN 8 CS 9 CLK 10 . N° de Matriz(es) 1 ▼
- Desenho nº 1 
- Desenho nº 2 
- Desenho nº 3 
- Desenho nº 4 
- Desenho nº 5 
- Desenho nº 6 

repetir para sempre

- Mostre o desenho nº 1 na Matriz 1 ▼
- esperar 0.2 \* escolher aleatório entre 2 e 30 segundo(s)
- Mostre o desenho nº 2 na Matriz 1 ▼
- esperar 0.1 segundo(s)
- Mostre o desenho nº 3 na Matriz 1 ▼
- esperar 0.1 segundo(s)
- Mostre o desenho nº 4 na Matriz 1 ▼
- esperar 0.1 segundo(s)
- Mostre o desenho nº 5 na Matriz 1 ▼
- esperar 0.1 segundo(s)
- Mostre o desenho nº 6 na Matriz 1 ▼
- esperar 0.1 segundo(s)



Quando duplicar e alterar para <Mostre o desenho nº 6> e <esperar 0.01 segundo(s)> seu robô terá fechado os olhos. Agora, é preciso programar na sequência inversa de desenhos para que ele volte a abrir os olhos. O processo é o mesmo que você já fez até agora. Duplica os blocos <Mostre o desenho...> e <esperar 0.01 segundo(s)>, alterando o desenho para nº 5 (figura 28).

Figura 28 - Programação da matriz.

The image shows a Scratch-style block diagram for an Arduino Uno. The blocks are as follows:

- Quando o Arduino Uno iniciar** (When the Arduino Uno starts)
- Configurar a Matriz nos pinos DIN 8 CS 9 CLK 10 . Nº de Matriz(es) 1** (Configure the Matrix on pins DIN 8 CS 9 CLK 10 . Nº de Matriz(es) 1)
- Desenho nº 1** (Drawing nº 1)
- Desenho nº 2** (Drawing nº 2)
- Desenho nº 3** (Drawing nº 3)
- Desenho nº 4** (Drawing nº 4)
- Desenho nº 5** (Drawing nº 5)
- Desenho nº 6** (Drawing nº 6)
- repetir para sempre** (repeat forever loop)
- Mostre o desenho nº 1 na Matriz 1** (Show drawing nº 1 on Matrix 1)
- esperar 0.2 \* escolher aleatório entre 2 e 30 segundo(s)** (wait 0.2 \* choose random between 2 and 30 seconds)
- Mostre o desenho nº 2 na Matriz 1** (Show drawing nº 2 on Matrix 1)
- esperar 0.1 segundo(s)** (wait 0.1 seconds)
- Mostre o desenho nº 3 na Matriz 1** (Show drawing nº 3 on Matrix 1)
- esperar 0.1 segundo(s)** (wait 0.1 seconds)
- Mostre o desenho nº 4 na Matriz 1** (Show drawing nº 4 on Matrix 1)
- esperar 0.1 segundo(s)** (wait 0.1 seconds)
- Mostre o desenho nº 5 na Matriz 1** (Show drawing nº 5 on Matrix 1)
- esperar 0.1 segundo(s)** (wait 0.1 seconds)
- Mostre o desenho nº 6 na Matriz 1** (Show drawing nº 6 on Matrix 1)
- esperar 0.1 segundo(s)** (wait 0.1 seconds)
- Mostre o desenho nº 5 na Matriz 1** (Show drawing nº 5 on Matrix 1)
- esperar 0.1 segundo(s)** (wait 0.1 seconds)

Duplicate novamente os dois blocos <Mostre o desenho...> e <esperar 1 segundo(s)>, alterando o número do desenho para 4. E continue realizando este procedimento (de duplicar), alterando para 3 e depois para o desenho 2. Seu código ficará completo (figura 29).

Figura 29 - Programação completa

The image shows a Scratch-style block-based programming environment for an Arduino Uno. The code is organized as follows:

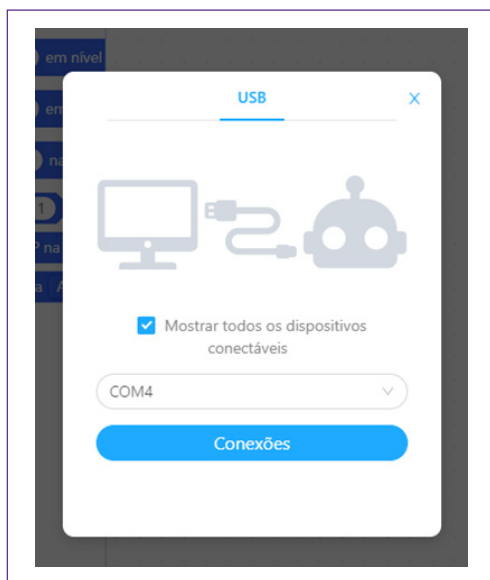
- Quando o Arduino Uno iniciar** (When the Arduino Uno starts):
  - Configurar a Matriz nos pinos DIN 8 CS 9 CLK 10 . N° de Matriz(es) 1
  - Desenho n° 1
  - Desenho n° 2
  - Desenho n° 3
  - Desenho n° 4
  - Desenho n° 5
  - Desenho n° 6
- repetir para sempre** (Repeat forever):
  - Mostre o desenho n° 1 na Matriz 1
  - esperar 0.2 \* escolher aleatório entre 2 e 30 segundo(s)
  - Mostre o desenho n° 2 na Matriz 1
  - esperar 0.1 segundo(s)
  - Mostre o desenho n° 3 na Matriz 1
  - esperar 0.1 segundo(s)
  - Mostre o desenho n° 4 na Matriz 1
  - esperar 0.1 segundo(s)
  - Mostre o desenho n° 5 na Matriz 1
  - esperar 0.1 segundo(s)
  - Mostre o desenho n° 6 na Matriz 1
  - esperar 0.1 segundo(s)
  - Mostre o desenho n° 5 na Matriz 1
  - esperar 0.1 segundo(s)
  - Mostre o desenho n° 4 na Matriz 1
  - esperar 0.1 segundo(s)
  - Mostre o desenho n° 3 na Matriz 1
  - esperar 0.1 segundo(s)
  - Mostre o desenho n° 2 na Matriz 1
  - esperar 0.1 segundo(s)



Agora conecte o Arduino via cabo USB no netbook (ou chromebook) para carregar o código pronto. Para isso, após conectado na USB, clique **Conectar** (figura 30).

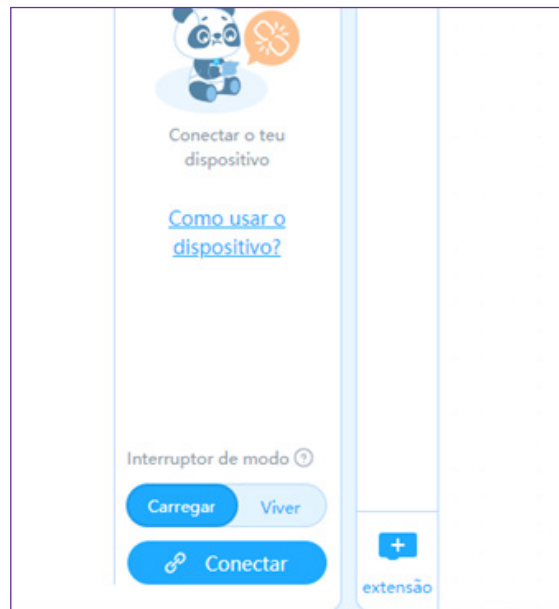
Quando clicar, aparecerá uma janela solicitando que se marque primeiro, “Mostrar todos os dispositivos conectáveis”. E depois, selecione qual a porta na qual está conectado o Arduino (COM1, COM2, COM4, etc). Opte sempre pela porta com o maior número. Clique depois em Conexões, como mostra a figura 31.

Figura 31 – Detalhe para o botão Conexões.



Fonte: mBlock 5.0, 2023.

Figura 30 – Detalhe para o botão Conectar.



Fonte: mBlock 5.0, 2023.

Após, clique em **Carregar**, acima do botão **Conectar** (figura 30). O código será carregado no Arduino e você poderá ver o funcionamento do seu protótipo.



## Desafios

- i. Que tal criar com papel ou compor com objetos disponíveis para criar o rosto do robô?
- ii. Por último, que tal criar jogos a partir do que aprendeu nesta aula, podendo por exemplo, fazer formatos diferentes dos olhos, com expressões (de raiva, medo etc), e que a equipe adversária adivinhe ou imite a expressão que seu robô está fazendo.

## 3. Feedback e finalização

Após realizar seu protótipo e apresentar aos colegas, converse sobre os projetos, atendendo os seguintes pontos:

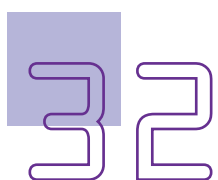
- a. Qual(is) conteúdo(s) você teve maior afinidade ou apresentou melhor desempenho?
- b. Houve momentos de dificuldade? Quais foram? De que forma você e seu grupo conseguiram superar os obstáculos que encontraram?
- c. Observe os conceitos que foram contemplados nos protótipos construídos nesta aula, por você e seus colegas.
- d. Quais melhorias, ajustes ou novas aplicações vocês observaram?

## Referências

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Brasília, 2018. Disponível em: [http://basenacionalcomum.mec.gov.br/images/BNCC\\_EI\\_EF\\_110518\\_versaofinal\\_site.pdf](http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf). Acesso em: 15 abr. 2022.

DANTAS, Tiago. Por que piscamos? **Mundo Educação**. Disponível em: <https://mundoeducacao.uol.com.br/curiosidades/por-que-piscamos.htm#:~:text=As%20l%C3%A1grimas%2C%20produzidas%20a%20todo,uma%20limpeza%20natural%20da%20c%C3%B3rnea>. Acesso em: 26 jan. 2023.

MAIA, Julio César. A linguagem corporal dos olhos que você não vê. **Blog Não Verbal**. Mundo Educação. Disponível em: <https://www.naoverbal.com.br/a-linguagem-corporal-dos-olhos-que-voce-nao-ve/>. Acesso em: 26 jan. 2023.



**DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)**  
**COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)**

**EQUIPE ROBÓTICA PARANÁ**

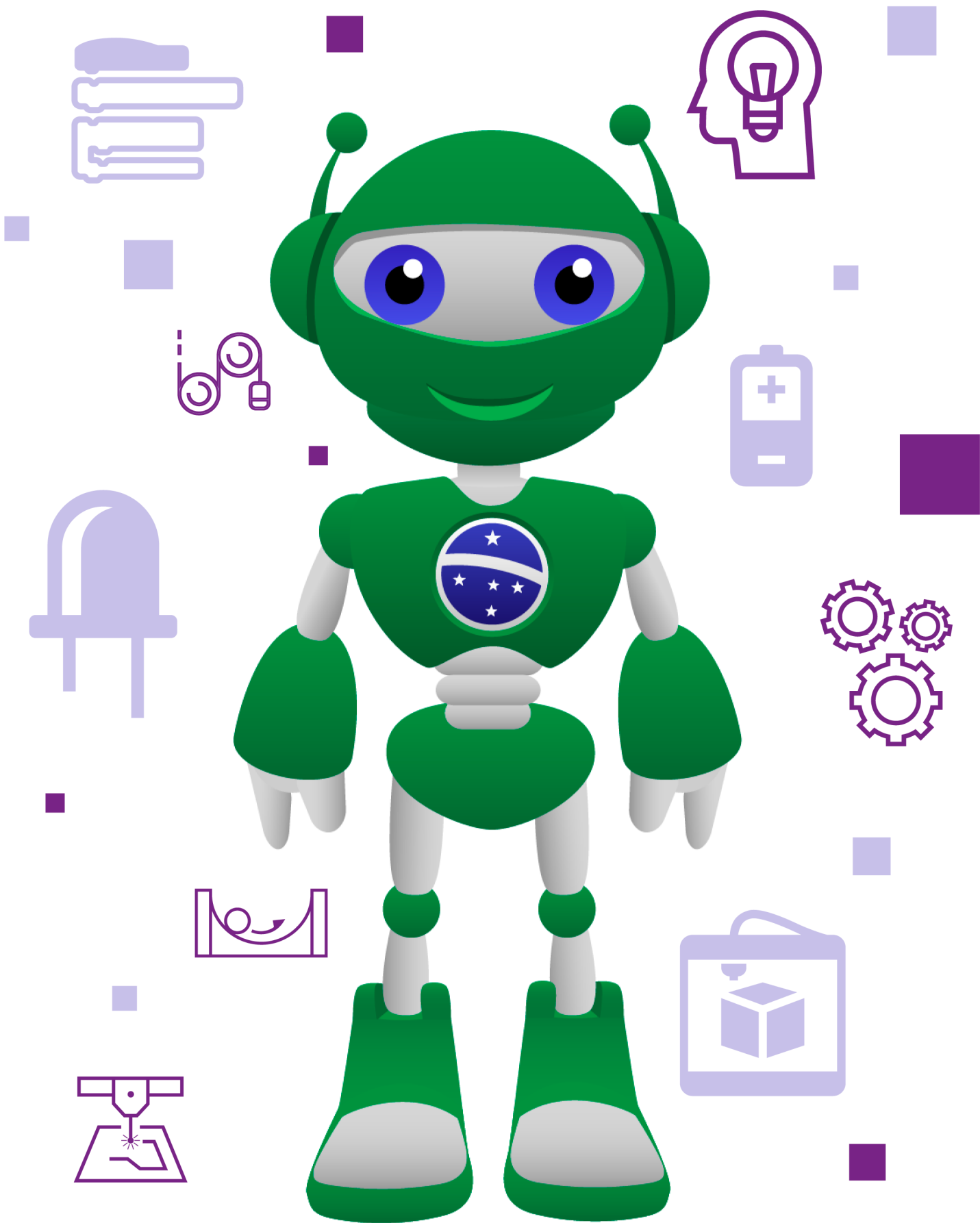
Andrea da Silva Castagini Padilha  
Cleiton Rosa  
Darice Alessandra Deckmann Zanardini  
Edgar Cavalli Júnior  
Edna do Rocio Becker  
José Feuser Meurer  
Marcelo Gasparin  
Michele Serpe Fernandes  
Michelle dos Santos  
Orlando de Macedo Júnior  
Roberto Carlos Rodrigues

Os materiais, aulas e projetos da “Robótica Paraná” foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação (Seed), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica.

Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.



Este trabalho está licenciado com uma Licença  
Creative Commons – CC BY-NC-SA  
[Atribuição - NãoComercial - Compartilha Igual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)



DTI - DIRETORIA DE TECNOLOGIA E INOVAÇÃO  
DETED - DEPARTAMENTO DE TECNOLOGIAS EDUCACIONAIS