

# ROBÓTICA

Primeiros Passos

Módulo 2



AULA **20**

## Equalizador com sensor de som

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

**Validação de Conteúdo**

Cleiton Rosa

**Revisão Textual**

Orlando de Macedo Junior

**Leitura Crítica**

Darice Alessandra Deckmann Zanardini

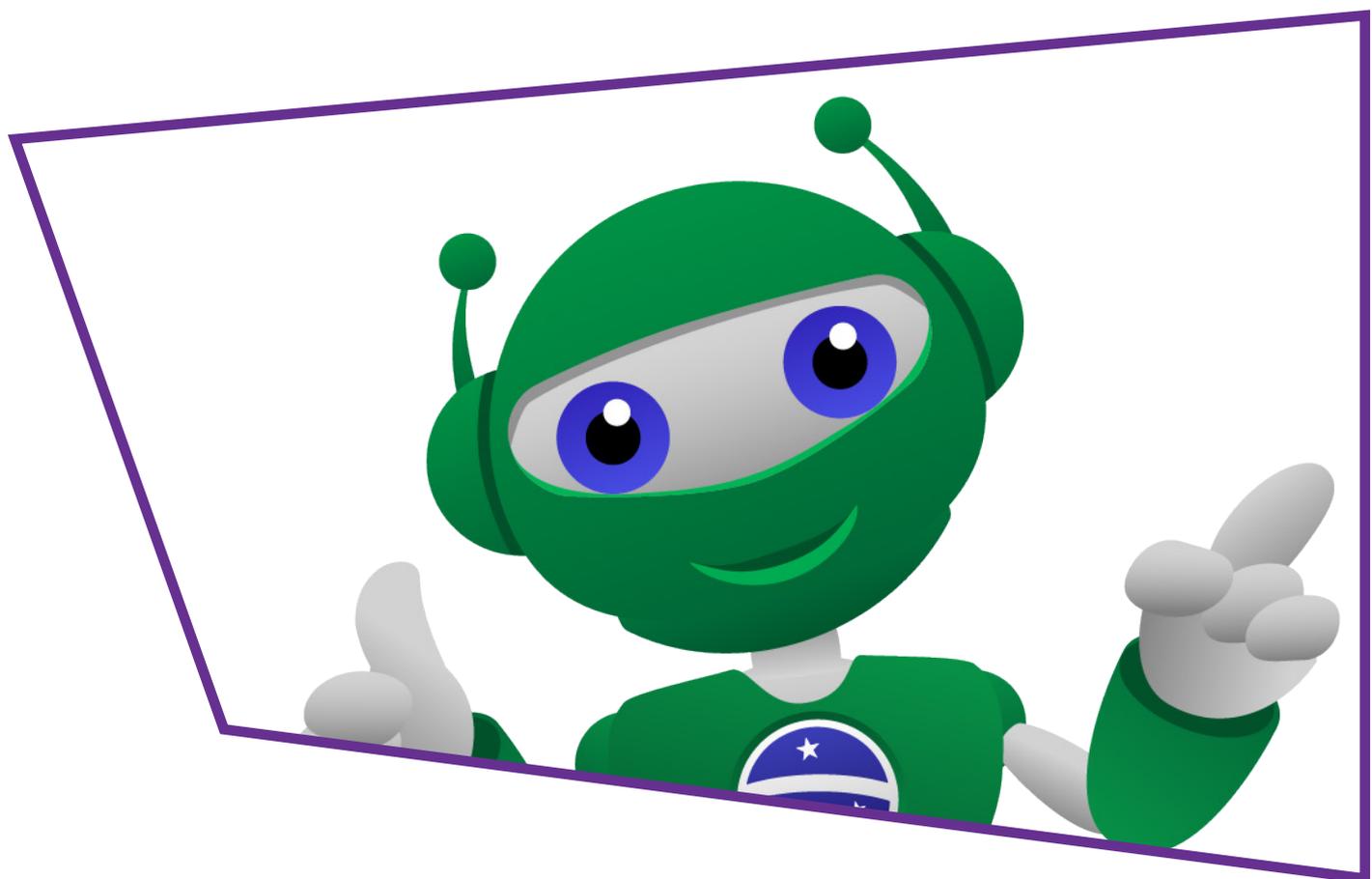
**Projeto Gráfico e Diagramação**

Edna do Rocio Becker

2023

# SUMÁRIO

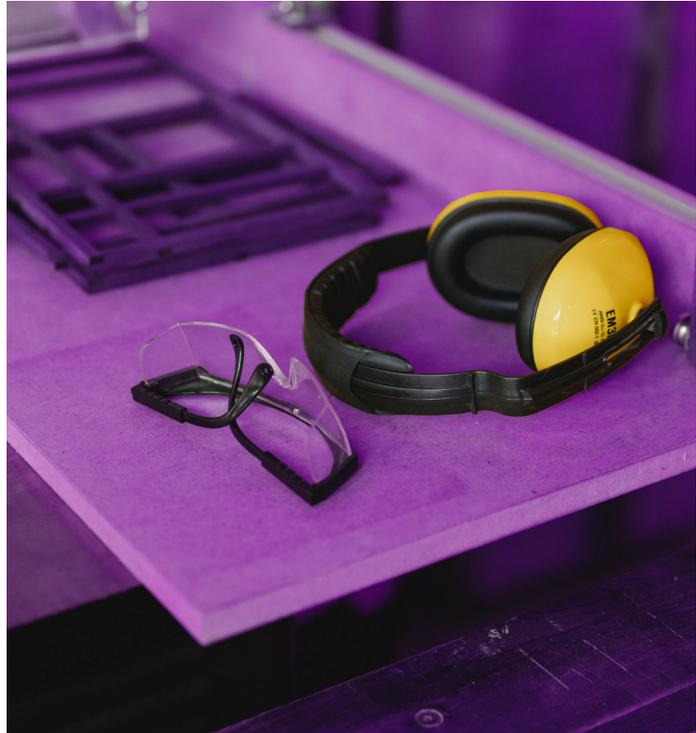
Introdução	2
Objetivos desta aula	2
Competências gerais previstas na BNCC	3
Habilidades do século XXI a serem desenvolvidas	4
Roteiro da aula	5
1. Contextualização	5
2. Conteúdo	6
3. Feedback	22
Referências	22



## Introdução

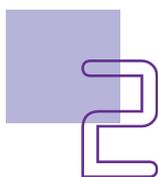
Quanto ruído tem no pátio da escola na hora do intervalo? E em um show de música ao vivo? Ruídos altos e até mesmo música alta podem ser prejudiciais para a saúde. Não seria interessante ter meios de medir a quantidade de ruídos em um ambiente para poder controlar o que seria adequado aos nossos ouvidos?

Nesta aula, vamos criar um medidor de intensidade sonora, que funcionará com LEDs, criando um show de luzes a partir de sons.



## Objetivos desta aula

- Integrar o sensor de som KY-038 a múltiplos LEDs;
- Diferenciar porta analógica de porta digital;
- Relacionar no código as diferenças na sensibilidade do sensor, atrelados à cor do LED;
- Associar a diferença da captura do som pelo sensor com a geração de ruídos.



## Competências gerais previstas na BNCC

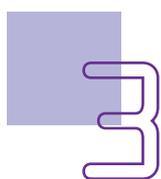
**[CG02]** - Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas)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

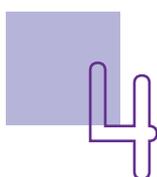


## Habilidades do século XXI a serem desenvolvidas

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

## Lista de materiais

- 1 placa Arduino Uno R3;
- 1 cabo USB;
- 1 placa de expansão (protoboard);
- 5 resistores de 220 ohms;
- 12 jumpers macho-macho;
- 1 barra de LED
- 1 sensor de som KY-038;
- 1 chave de fenda Phillips;
- 1 notebook;
- Software mBlock.



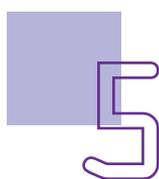


## Roteiro da aula

### 1. Contextualização

Usamos sensores acústicos, seja para avisar da presença de alguém (pelos ruídos), detectar níveis de ruídos em um ambiente ou para identificar as notas musicais e seu volume. Esses componentes são utilizados na segurança residencial, por exemplo, para perceber sons, identificando as ondas sonoras de um ambiente. Mas esse sensor poderia também, ser utilizado em bibliotecas e hospitais, locais que requerem silêncio e baixa emissão de ruídos.

Ao invés de criar um alarme de som, vamos, nesta aula, criar um show de luzes com LEDs que serão acionados conforme os sons detectados por uma música.



## 2. Conteúdo

O componente que utilizaremos nesta aula é o módulo sensor de som KY-038. Ele tem as funções de perceber (detectar) a quantidade de ruído do ambiente e de acender ou apagar os LEDs conforme a variação sonora, pois possui um microfone de condensador elétrico.

Este componente possui um microfone, um ajuste de sensibilidade e quatro pinos:

**AO** - Pino de saída analógico (que faz a leitura da intensidade do som). É recomendado usar essa saída para medir a variação do som.

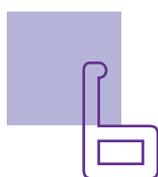
**G** - (GND) Ground (alimentação negativa que deve ser conectada à porta GND do Arduino).

**+ - Saída 5 V** (alimentação positiva que deve ser conectada à porta 5V do Arduino).

**DO** - Pino de saída digital. Em um ambiente silencioso, essa saída digital se mantém em nível baixo e altera para nível alto quando detecta algum som.

O componente sensor de som KY-038 tem uma parte que permite o ajuste da sensibilidade de captura de som, é chamado trimpot e você pode identificá-lo como a parte azul, que tem um parafuso. Utilize uma chave de fenda para fazer a calibragem dele, testando com o som de palmas, ou de estalar de dedos (ou mesmo música) para ver se os LEDs do sensor acendem (o que significa que o sensor já capta os decibéis desse ruído).

Figura 1 - Sensor de som KY-038.



## Curiosidade

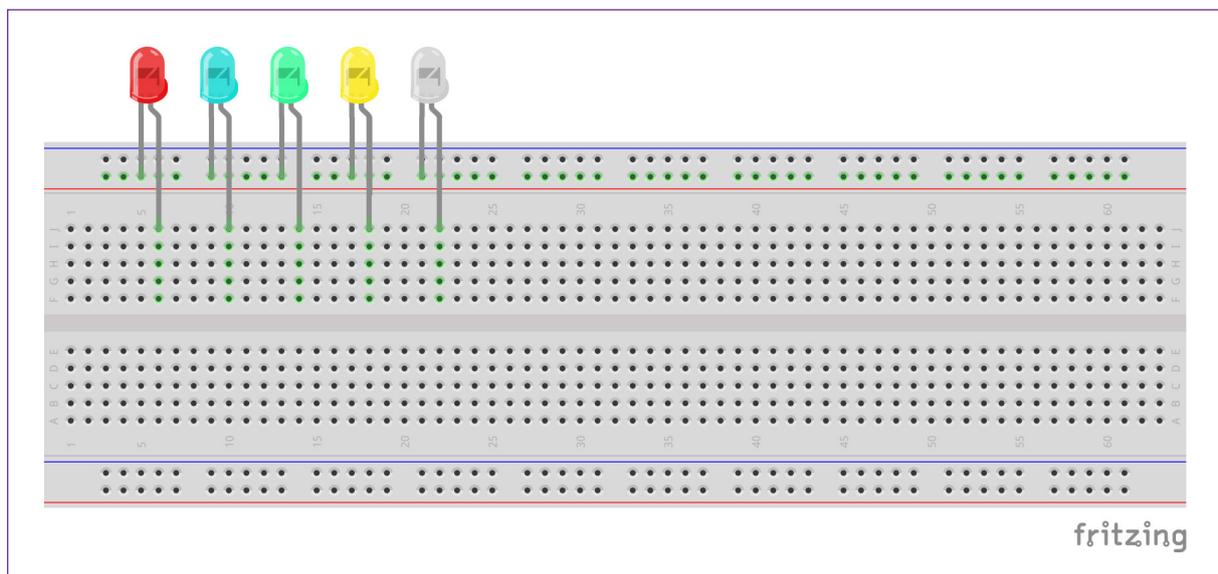
**TRIMPOT** - é um termo que vem da língua inglesa e reúne duas palavras - *Trimmer potentiometer*, ou seja, é um potenciômetro miniatura ajustável. Eles são circuitos de precisão presentes em componentes de áudio e vídeo. Nos componentes de robótica com Arduino, os trimpots são montados diretamente na placa de circuitos, e podem ser ajustados com uma pequena chave de fenda.

Fonte: Wikipedia - Trimpot. Disponível em: <https://pt.wikipedia.org/wiki/Trimpot>.

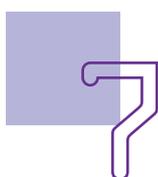
## Montagem

Após separar os materiais, insira 5 LEDs na protoboard. Observe qual é o terminal negativo da ponta de cada LED - esse deverá ser conectado na borda da protoboard, na linha azul (polo negativo da protoboard). O terminal positivo de cada LED deve ser conectado em outra coluna, nas linhas mais ao centro da protoboard (figura 2).

Figura 2 - Inserção dos LEDs na protoboard



Fonte: Fritzing, 2022

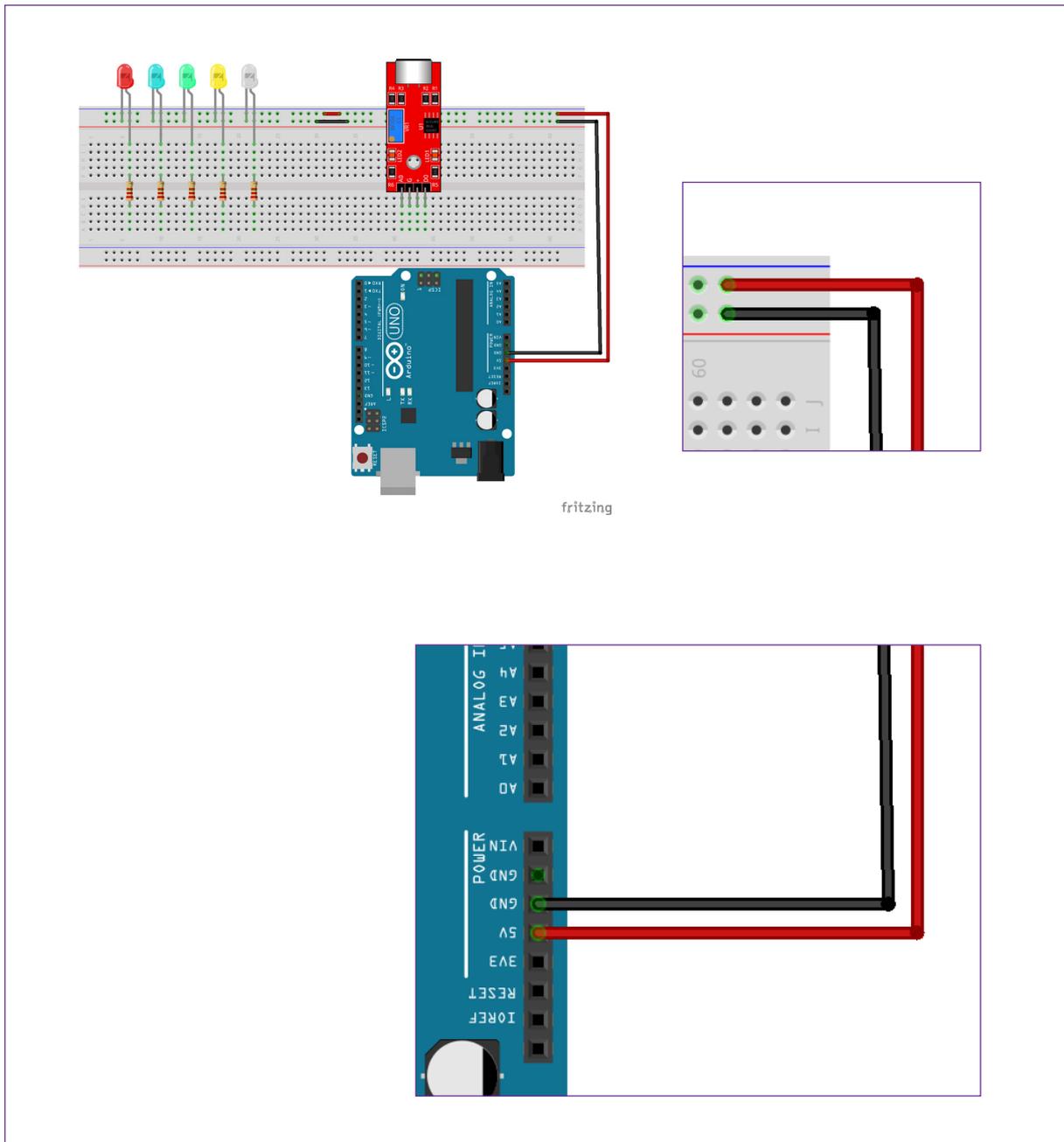




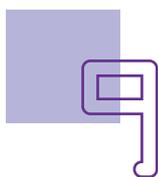
# Equalizador com sensor de som

Agora, insira o módulo sensor de som KY-038 na protoboard (figuras 3 e 4), e um jumper na porta 5V do Arduino e a outra ponta na linha positiva da protoboard. Conecte também outro jumper na porta GND do Arduino e o outro conector, insira na linha negativa da protoboard.

Figura 4 - Inserção dos jumpers para energização da protoboard.



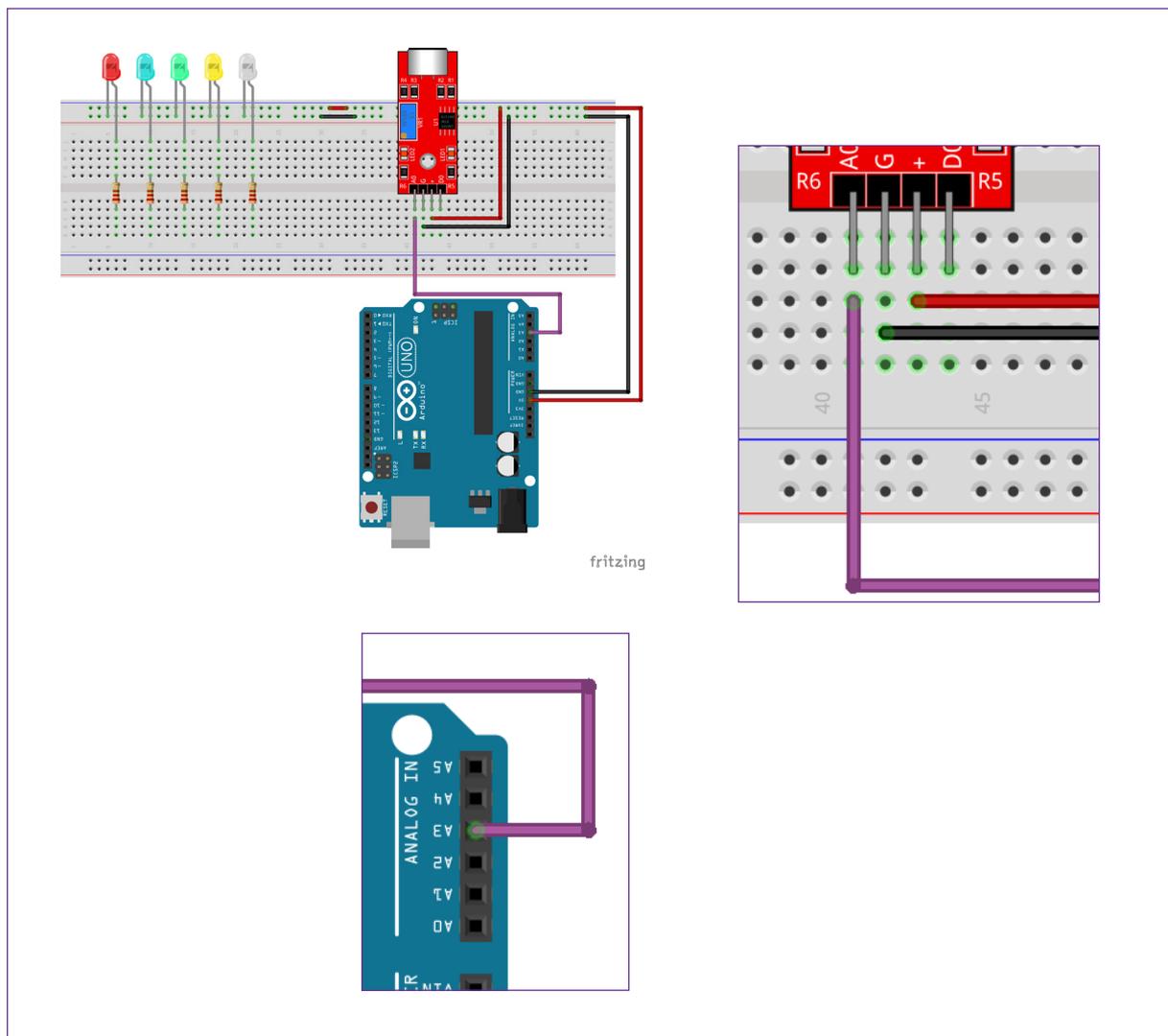
Fonte: Fritzing, 2022



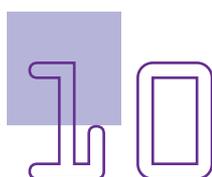
# Equalizador com sensor de som

Insira os jumpers relativos aos terminais do sensor de som KY-038. Primeiro, conecte um jumper na mesma coluna do terminal positivo (indicado com um sinal de +) e a outra ponta, insira na linha positiva da borda da protoboard. Um segundo jumper deve ser conectado na coluna do pino G (que é a saída negativa do sensor) e a outra ponta, insira na linha negativa da borda da protoboard. E por fim, um terceiro jumper deve ser inserido na coluna do terminal A0 do módulo sensor de som KY-038. A outra ponta deste jumper deve ser conectada ao pino A3 da placa Arduino (figura 5).

Figura 5 - Inserção dos jumpers e módulo sensor de som KY-038.



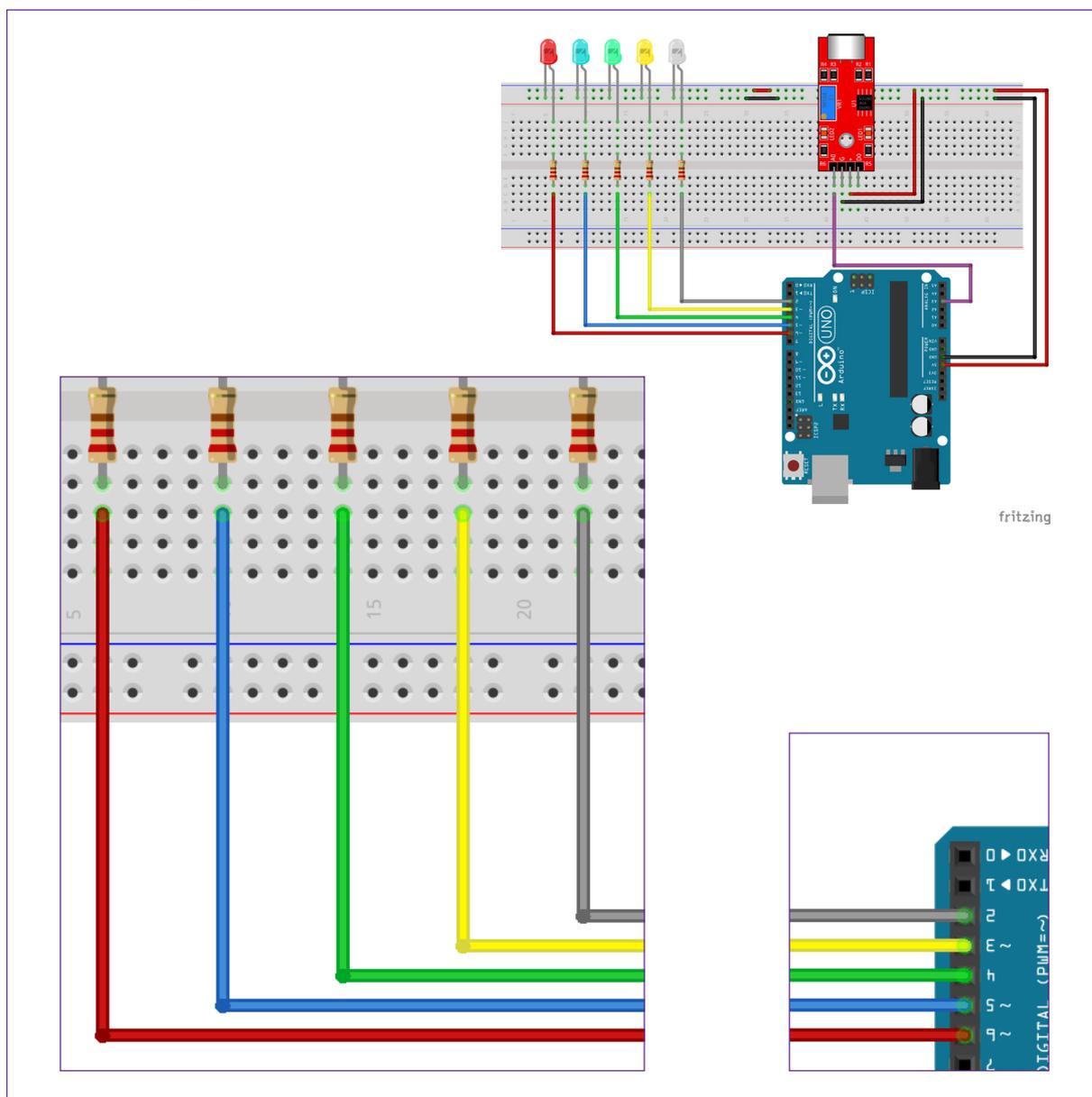
Fonte: Fritzing, 2022



# Equalizador com sensor de som

Agora, separe cinco jumpers para conectar as saídas positivas dos LEDs à porta Arduino. Insira cada um dos jumpers na coluna que está o resistor de cada LED. As outras pontas dos jumpers devem ser inseridas nas portas 2, 3, 4, 5 e 6 (figura 6).

Figura 6 - Inserção dos jumpers relacionados aos LEDs.

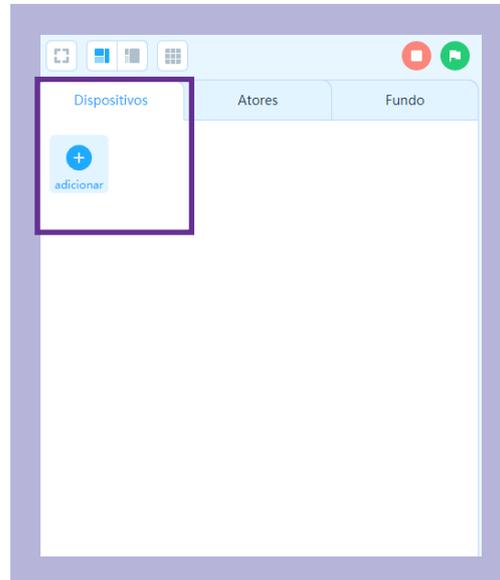


Fonte: Fritzing, 2022



Com isso a montagem está pronta, agora, vamos para a programação utilizando o mBlock. Inicie o programa em seu notebook ou computador. Conecte o mBlock ao Arduino, usando o cabo USB, e depois, clique no ícone <adicionar>, localizado no campo <Dispositivos>.

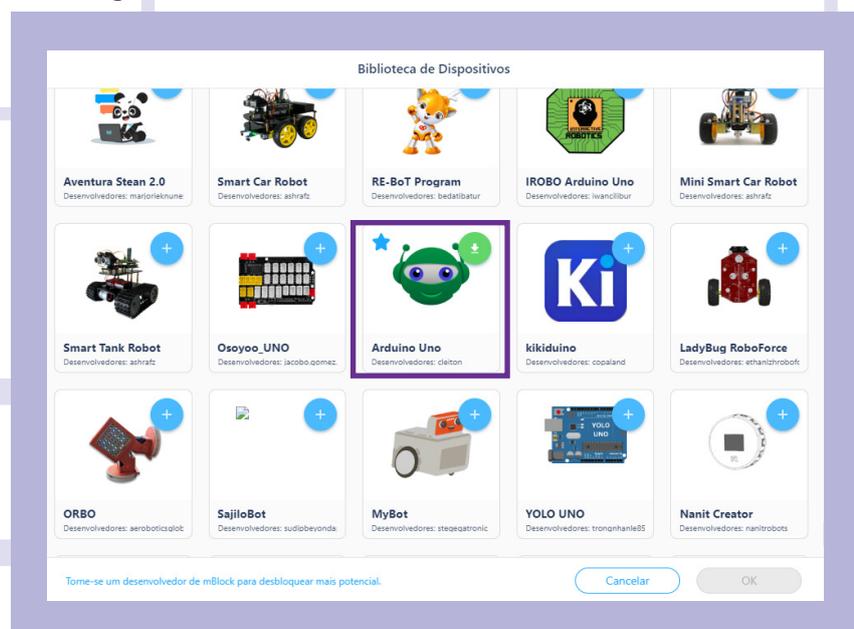
Figura 7 - Conectando o Arduino no mBlock.



Fonte: site mBlock oficial

Figura 8 - Conectando o Arduino no mBlock.

Selecione o Arduino na biblioteca de dispositivos do mBlock, clicando na sequência, no botão <OK>.



Fonte: site mBlock oficial

# Equalizador com sensor de som

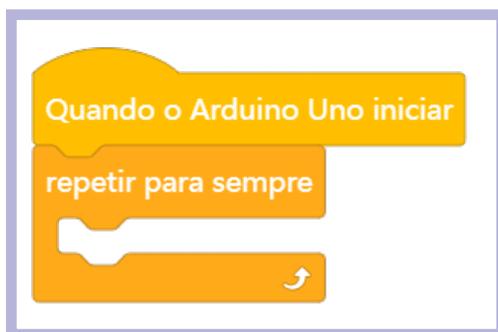
Agora, você poderá acessar os blocos que farão a comunicação com a placa Arduino. Comece clicando em Eventos, na barra de ferramentas, e selecione o bloco **<quando o Arduino Uno começar>**, puxando-o e arrastando-o para o palco de programação.

Figura 9 – Bloco de início do Arduino.



Fonte: site mBlock oficial

Figura 10 – Bloco repetir para sempre.



Fonte: site mBlock oficial

Nesta programação, o objetivo é que o projeto funcione para sempre, portanto, você precisa selecionar na barra de ferramentas, em Controle, o bloco **<repetir para sempre>**. Dentro deste bloco serão inseridos os demais que farão parte da programação e funcionarão em looping (de forma infinita).

Figura 11 – Bloco se... então.

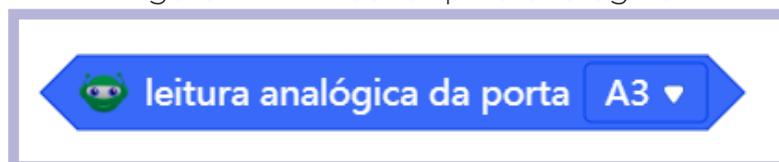


Fonte: site mBlock oficial

O primeiro bloco de programação é uma condicional, pois queremos que se o sensor de som KY-038 captar som, os LEDs se acenderão. Se não houver som, os LEDs ficarão apagados. Portanto, o bloco necessário para esta programação é o **<se ... então...>**

No hexágono após a palavra “se”, deverá ser inserida a variável que será lida pelo Arduino. O módulo sensor de som KY-038 está ligado ao Arduino na porta Analógica A3. Você deve, agora, localizar na barra de ferramentas, na categoria Pin, o bloco **<ler pino analógico>** e alterar o número do círculo para 3 (conforme feito na montagem).

Figura 12 – Bloco ler pino analógico.



Fonte: site mBlock oficial

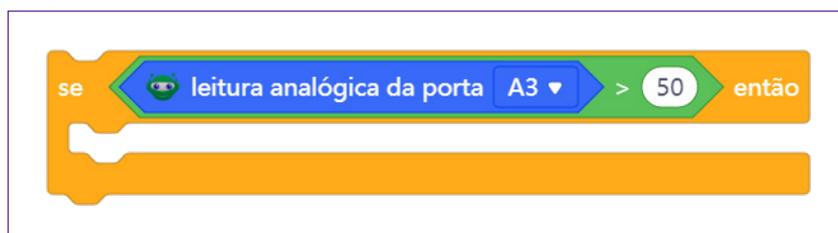


# Equalizador com sensor de som

Essa informação, do pino a ser lido, deverá ser encaixada em outro bloco, presente na categoria <Operadores>, na barra de ferramentas. Selecione o bloco que tem o espaço de informação seguido do sinal de menor, seguido de outro espaço de informação. Arraste esse bloco para o painel de programação, e encaixe o bloco <ler pino analógico> para o primeiro espaço. O segundo espaço, você poderá editar colocando o valor 50.

Repare que esse valor é apenas um exemplo. A porta analógica lê valores de 0 até 1023, e você poderá personalizar o valor inicial e a sensibilidade no *trimpot*, de forma que ele capture sons audíveis. Em Desafios, você poderá explorar mais essa característica da porta analógica desse sensor.

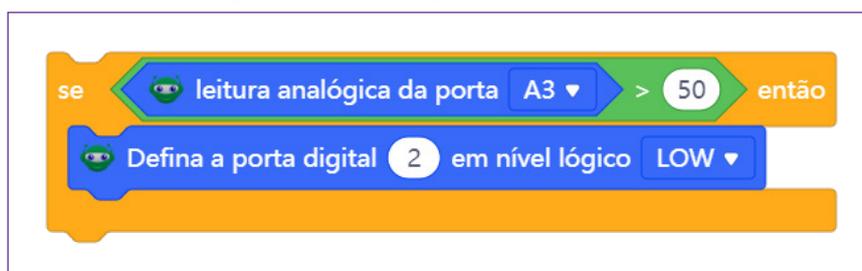
Figura 13 – Bloco variável e sua inserção no bloco se.



Fonte: site mBlock oficial

Agora, vamos programar o primeiro LED que está associado à saída digital 2 do Arduino. Para isso, selecione e arraste o bloco <definir a saída do pino digital ... como >. Substitua o número da porta digital por 2, mantenha a situação <baixo> para o LED. Em termos práticos, significa que até o valor de 50, os LEDs estarão apagados. Você poderá personalizar os intervalos a serem lidos pela porta analógica do sensor de som KY-038, ajustando o *trimpot* ou alterando o numeral dentro do bloco variável (figura 14).

Figura 14 – Programação inicial



Fonte: site mBlock oficial



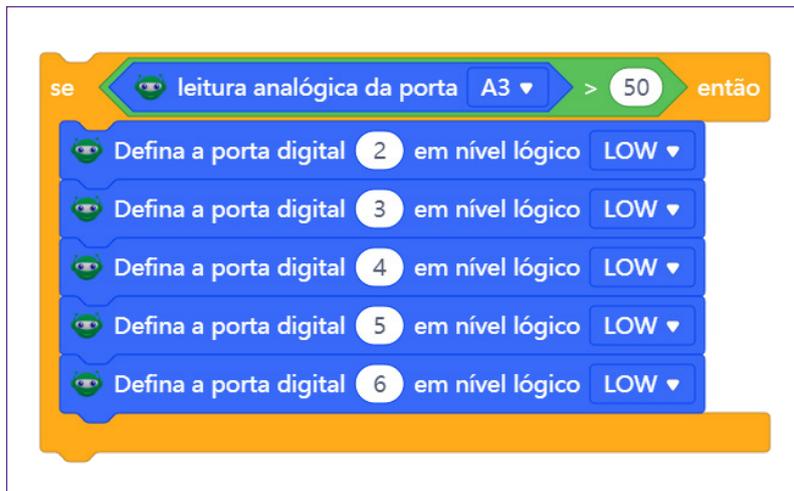
# Equalizador com sensor de som

Duplique o bloco <definir a saída do pino digital...> quatro vezes. Insira um abaixo do outro. Retorne no segundo bloco, e altere o número do pino digital para 3, o seguinte para 4, 5 e 6 (figura 15). Neste primeiro bloco, todos os pinos digitais devem estar como <baixo> (apagados).

Figura 15 - Duplicação

Agora, você duplicará o bloco <se ler pino analógico...> inteiro. Pouse o mouse sobre este bloco e clique com o botão direito do mouse. Selecione Duplicar. Posicione o bloco duplicado abaixo do primeiro (original), dentro do bloco <repetir para sempre>. Selecione e arraste o bloco <ler pino analógico> para fora do conjunto duplicado e depois faça o mesmo com o bloco verde (, figura 16).

do bloco referente aos LEDs



Fonte: site mBlock oficial

Figura 16 - Duplicação do primeiro conjunto de variáveis.



Fonte: site mBlock oficial

# Equalizador com sensor de som

Procure em <Operadores> o bloco que tem a seguinte configuração: .

Arraste este bloco para dentro do espaço presente no bloco <se ... então>. O que programaremos aqui é o intervalo de sons que acionarão os LEDs. Procure em <Operadores> o bloco com o sinal de maior,  e com o sinal de menor . Insira o primeiro

operador, dentro do primeiro espaço e o segundo operador (com o sinal de menor) no último espaço (figura 17).

Figura 17 - Indicação da programação das variáveis.



Fonte: site mBlock oficial

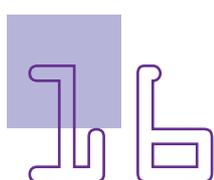
É necessário indicar o pino analógico do sensor de som KY-038 na programação acima. Para tanto, você deve selecionar e arrastar o bloco <ler pino analógico (A)> para o primeiro parâmetro e depois duplique, altere o número para 3 em ambos os blocos (figura 18).

Figura 18 - Indicação da programação das variáveis.



Fonte: site mBlock oficial

Conecte este bloco dentro do espaço do bloco < se... então...>. Altere os números que correspondem a 50 e 50 para 145 e 160 e defina a saída do pino digital 2 como alto (ligado). Esta programação significa que quando o som emitido no ambiente estiver no intervalo numérico indicado, o LED que está ligado na porta 2 do Arduino se acenderá, e apenas ele. Duplique o bloco <se... então> e altere os valores correspondentes a 145 e 160 para 160 e 175 (figura 19).



# Equalizador com sensor de som

Figura 19 – Indicação da programação das variáveis.

Quando o Arduino Uno iniciar

repetir para sempre

se leitura analógica da porta A3 > 50 então

- Defina a porta digital 2 em nível lógico LOW
- Defina a porta digital 3 em nível lógico LOW
- Defina a porta digital 4 em nível lógico LOW
- Defina a porta digital 5 em nível lógico LOW
- Defina a porta digital 6 em nível lógico LOW

se leitura analógica da porta A3 > 145 e leitura analógica da porta A3 < 160 então

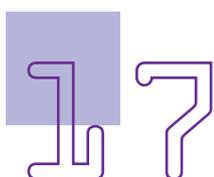
- Defina a porta digital 2 em nível lógico LOW
- Defina a porta digital 3 em nível lógico LOW
- Defina a porta digital 4 em nível lógico LOW
- Defina a porta digital 5 em nível lógico LOW
- Defina a porta digital 6 em nível lógico LOW

se leitura analógica da porta A3 > 160 e leitura analógica da porta A3 < 175 então

- Defina a porta digital 2 em nível lógico HIGH
- Defina a porta digital 3 em nível lógico LOW
- Defina a porta digital 4 em nível lógico LOW
- Defina a porta digital 5 em nível lógico LOW
- Defina a porta digital 6 em nível lógico LOW

Fonte: site mBlock oficial

No próximo passo da programação, você irá duplicar este bloco e alterar os intervalos de 15 em 15, por exemplo 175 e 190 (figura 20). Também altere o status do LED, no pino digital 3, de baixo para alto.



# Equalizador com sensor de som

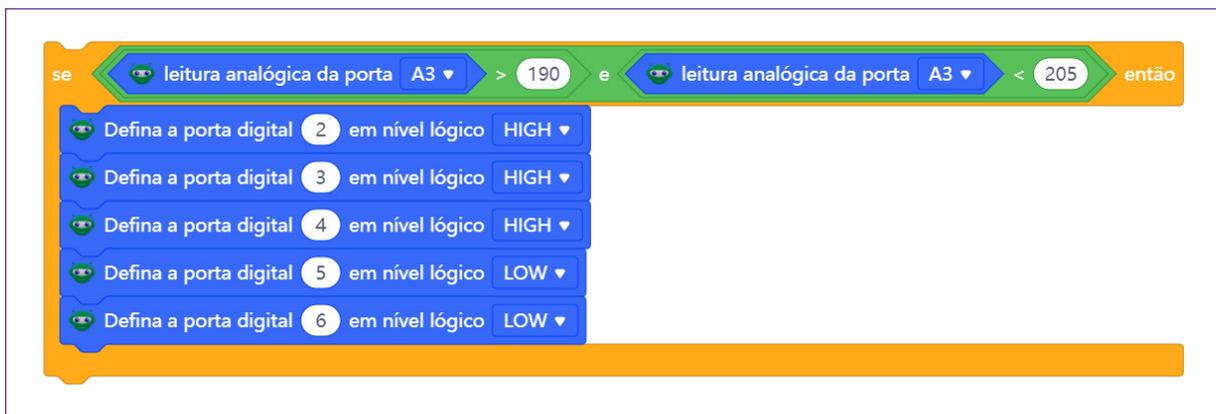
Figura 20 – Programação com a primeira duplicação de intervalo de som.



Fonte: site mBlock oficial

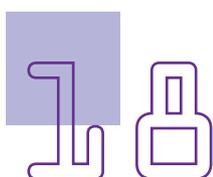
Duplicate novamente o bloco todo de <se... então...> e altere os intervalos de 15 em 15, por exemplo 190 e 205, repetindo no bloco seguinte o último número, 165 e 180. Também altere o status do LED, no pino digital 4, de baixo para alto (figura 21).

Figura 21 – Programação com a duplicação de intervalo de som.



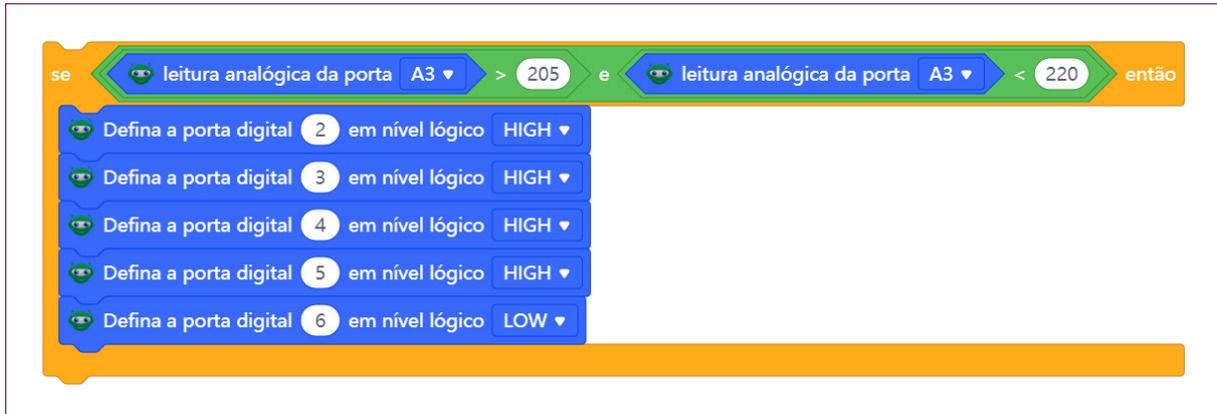
Fonte: site mBlock oficial

Duplicate, novamente, o bloco <se...> e altere a numeração, mantendo o mesmo número da anterior (180) e no seguinte insira 195. Altere também o status de baixo para alto do pino digital 5.



# Equalizador com sensor de som

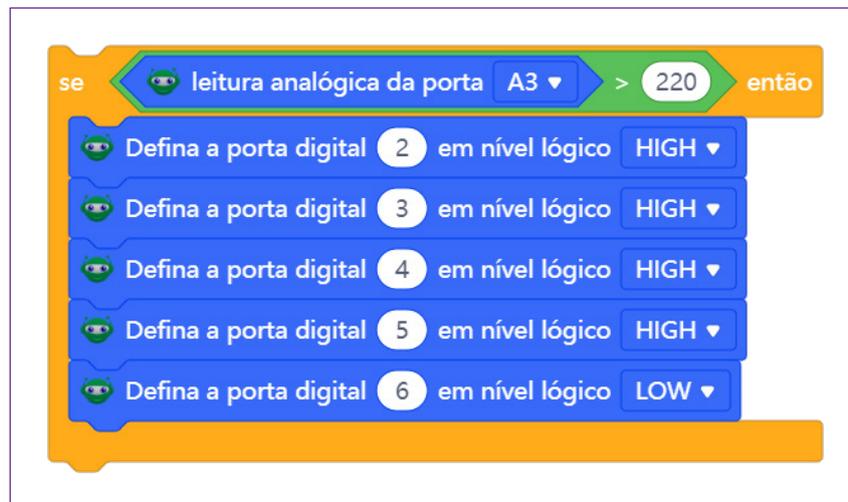
Figura 22 - Programação com a duplicação de intervalo de som.



Fonte: site mBlock oficial

Selecione o bloco <se... então...>, dentro do primeiro espaço de variáveis, selecione em <Operadores> o bloco com o sinal de maior, <50>, edite o segundo espaço, inserindo o número 220. Para o primeiro espaço, você precisa selecionar na categoria <Pin>, o bloco ler pino analógico (A). Insira este bloco no espaço do bloco <se... então...>. Duplicue os blocos de pinos digitais dos blocos anteriores, inserindo-os no bloco (figura 23). Altere o status de todos os pinos digitais (de 2 a 6) para alto.

Figura 23 - Programação com a duplicação de intervalo de som.



Fonte: site mBlock oficial

A programação completa ficou da seguinte forma:

# Equalizador com sensor de som

```

Quando o Arduino Uno iniciar
  repetir para sempre
    se leitura analógica da porta A3 > 50 então
      Defina a porta digital 2 em nível lógico LOW
      Defina a porta digital 3 em nível lógico LOW
      Defina a porta digital 4 em nível lógico LOW
      Defina a porta digital 5 em nível lógico LOW
      Defina a porta digital 6 em nível lógico LOW
    se leitura analógica da porta A3 > 145 e leitura analógica da porta A3 < 160 então
      Defina a porta digital 2 em nível lógico LOW
      Defina a porta digital 3 em nível lógico LOW
      Defina a porta digital 4 em nível lógico LOW
      Defina a porta digital 5 em nível lógico LOW
      Defina a porta digital 6 em nível lógico LOW
    se leitura analógica da porta A3 > 160 e leitura analógica da porta A3 < 175 então
      Defina a porta digital 2 em nível lógico HIGH
      Defina a porta digital 3 em nível lógico LOW
      Defina a porta digital 4 em nível lógico LOW
      Defina a porta digital 5 em nível lógico LOW
      Defina a porta digital 6 em nível lógico LOW
    se leitura analógica da porta A3 > 175 e leitura analógica da porta A3 < 190 então
      Defina a porta digital 2 em nível lógico HIGH
      Defina a porta digital 3 em nível lógico HIGH
      Defina a porta digital 4 em nível lógico LOW
      Defina a porta digital 5 em nível lógico LOW
      Defina a porta digital 6 em nível lógico LOW
    se leitura analógica da porta A3 > 190 e leitura analógica da porta A3 < 205 então
      Defina a porta digital 2 em nível lógico HIGH
      Defina a porta digital 3 em nível lógico HIGH
      Defina a porta digital 4 em nível lógico HIGH
      Defina a porta digital 5 em nível lógico LOW
      Defina a porta digital 6 em nível lógico LOW
    se leitura analógica da porta A3 > 205 e leitura analógica da porta A3 < 220 então
      Defina a porta digital 2 em nível lógico HIGH
      Defina a porta digital 3 em nível lógico HIGH
      Defina a porta digital 4 em nível lógico HIGH
      Defina a porta digital 5 em nível lógico HIGH
      Defina a porta digital 6 em nível lógico LOW
    se leitura analógica da porta A3 > 220 então
      Defina a porta digital 2 em nível lógico HIGH
      Defina a porta digital 3 em nível lógico HIGH
      Defina a porta digital 4 em nível lógico HIGH
      Defina a porta digital 5 em nível lógico HIGH
      Defina a porta digital 6 em nível lógico LOW
  
```

## Desafios

Que tal você alterar a calibragem do módulo sensor de som para ficar mais sensível na captação de sons, a fim de criar um espaço de meditação na sua escola ou que priorize o silêncio?

E se mais LEDs forem inseridos para criar uma sequência maior de sons e acendimento de LEDs?

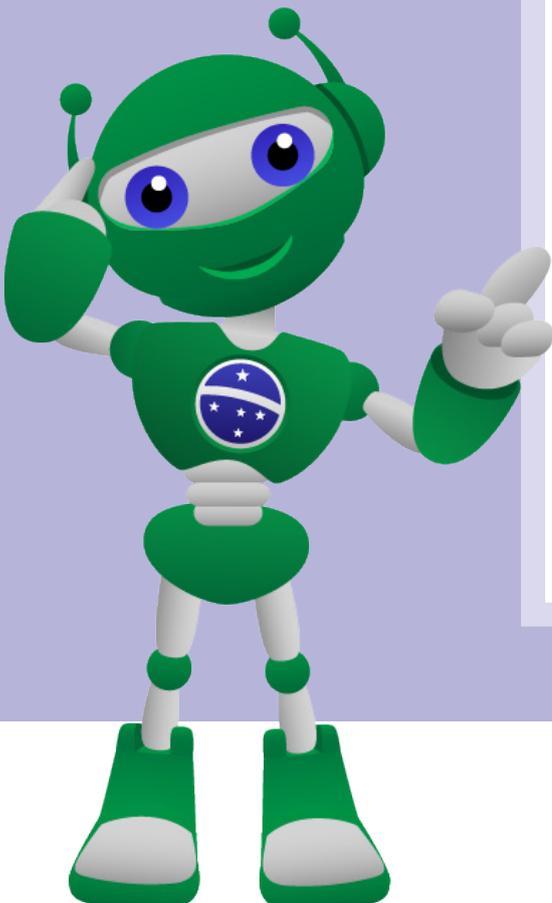
E já pensou em utilizar diferentes músicas para ver esse acionamento?

### Atenção:

Não há uma tabela de referência única que relacione os valores de 0 a 1023 da porta analógica do sensor de som KY-038 com decibéis. Mas você poderá elaborá-la, utilizando aplicativos de celular que quantificam valores sonoros no ambiente, relacionando com o acendimento do LED, relativo à programação.

Alguns aplicativos para Android que detectam sons:

- Decibelímetro;
- Medidor de som;
- Detector de decibéis e ruído.



## E se...?

O projeto não funcionar?

**a.** Verifique se a montagem foi feita corretamente, temos aqui muitos jumpers conectados.

**b.** Verifique se as portas digitais e a porta analógica estão corretamente identificadas na programação.

**c.** Verifique se não está faltando nenhum bloco de programação.

**d.** Verifique a calibragem do *trimpot* do sensor de som KY-038 até que a sensibilidade dele capture o som do ambiente.

## 3. Feedback

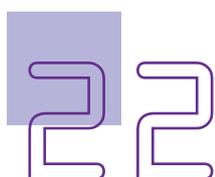
a. Confira, compartilhando seu projeto com os demais colegas, se o objetivo foi alcançado.

b. Analise seu projeto, como você poderia alterá-lo para captar sons mais baixos ou mais altos?

## 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: 12 jul. 2022.

TRIMPOT. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2021. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Trimpot&oldid=62169204>. Acesso em: 08 de nov. 2022.



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

**EQUIPE ROBÓTICA PARANÁ**

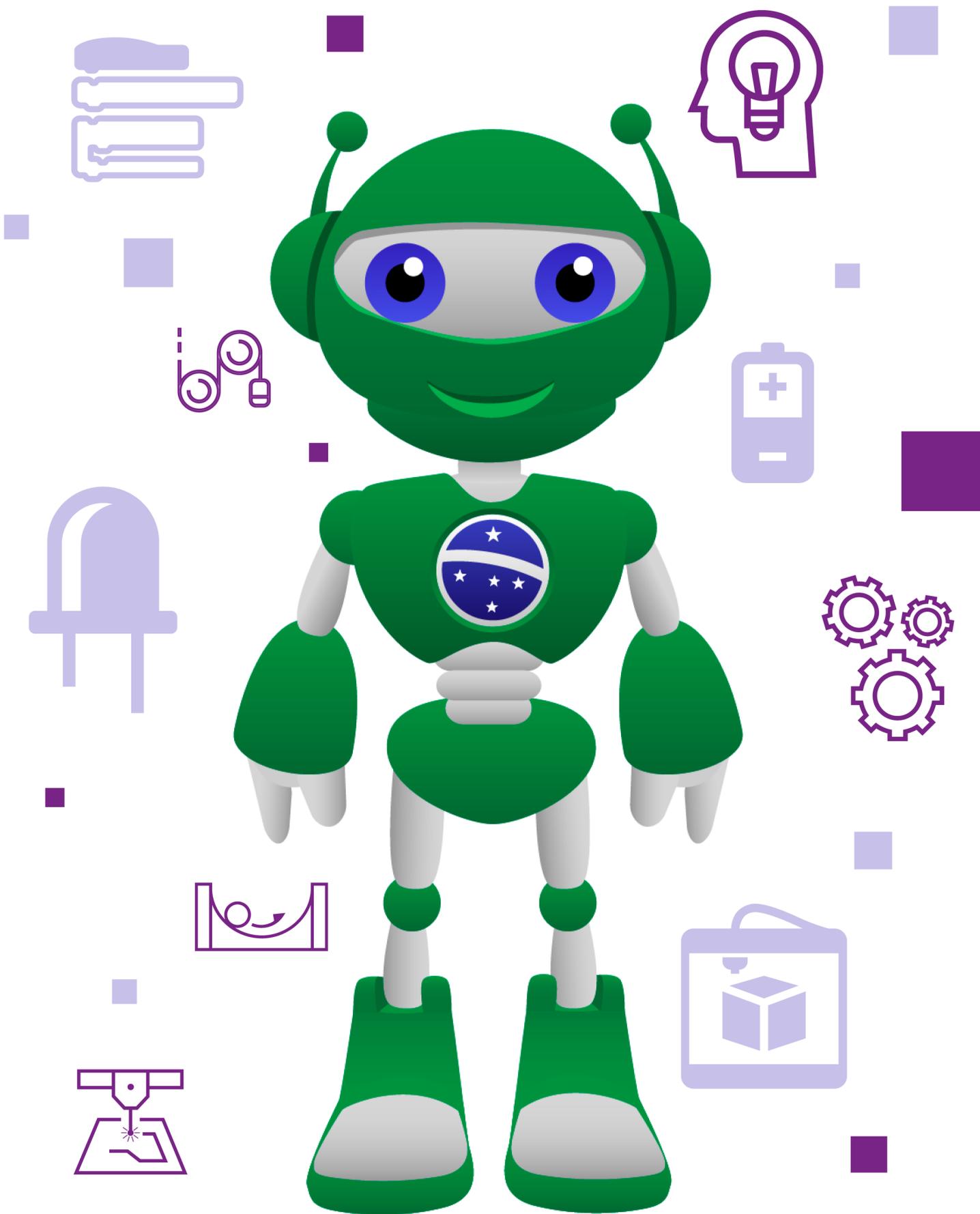
Andrea da Silva Castagini Padilha  
Cleiton Rosa  
Darice Alessandra Deckmann Zanardini  
Edgar Cavalli Junior  
Edna do Rocio Becker  
José Feuser Meurer  
Marcelo Gasparin  
Michele Serpe Fernandes  
Michelle dos Santos  
Orlando de Macedo Junior  
Roberto Carlos Rodrigues

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