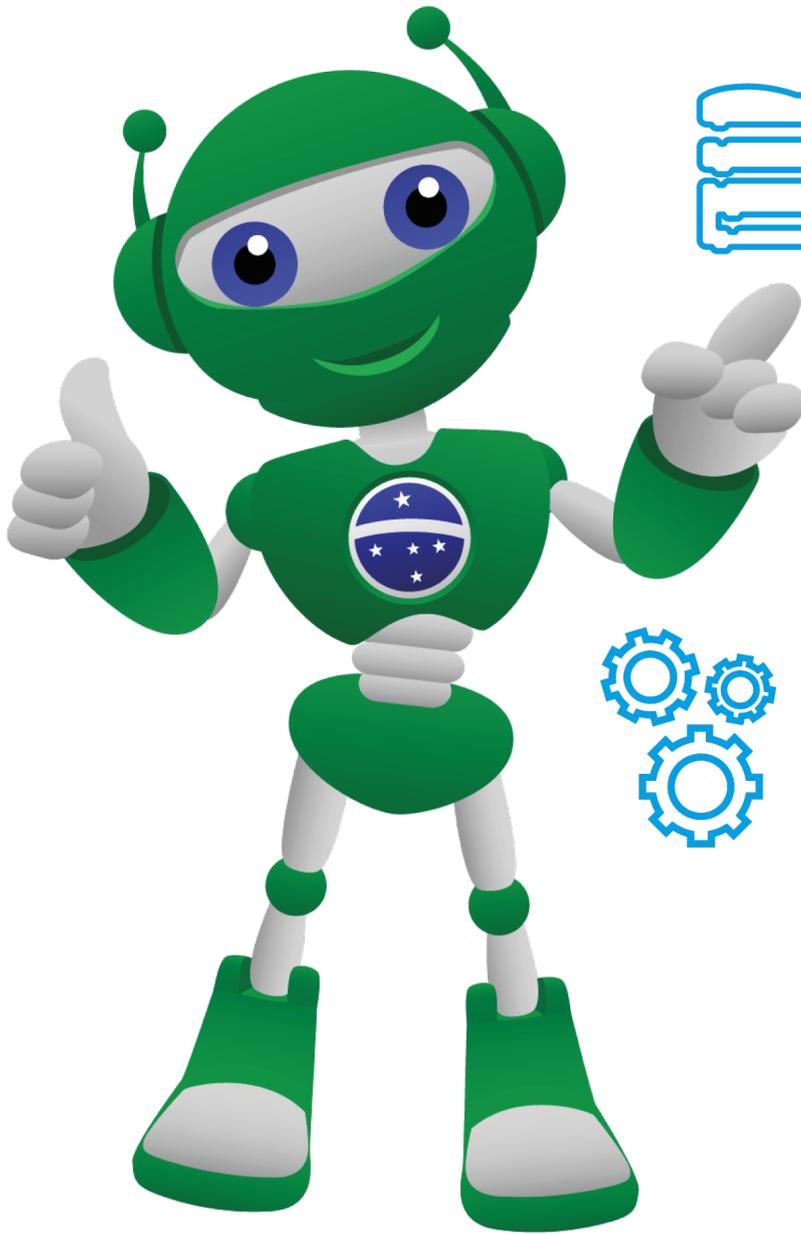


ROBÓTICA

Primeiros Passos

Módulo 1



AULA 38

Interação entre Mascotes

Diretoria de Tecnologia e Inovação

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Renato Feder

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Andre Gustavo Souza Garbosa

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Darice Alessandra Deckmann Zanardini

Validação de Conteúdo

Cleiton Rosa

Revisão Textual

Adilson Carlos Batista

Normalização Bibliográfica

Ricardo Hasper

Projeto Gráfico e Diagramação

Edna do Rocio Becker

Ilustração

Jocelin Vianna

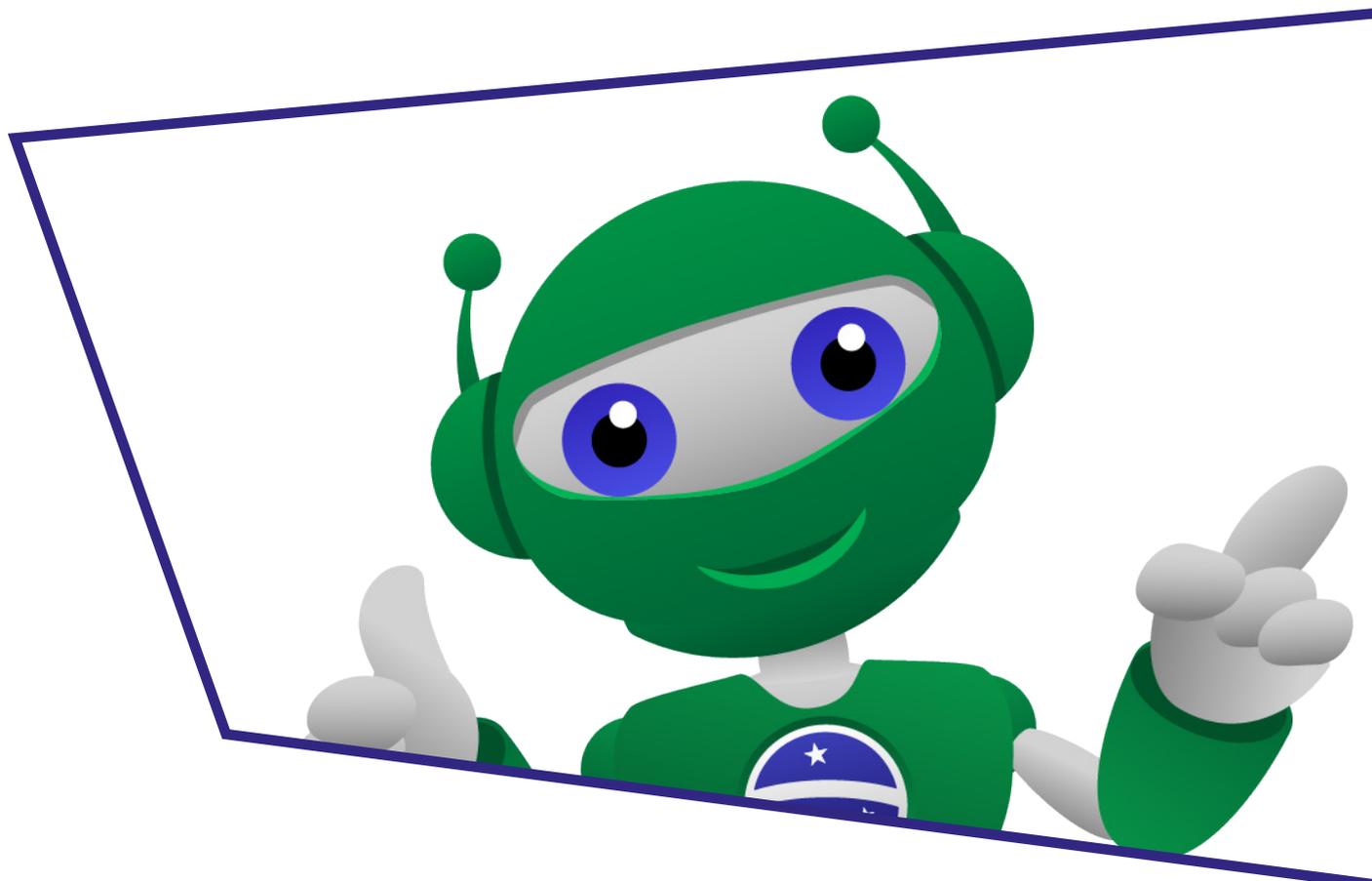
Modelagem Blender

Cleiton Rosa, Darice Alessandra Deckmann Zanardini,
Roberto Carlos Rodrigues

2022

SUMÁRIO

Introdução	2
Objetivos desta Aula	2
Competências Gerais Previstas na BNCC	3
Habilidades do Século XXI a Serem Desenvolvidas	4
Lista de Materiais	4
Roteiro da aula	5
1. Contextualização	5
2. Conteúdo	6
3. Feedback	34
Referências	35
Anexo	36





Introdução

Na última aula, desenvolvemos um projeto que simula a comunicação da nossa mascote BIT pelo acionamento de LEDs, representando seus olhos, e buzzer, representando sua “fala”. Agora, chegou o momento de desenvolvermos o projeto que simula uma interação entre duas mascotes, como se ambas estivessem em uma conversa por código.



Objetivos desta Aula

- Atribuir sons à mascote com o buzzer passivo;
- Sincronizar o acendimento de LEDs ao som que a mascote emite pelo buzzer passivo;
- Simular uma interação simultânea entre robôs;
- Sincronizar a execução de dois projetos;
- Conhecer o código Morse.





Competências Gerais Previstas na BNCC

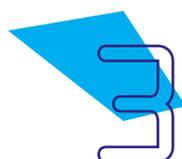
[CG02] - Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas)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





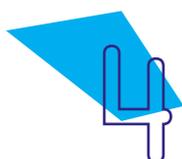
Habilidades do Século XXI a Serem Desenvolvidas

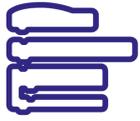
- Pensamento crítico;
- Afinidade digital;
- Resiliência;
- Resolução de problemas;
- Colaboração;
- Comunicação;
- Criatividade.



Lista de Materiais

- Papel;
- Impressora;
- Canetas ou lápis coloridos;
- Fita adesiva;
- 01 Arduino Uno R3;
- 01 Buzzer passivo;
- 02 LEDs;
- 02 Resistores 220 Ohms;
- 02 Jumpers macho-fêmea;
- 06 Jumpers macho-macho;
- 01 Notebook;
- Software mBlock.





Roteiro da Aula

1. Contextualização:

No decorrer dos últimos tempos, a Inteligência Artificial (AI) tem se desenvolvido em áreas na qual sistemas, robôs ou outros dispositivos não apenas realizam atividades repetitivas, como a ideia inicial de robôs que tivemos, mas também desenvolvem a capacidade de análise e tomada de decisões a partir de dados, algoritmos e sistemas de aprendizado, o que amplia seus conhecimentos, aprendendo, inclusive, comportamentos e demais características humanas por meio de Machine Learning (aprendizagem automática ou aprendizado de máquina, em inglês), ampliando suas capacidades através de Deep Learning (aprendizagem profunda, em inglês) e melhorando a compreensão de textos através de PLN (Processamento de Linguagem Natural).

A partir de seus recursos, a Inteligência Artificial pode ser aplicada em três áreas principais:

- Para otimizar processos e simplificar tarefas do cotidiano quando realiza tarefas programadas e toma decisões baseadas em dados;
- Para atuar em sistemas mais complexos que envolvem a capacidade de tomada de decisões coletivas através de sistemas interconectados com foco na solução dos mais variados problemas;
- Para criar, nos dispositivos, interfaces digitais mais humanizadas e inteligentes com comportamento semelhante ao dos seres humanos.

Vamos a um exemplo desta última área principal? Você já visualizou ou teve experiência em sua casa com alguma conta de rede social ou página de internet, como de lojas ou instituições, que disponibiliza o recurso “Fale Conosco”, em que é possível conversar com um atendente virtual, por vezes bem simpático? Se sim, você conversou com um chatterbot (robô para conversas), uma das principais áreas em que a Inteligência Virtual é aplicada!

Partindo desta ideia de conversa, nesta aula veremos a simulação de interação entre dois robôs, os quais expressarão, através de seus LEDs e buzzer, uma mensagem para você e seus colegas decifrarem!





Para saber mais...



[Dança das abelhas é aplicada na comunicação entre robôs](#)

Com base nessa comunicação entre os insetos, os robôs entendem a direção e distância do objetivo - e acertam em 90% das vezes.



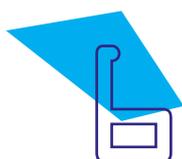
[Como este estudo ensinou um robô a ter “empatia”](#)

Em teste feito na Universidade Columbia, nos EUA, a máquina aprendeu a prever as “falhas” de um colega.

2. Conteúdo:

Hoje, ampliaremos nosso projeto iniciado na **Aula 37 - A mascote se comunica**, reunindo duas equipes para que as mascotes se comuniquem entre si por meio de um sistema codificado de comunicação. Para isso, combinaremos, além de duas montagens do nosso protótipo do B1T com buzzer passivo e LEDs, a programação de cada mascote - nesta aula, você e seus colegas trabalharão entre equipes para que cada B1T emita uma palavra.

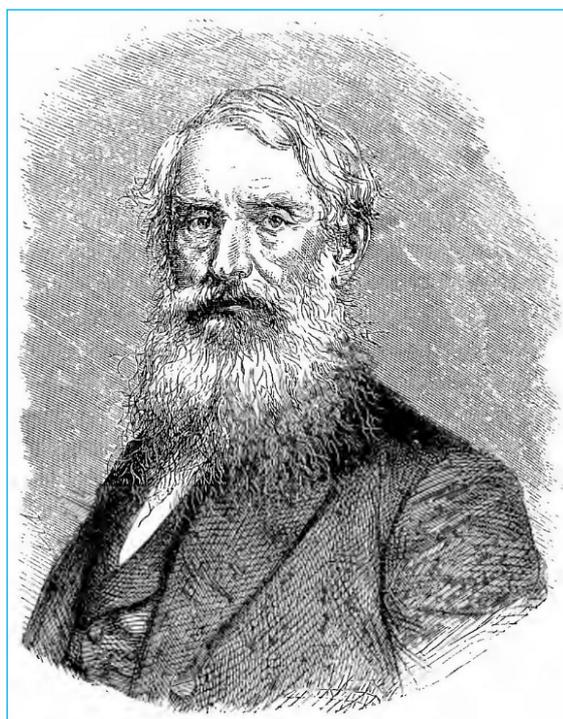
Para a comunicação de nossas mascotes, é importante focarmos no sincronismo entre as programações. Afinal, em conversas educadas um respeita o momento de fala do outro, não é mesmo?



Você sabia que há uma forma de comunicação conhecida internacionalmente que não utiliza palavras faladas, mas a emissão de sinais sonoros ou visuais que correspondem a letras e caracteres?

O código Morse foi desenvolvido em 1835 pelo inventor, físico e pintor Samuel Morse para a transmissão de mensagens por sinais curtos ou longos, através dos pulsos elétricos do telégrafo, aparelho que usa a corrente elétrica para enviar pulsos ou sinais curtos ou longos, gerando mensagens codificada em pontos ou traços. O telégrafo não é mais utilizado, mas o código Morse sim, seja através do acionamento de lanterna, batidas em uma superfície ou um apito!

Figura 01: Samuel Morse



Fonte: Wikimedia Commons

Cada combinação dos sinais pode significar uma letra, número ou sinal de pontuação, conforme indicado na tabela abaixo em que o ponto representa um sinal curto e o traço um sinal longo nesta equivalência de letras, números e sinais de pontuação em código Morse.

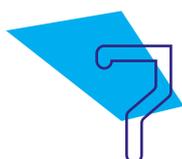
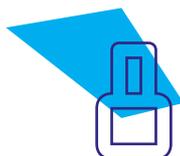


Tabela 1: Equivalência de letras, números e sinais de pontuação em código Morse

Letras	Código Internacional	Números	Código internacional	Pontuações comuns	Código internacional
A	.-	1	.-----	Ponto [.]	.-.-.-.-
B	-...	2	..---	Vírgula [,]	---...-
C	-.-.	3	...--	Interrogação [?]	..-...
D	-..	4-	Apóstrofo [']	.-----
E	.	5	Exclamação [!]	-.-.-.-
F	..-.	6	-.....	Barra [/]	-...-.
G	--.	7	--...	Parênteses [(]	-.---.
H	8	---..	Parênteses [)]	-...--
I	..	9	----.	E comercial [&]	.-....
J	.---	0	-----	Dois pontos [:]	---...
K	-.-			Ponto e vírgula [;]	-.-.-.-
L	.-...			Igual [=]	-...-
M	--			Hífen [-]	-.....-
N	-.			Linha baixa [_]	..---.-
O	---			Aspas [“”]	.-...-.
P	.---.			Cifrão [\$]	...-...-
Q	--.-			Arroba [@]	..-...
R	.-.				
S	---				
T	-				
U	..-				
V	...-				
W	.-.-				
X	-...-				
Y	-.--				
Z	--..				



Quando uma mensagem é enviada por código Morse, através da emissão de sinais sonoros ou visuais, é preciso prestar atenção na sequência dos sinais para identificar as letras, números ou caracteres correspondentes. Quer ver um exemplo bastante utilizado em situações de perigo? O sinal SOS, internacionalmente conhecido como um pedido de socorro, em código Morse é expresso por três pulsos curtos para o **S** e três pulsos longos para o **O** (... --- ...).

O código Morse pode ser utilizado também com a emissão de luz, como a de uma lanterna, no lugar de som! Está é uma forma de comunicação, por exemplo, a longa distância, como em uma ilha, ou quando você não pode fazer barulho.



Para saber mais...

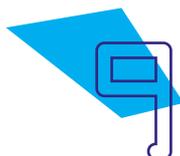


[SOS NÃO É O QUE PARECE! Entenda o Código Morse](#)
(10min53)

Após a montagem, caso seja necessário repeti-la, e programação de cada um dos protótipos da mascote, retornaremos à tabela **Equivalência de letras, números e sinais de pontuação em código Morse** para você decifrar a mensagem expressa pelas duas mascotes.

A montagem indicada abaixo é a mesma que realizamos na **Aula 37 - A mascote se comunica**. Repita-a, caso seja necessário e você não esteja com o protótipo em mãos, lembrando que, por estarem trabalhando entre equipes nesta aula, serão necessários **dois protótipos da mascote BIT** para simularmos a comunicação entre eles.

Vamos relembrar a montagem do protótipo da mascote?



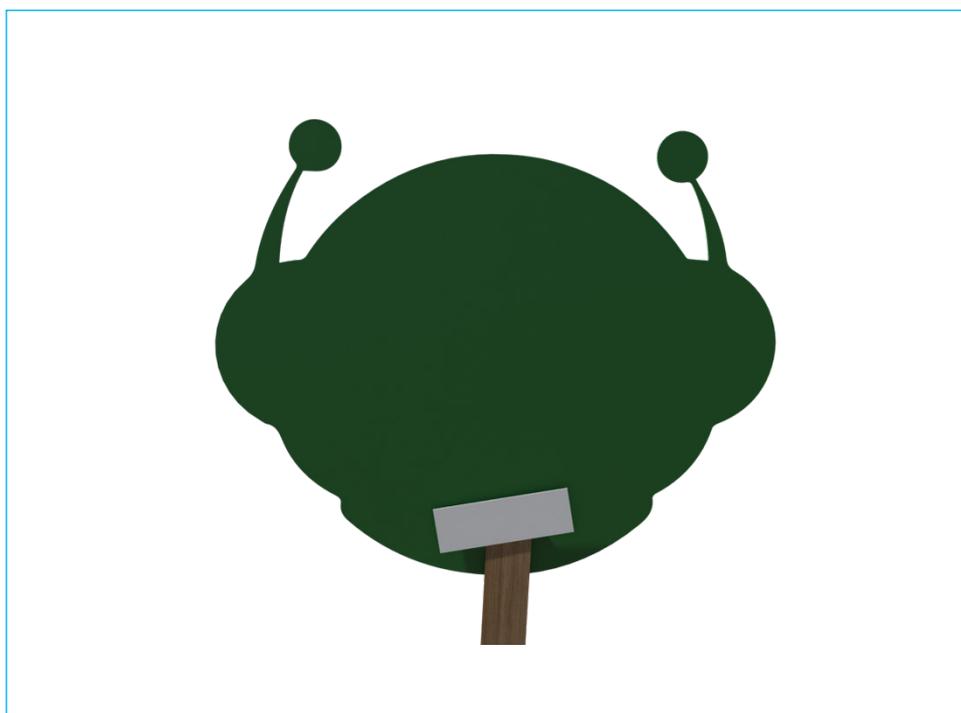
No **Anexo 01** desta aula, encontramos o rosto do B1T. Recorte-o para fazer as conexões e cole um palito ou lápis no verso, que será seu suporte.

Figura 02 - Rosto do B1T (frente)



Fonte: SEED/DTI/CTE

Figura 03 - Rosto do B1T (verso)



Na região dos olhos, insira dois LEDs.

Figura 04 - Inserção dos LEDs nos olhos do BIT (frente)

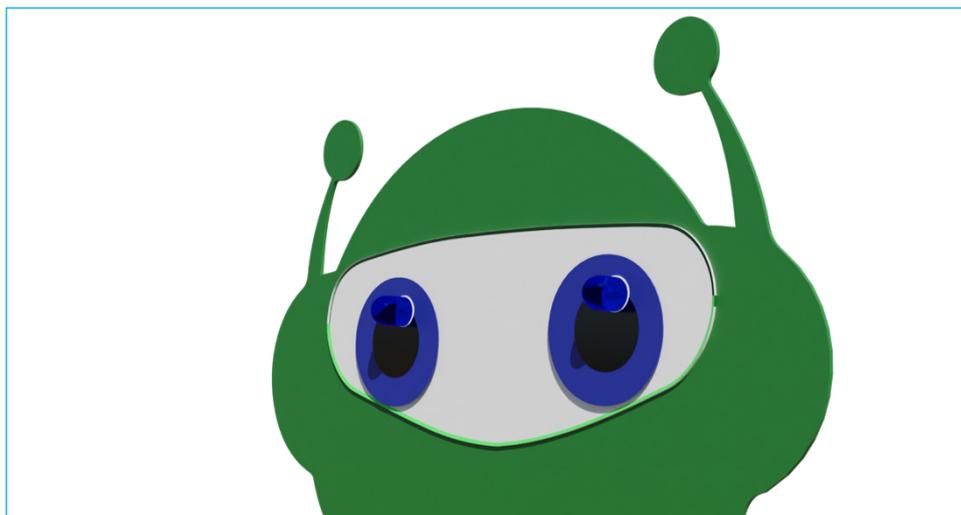
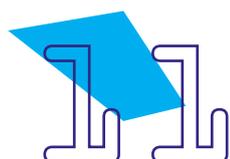
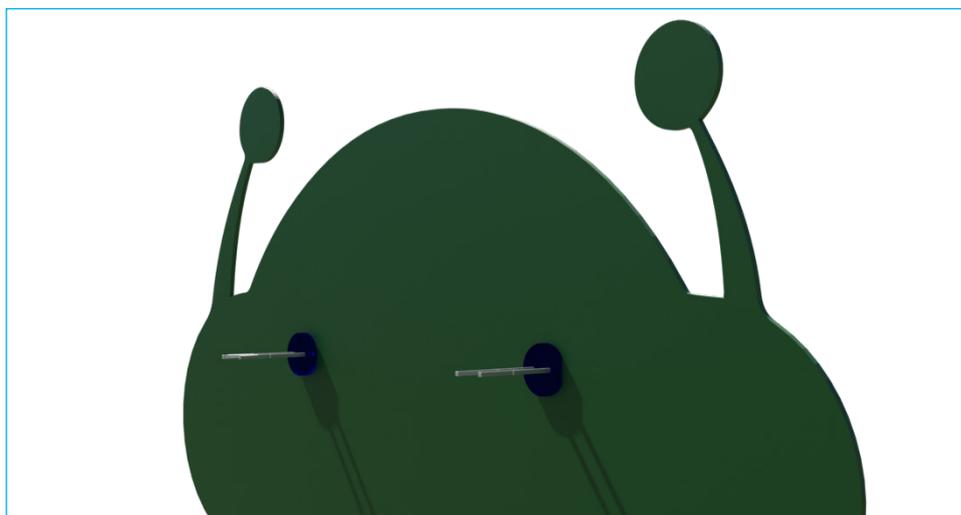


Figura 05 - Inserção dos LEDs nos olhos do BIT (detalhe verso)

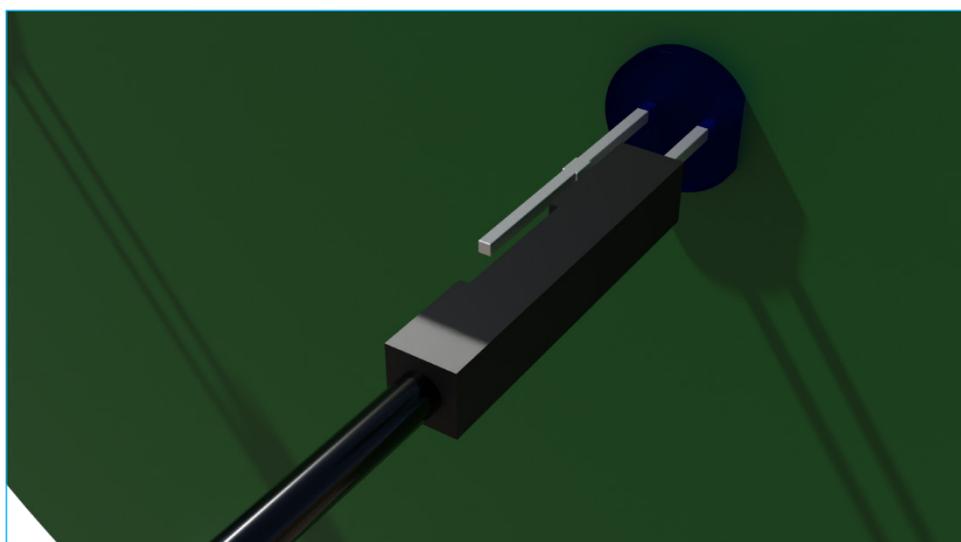


Agora, vamos conectar os LEDs ao Arduino, iniciando pelo polo negativo do LED (lado chanfrado). No pino do LED, conecte um jumper fêmea-fêmea e, se necessário, envolva-o com uma fita adesiva para fixar a conexão.

Figura 06 - Conexão dos jumpers fêmea-fêmea aos polos negativos do LED



Figura 07 - Conexão dos jumpers fêmea-fêmea aos polos negativos do LED (detalhe do lado chanfrado)



Na outra extremidade do jumper fêmea-fêmea, conecte um resistor de 220 Ohms, fixando-o também com fita, visto que a haste do resistor é fininha. A outra extremidade de cada resistor, conecte nas portas GND do Arduino, localizadas na área “Power”.

Figura 08 - Visão superior da conexão de cada resistor 220 Ohms entre jumper fêmea-fêmea e portas GND do Arduino

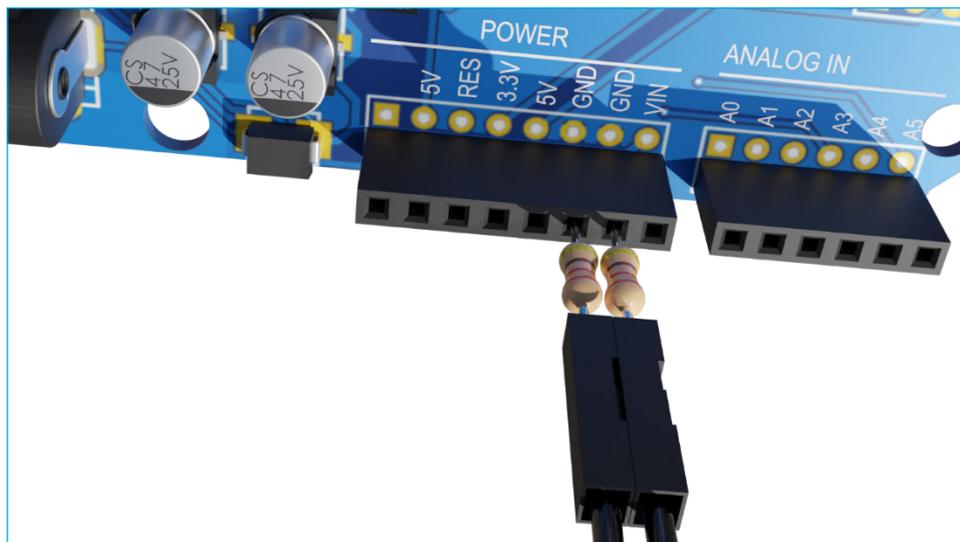
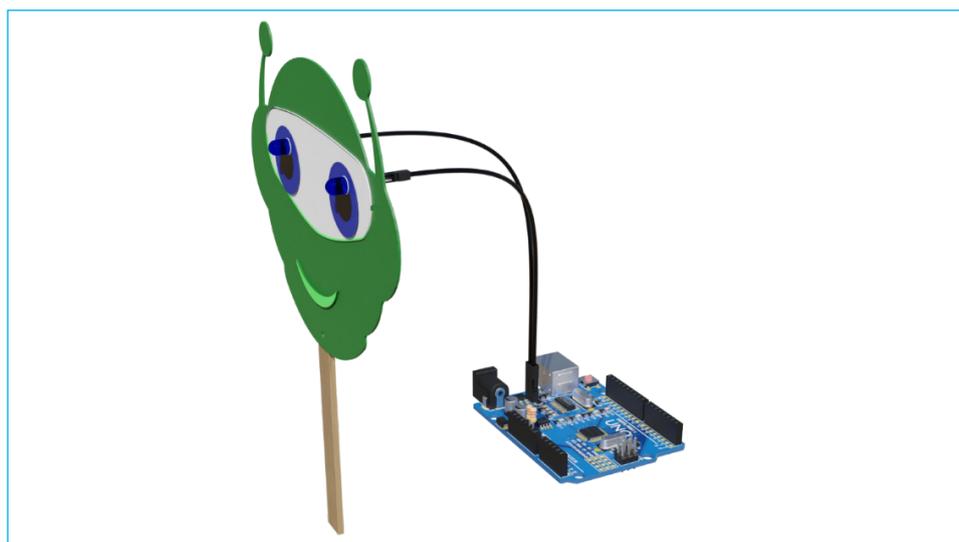


Figura 09 - Conexão resistor 220 Ohms entre jumper fêmea-fêmea e Arduino



Vamos, agora, fazer a conexão dos polos positivos dos LEDs às portas digitais do Arduino. Conecte em cada haste positiva dos LEDs o jumper fêmea-fêmea e, na outra extremidade deste jumper, o jumper macho-macho. Se necessário, passe uma fita adesiva ao redor das áreas plásticas para que o pino não se solte.

Figura 10 - Conexão dos jumpers fêmea-fêmea aos polos positivos dos LEDs

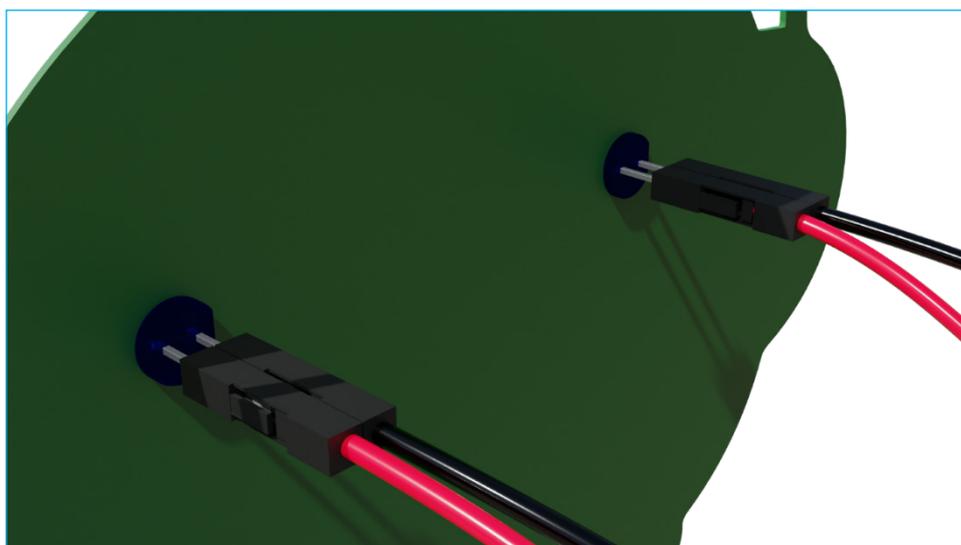
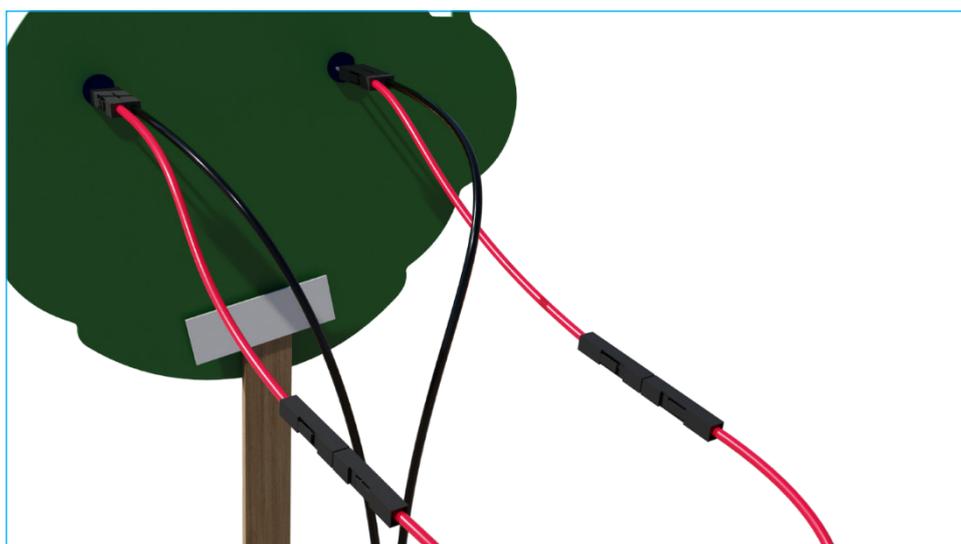


Figura 11 - Conexão dos jumpers fêmea-fêmea aos jumpers macho-macho



Por fim, conecte cada polo positivo do LED, através do jumper macho-macho, à porta digital 8 e 9 do Arduino.

Figura 12 - Conexão dos jumpers macho-macho às portas digitais 8 e 9 do Arduino

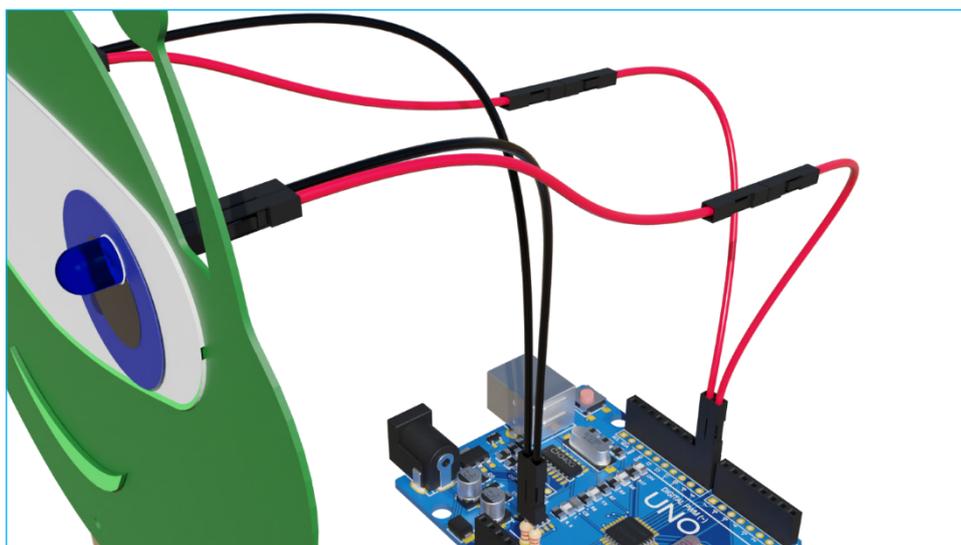
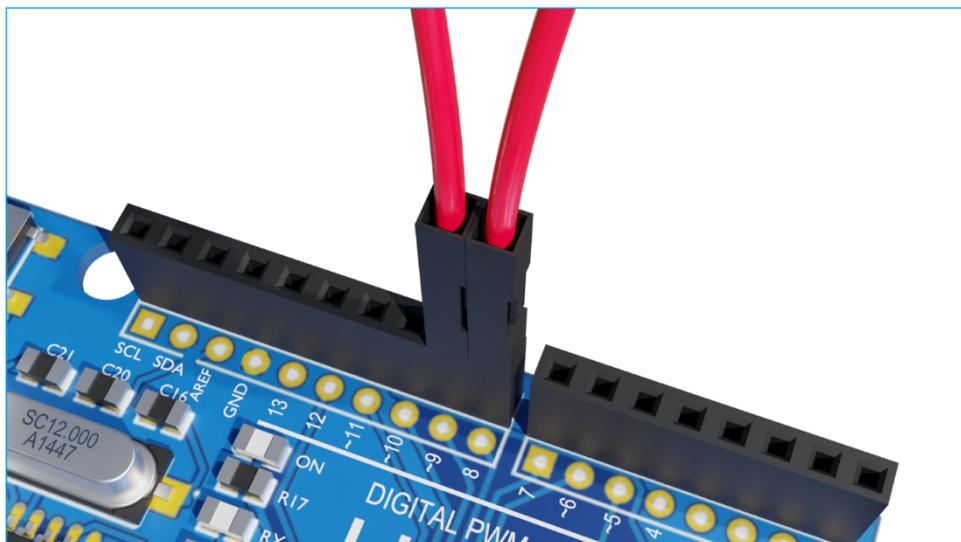


Figura 13 - Conexão dos jumpers macho-macho às portas digitais 8 e 9 do Arduino (detalhe)



Quanto à conexão do buzzer, neste projeto ela também será feita diretamente no B1T, ao contrário da última aula, em que o conectamos diretamente ao Arduino. Para isso, fixe o buzzer no verso do B1T, no lugar que você julgar mais adequado, deixando os pinos do buzzer expostos para as conexões. Aqui, fixamos, como sugestão, no palito, próximo à boca.

Figura 14 - Encaixe e conexão do buzzer à mascote B1T

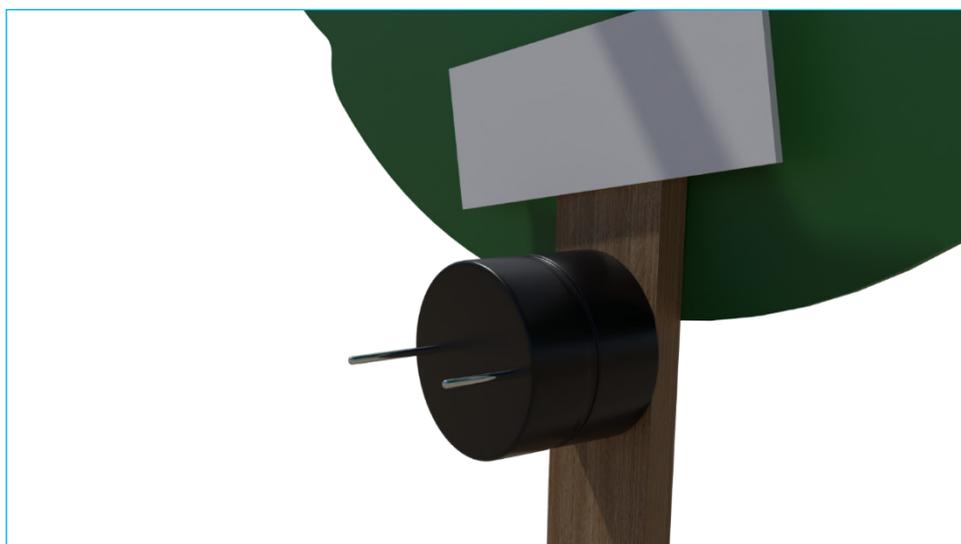
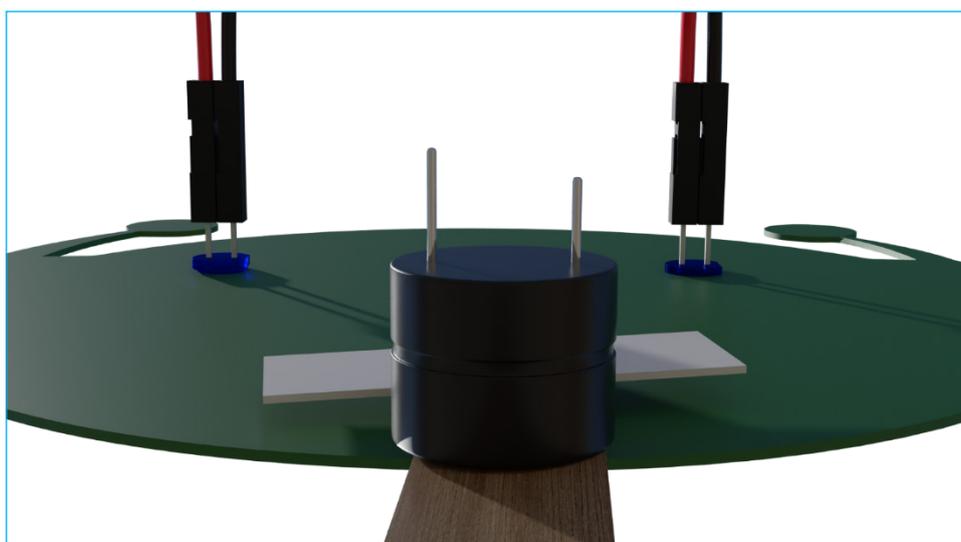


Figura 15 - Detalhe dos pinos do buzzer



Assim como fizemos com os pinos positivos dos LEDs, conecte um jumper fêmea-fêmea em cada pino do buzzer e um jumper macho-macho na extremidade vazia do jumper fêmea-fêmea.

Figura 16 - Conexão dos jumpers ao buzzer (detalhe)

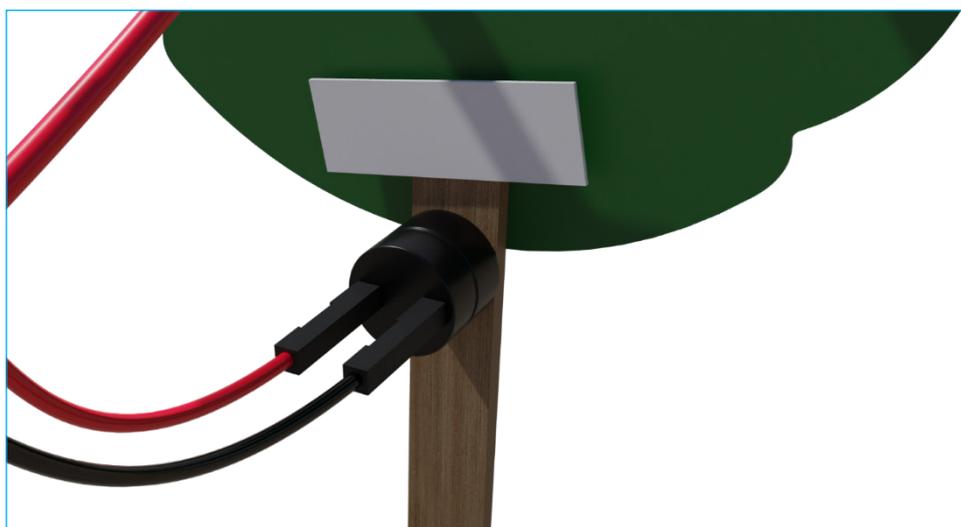
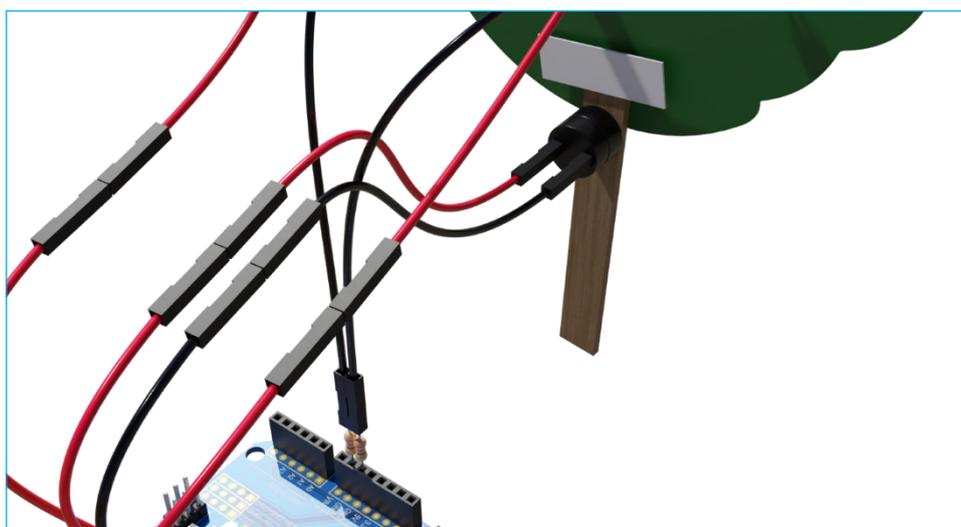


Figura 17 - Conexão dos jumpers ao buzzer



Observe a posição do buzzer, com o lado indicativo da conexão 5V (sinal positivo e haste mais longa) para conectar seus jumpers à porta digital 11 do Arduino, lembrando que nas portas digitais 8 e 9 já estão conectados os jumpers dos polos positivos dos LEDs.

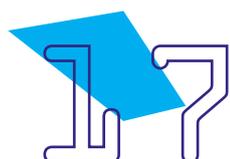
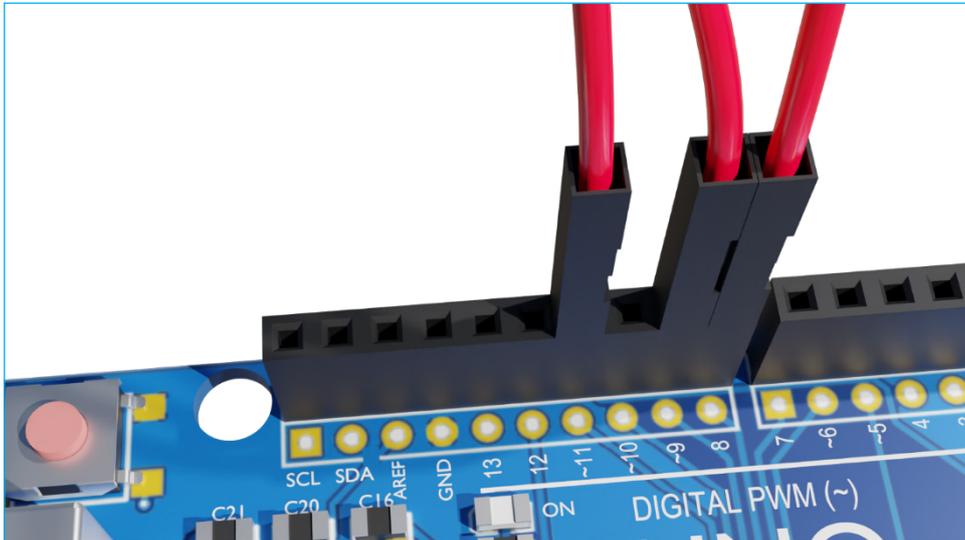
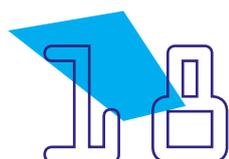
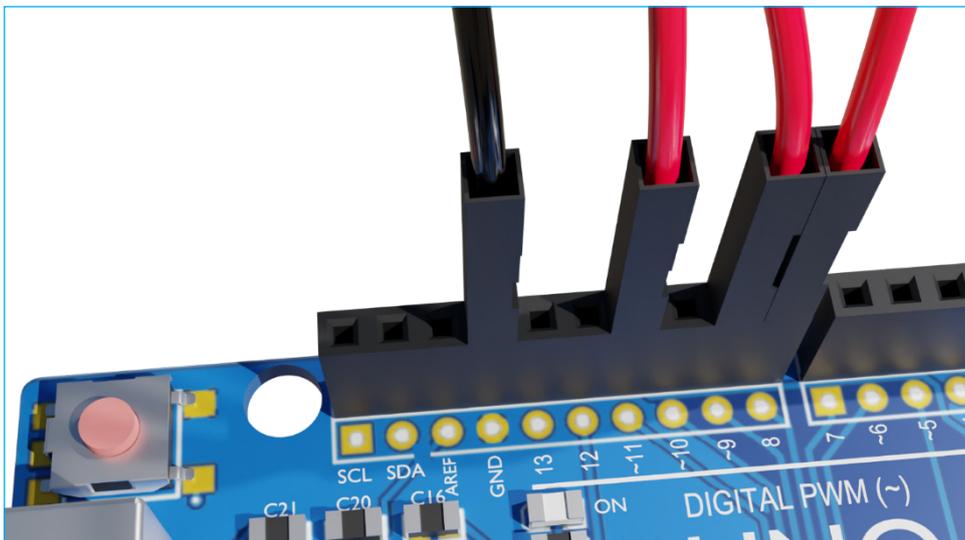


Figura 18 - Conexão do buzzer à porta 11 do Arduino



Em seguida, conecte os jumpers ligados ao lado sem sinalização do buzzer e com a haste mais curta à outra porta GND do Arduino, localizada no mesmo lado das portas digitais.

Figura 19 - Conexão do buzzer à porta GND do Arduino



Finalizadas as conexões, as duas mascotes ficarão assim:

Figura 20 - Visão geral das conexões para cada BIT (frente)

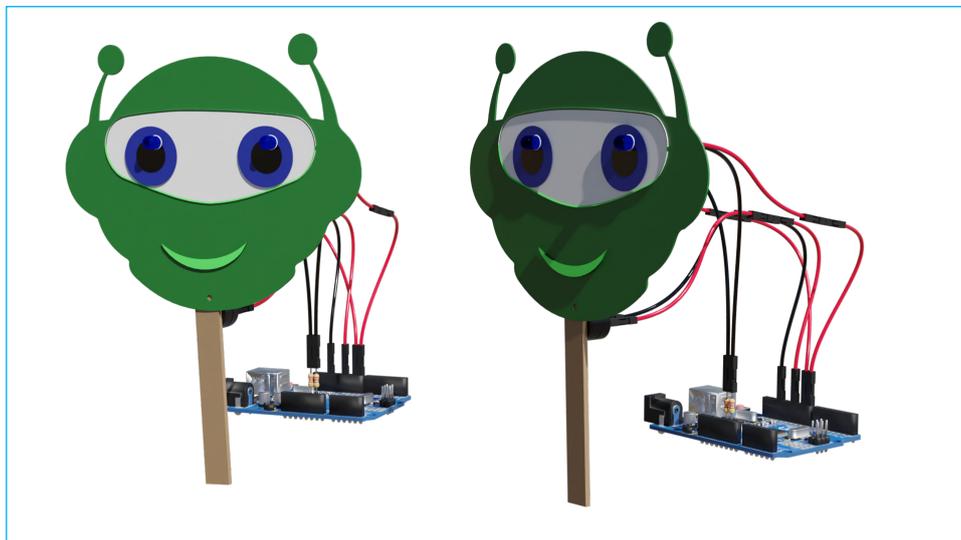
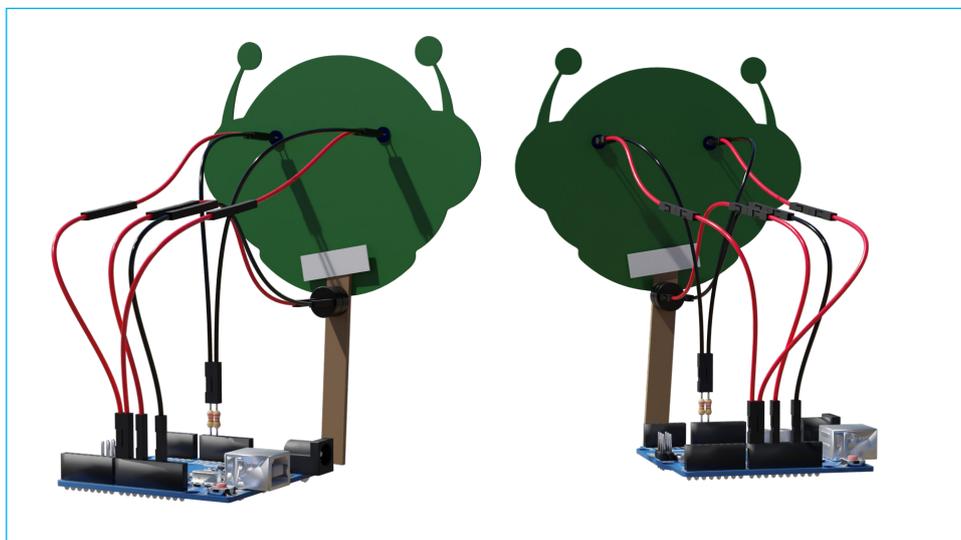


Figura 21 - Visão geral das conexões para cada BIT (verso)



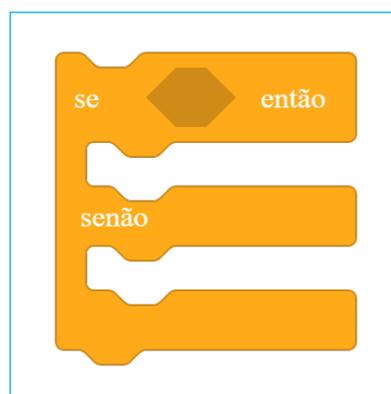
Fonte: SEED/DTI/CTE

Atenção: para evitar danos ao projeto por conexões erradas, atente-se à polaridade de cada componente! Para os LEDs, o lado chanfrado, ou com um corte reto, corresponde ao polo negativo, ligado ao GND do Arduino, e o lado arredondado, curvo, corresponde ao polo positivo, ligado às portas digitais do Arduino. No caso do buzzer, o lado conectado à porta 11, com pino longo, é indicado pelo sinal positivo (+) e o lado conectado à porta GND, com pino curto, não possui indicação de sinal.

Vamos programar os BITs? Na nossa programação, utilizaremos um sinal com delay de 0.3 segundos para simbolizar os sinais curtos e um sinal de 0.9 segundos para simbolizar os sinais longos. Cada mascote terá uma programação específica e, no momento de execução do projeto, respeite a ordem de cada uma delas!

Observe que, para otimizar nossa programação, experimentaremos usar um bloco novo: <se X então ..., senão ...>. Este bloco faz que uma determinada ação ocorra enquanto outra acontece. No caso das programações desta aula, faz com que os LEDs ligados às portas digitais 8 e 9 sejam acionados enquanto ocorrer a leitura da porta digital 11, na qual conectamos o buzzer passivo. Lembrando que utilizamos as mesmas conexões em cada uma das mascotes.

Figura 22 - Bloco “se X então... senão...”



Fonte: mBlock

Outro bloco que você pode utilizar para otimizar nosso código é o **<repetir X>**, já aplicado em programações anteriores. Como o código Morse utiliza a repetição de sinais para algumas letras, este é um recurso para minimizar o número de linhas do nosso código de programação. Que tal experimentar usá-lo quando você visualizar a programação?

Vamos agora à programação de cada mascote, iniciando pela **programação 1, da mascote que iniciará o diálogo**.

Programação 1 - Mascote que iniciará o diálogo (Equipe 1)

Iniciaremos esta programação definindo o acendimento dos LEDs quando ocorrer a leitura da porta digital 11:

Figura 23 - Início da programação para acionamento dos LEDs da mascote 1



A sequência da programação será inserida **abaixo** deste bloco de controle, letra a letra.



Como as mascotes transmitirão uma mensagem, faremos a programação de cada letra, com um delay de 0.6 segundos entre cada uma, sem revelar neste momento quais serão - o desafio para você e seus colegas é descobrir qual mensagem está sendo “dita”, decifrando o código Morse!

Figura 24 - Primeira letra da programação 1



Figura 25 - Segunda letra da programação 1

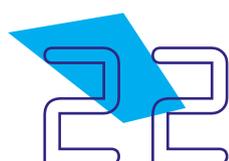


Figura 26 - Terceira letra da programação 1



Figura 27 - Quarta letra da programação 1



Figura 28 - Quinta letra da programação 1



Figura 29 - Sexta letra da programação 1



Figura 30 - Sétima letra da programação 1

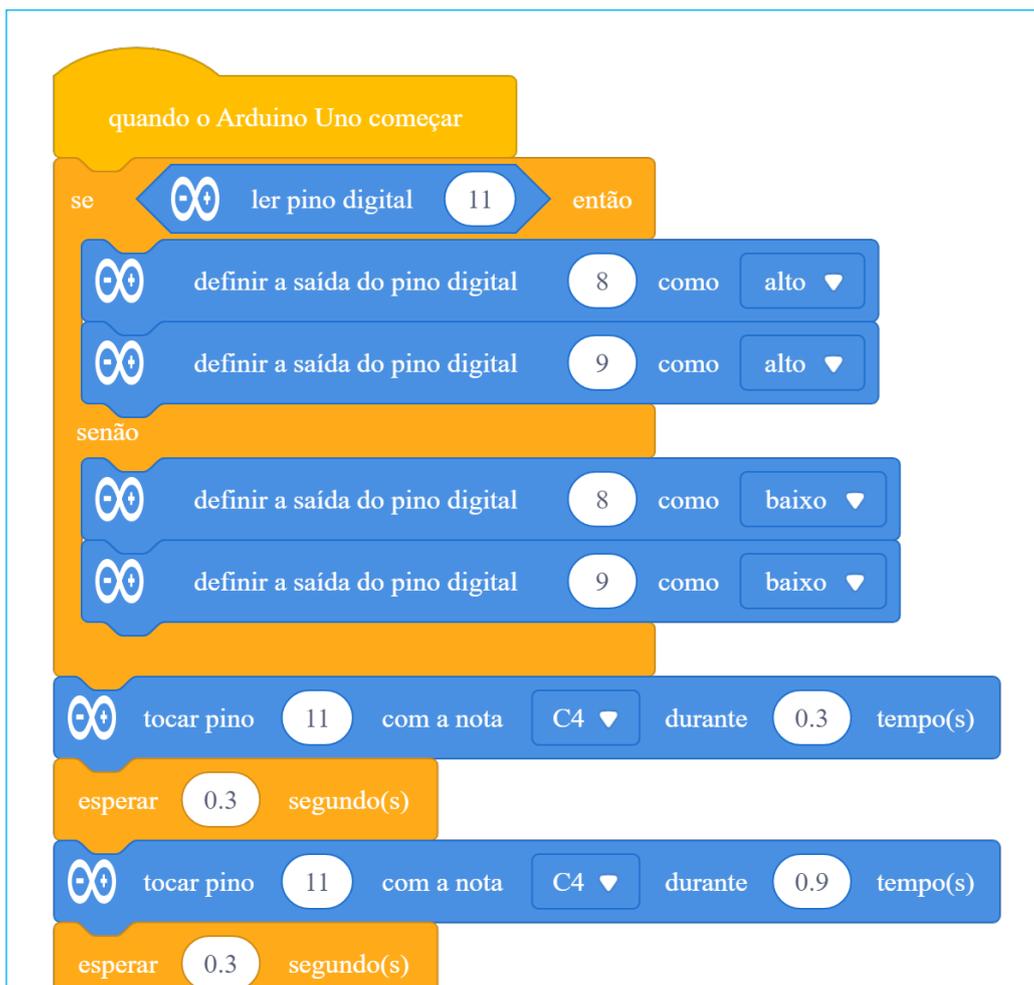


Figura 31 - Oitava letra da programação 1



A programação completa da mascote 1 fica assim:

Figura 32 - Programação 1 completa para a mascote 1



The image displays a sequence of 12 Scratch code blocks arranged vertically. Each block consists of a blue 'tocar pino' (play note) block followed by an orange 'esperar' (wait) block. The 'tocar pino' blocks are configured with a pitch of 11, a duration of 0.3 seconds, and a note name of C4. The 'esperar' blocks are configured with a duration of 0.6 seconds. The sequence of durations for the 'tocar pino' blocks is: 0.3, 0.9, 0.9, 0.9, 0.9, 0.9, 0.3, 0.3, 0.3, 0.3, 0.9, 0.9 seconds.



The image displays a sequence of ten Scratch code blocks arranged vertically. Each block consists of a blue 'tocar pino' (play note) block and an orange 'esperar' (wait) block. The 'tocar pino' blocks are configured with a pitch of 11, a note of C4, and a duration of either 0.9 or 0.3 seconds. The 'esperar' blocks are configured with wait times of either 0.6 or 0.3 seconds. The sequence of blocks is as follows:

- 1. tocar pino (11, C4, 0.9) followed by esperar (0.6)
- 2. tocar pino (11, C4, 0.9) followed by esperar (0.6)
- 3. tocar pino (11, C4, 0.3) followed by esperar (0.3)
- 4. tocar pino (11, C4, 0.9) followed by esperar (0.6)
- 5. tocar pino (11, C4, 0.9) followed by esperar (0.3)
- 6. tocar pino (11, C4, 0.3) followed by esperar (0.3)
- 7. tocar pino (11, C4, 0.9) followed by esperar (0.3)
- 8. tocar pino (11, C4, 0.3) followed by esperar (0.6)
- 9. tocar pino (11, C4, 0.3) followed by esperar (0.3)
- 10. tocar pino (11, C4, 0.9) followed by esperar (0.6)



Finalizamos a programação da primeira mascote! Agora, vamos programar a segunda, lembrando que, no momento de execução do projeto, carregue a respectiva programação para cada uma.

Programação 2 - Mascote que responderá (Equipe 2)

Assim como foi feito na Programação 1, iniciaremos também a Programação 2 definindo o acendimento dos LEDs quando ocorrer a leitura da porta digital 11:

Figura 33 - Início da programação para acionamento dos LEDs da mascote 2

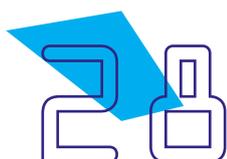


Figura 34 - Primeira letra da programação 2



Figura 35 - Segunda letra da programação 2



Figura 36 - Terceira letra da programação 2

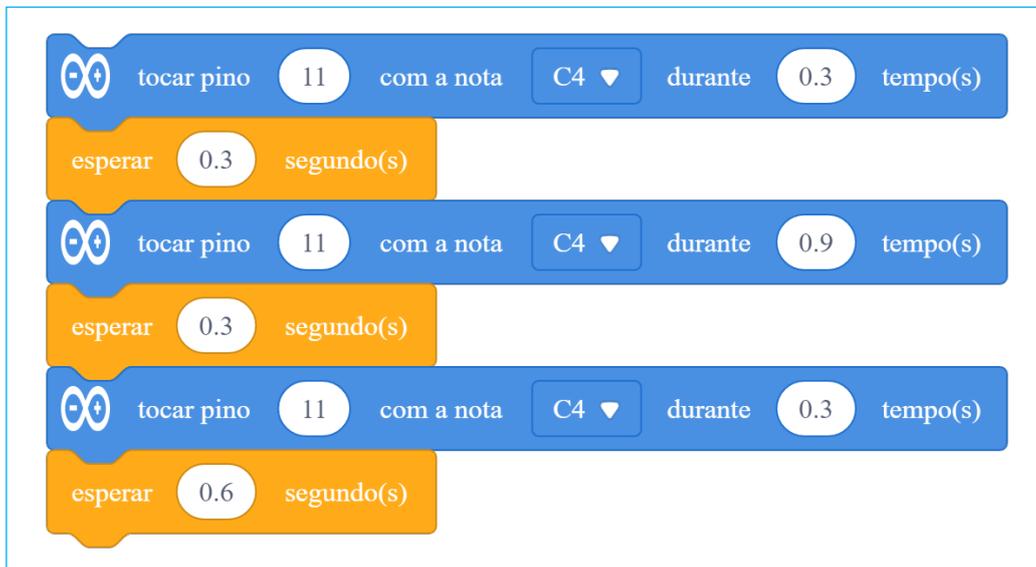


Figura 37 - Quarta letra da programação 2



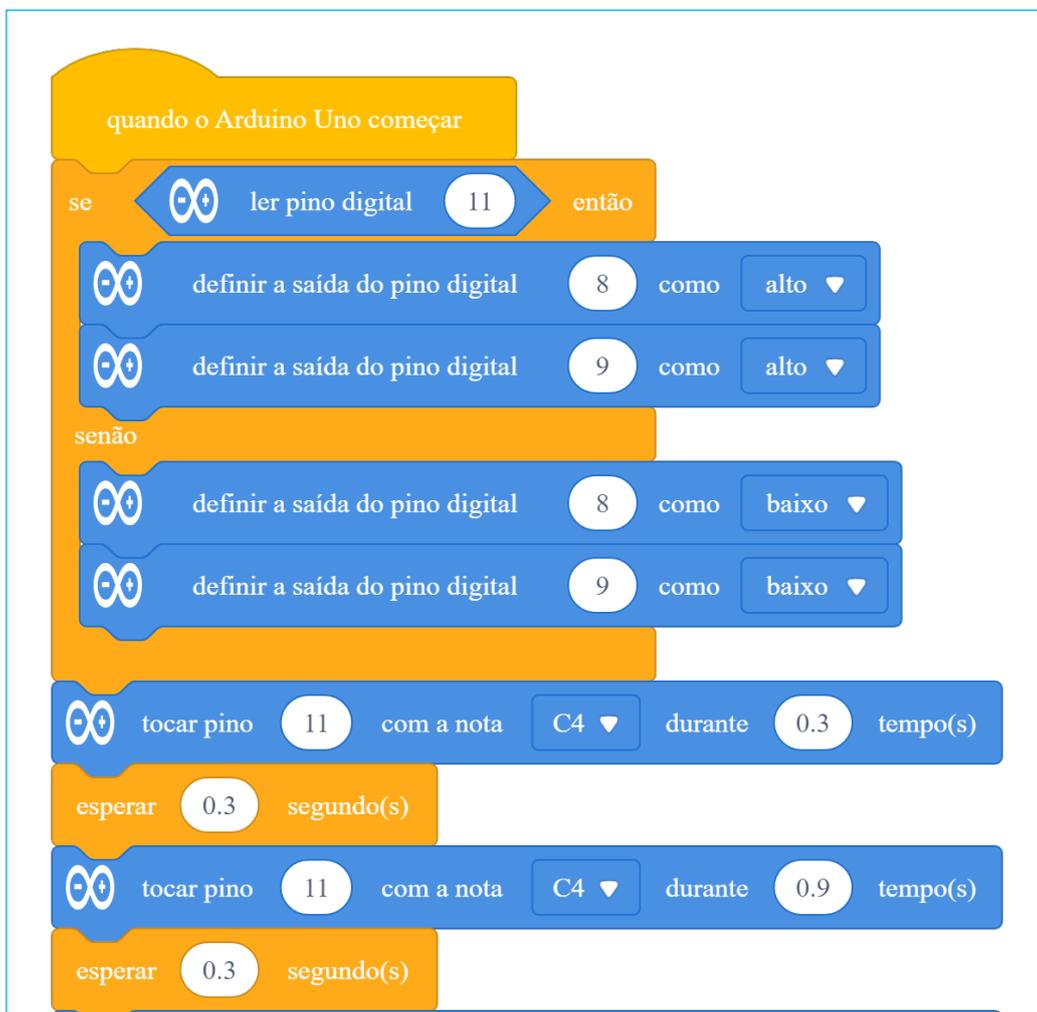
Figura 37 - Quinta letra da programação 2



Figura 38 - Sexta letra da programação 2



Figura 39- Programação 2 completa para a mascote 2



Fonte: mBlock

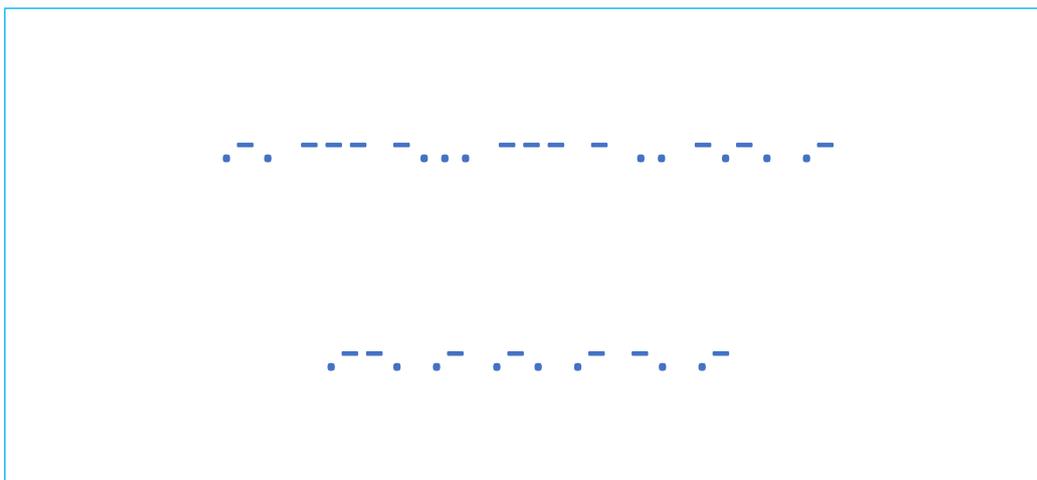


The image displays a sequence of 10 Scratch code blocks arranged vertically. Each block consists of a blue 'tocar pino' (play note) block and an orange 'esperar' (wait) block. The 'tocar pino' blocks are set to play note 11 on piano C4 for a duration of 0.9 seconds. The 'esperar' blocks are set to wait for 0.3 seconds. The sequence of wait times is: 0.3, 0.6, 0.3, 0.6, 0.3, 0.9, 0.3, 0.6, 0.3, 0.6 seconds.





Figura 40 - Transcrição do código Morse das mascotes 1 e 2



Finalizadas as programações, cada equipe carrega um código para sua mascote. Acompanhem a emissão sonora e visual de cada BIT e procurem identificar quais são os sinais longos e quais são os sinais curtos. Agora, repita as emissões e desenhe ou escreva em seu caderno ou papel uma bolinha para o sinal curto e um tracinho para o sinal longo.

Confira, na tabela **Equivalência de letras, números e sinais de pontuação em código Morse**, a letra correspondente a cada sinal que você anotou e procure decifrar a mensagem que o BIT disse a você!

3. Feedback:

a. Compartilhe com seus colegas cada projeto e a programação, conferindo as mensagens “ditas” pelas mascotes;

b. Analise e troque informações com os colegas sobre como foi a experiência de montar cada protótipo e trabalhar entre equipes;

c. Reflita se as seguintes situações ocorreram:

i. Você e os seus colegas trocaram ideias no momento da montagem dos protótipos e programação do buzzer e dos LEDs de cada mascote?

ii. Você teve algum problema ao realizar a montagem e programação dos protótipos para que cada um expressasse suas mensagens? Qual? Como você resolveu?

Desafio:

i. Com a tabela do código Morse, programe as mascotes para que elas expressem outras mensagens, ampliando seu repertório de comunicação.

E se?

... o buzzer e os LEDs não acionarem, verifique as conexões realizadas.

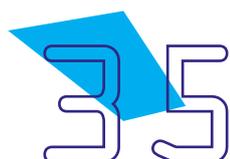
... o projeto não funcionar, verifique também se a programação está adequada à montagem e com as portas digitais indicadas corretamente e o programa carregado ao Arduino Uno R3.



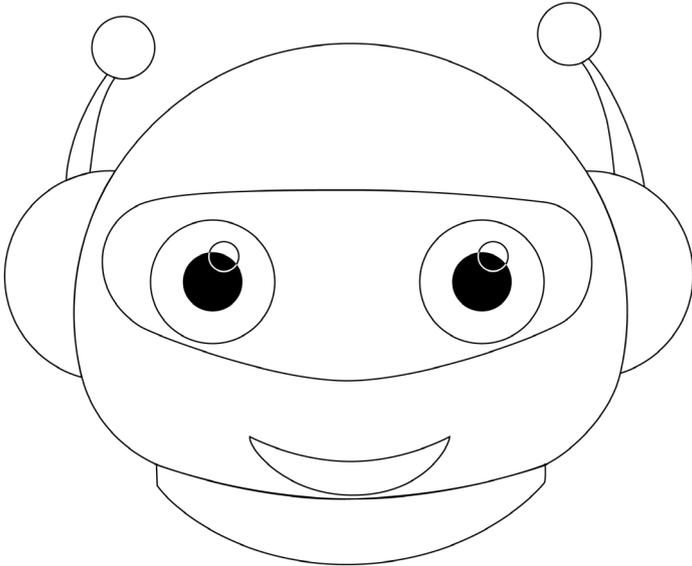


Referências:

- 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. Acesso em: 22 jul. 2022.
- EVERS, Vanessa. **Sobre robôs e humanos**. 2018. Disponível em: <https://pt.unesco.org/courier/2018-3/robos-e-humanos>. Acesso em: 18 nov. 2022.
- MAKEBLOCK. mBlock. **Download mBlock**. Disponível em: <https://mblock.makeblock.com/en-us/download/>. Acesso em: 03 ago. 2022.
- MAKEBLOCK. MBlock. **Make with Code**. Disponível em: <https://mblock.makeblock.com/en-us/>. Acesso em: 03 ago. 2022.
- SUPERINTERESSANTE. **Como este estudo ensinou um robô a ter “empatia”**. Disponível em: <https://super.abril.com.br/tecnologia/como-este-estudo-ensinou-um-robo-a-ter-empatia/>. Acesso em: 11 nov. 2022.
- SALGADO, Hugo David Marques. **Código Morse: o que é e como surgiu**. Disponível em: <https://student.dei.uc.pt/~hsalgado/CP/artigo.htm>. Acesso em: 16 nov. 2022.
- SUPERINTERESSANTE. **Dança das abelhas é aplicada na comunicação entre robôs**. Disponível em: <https://super.abril.com.br/tecnologia/danca-das-abelhas-e-aplicada-na-comunicacao-entre-robos/>. Acesso em: 11 nov. 2022.
- MANUAL DO MUNDO. **SOS não é o que parece**. Entenda o código Morse. YouTube. Disponível em: <https://youtu.be/ciGkt7nse2U>. Acesso em: 11 nov. 2022.



Anexo 01 - Rosto da mascote BIT



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

EQUIPE ROBÓTICA PARANÁ

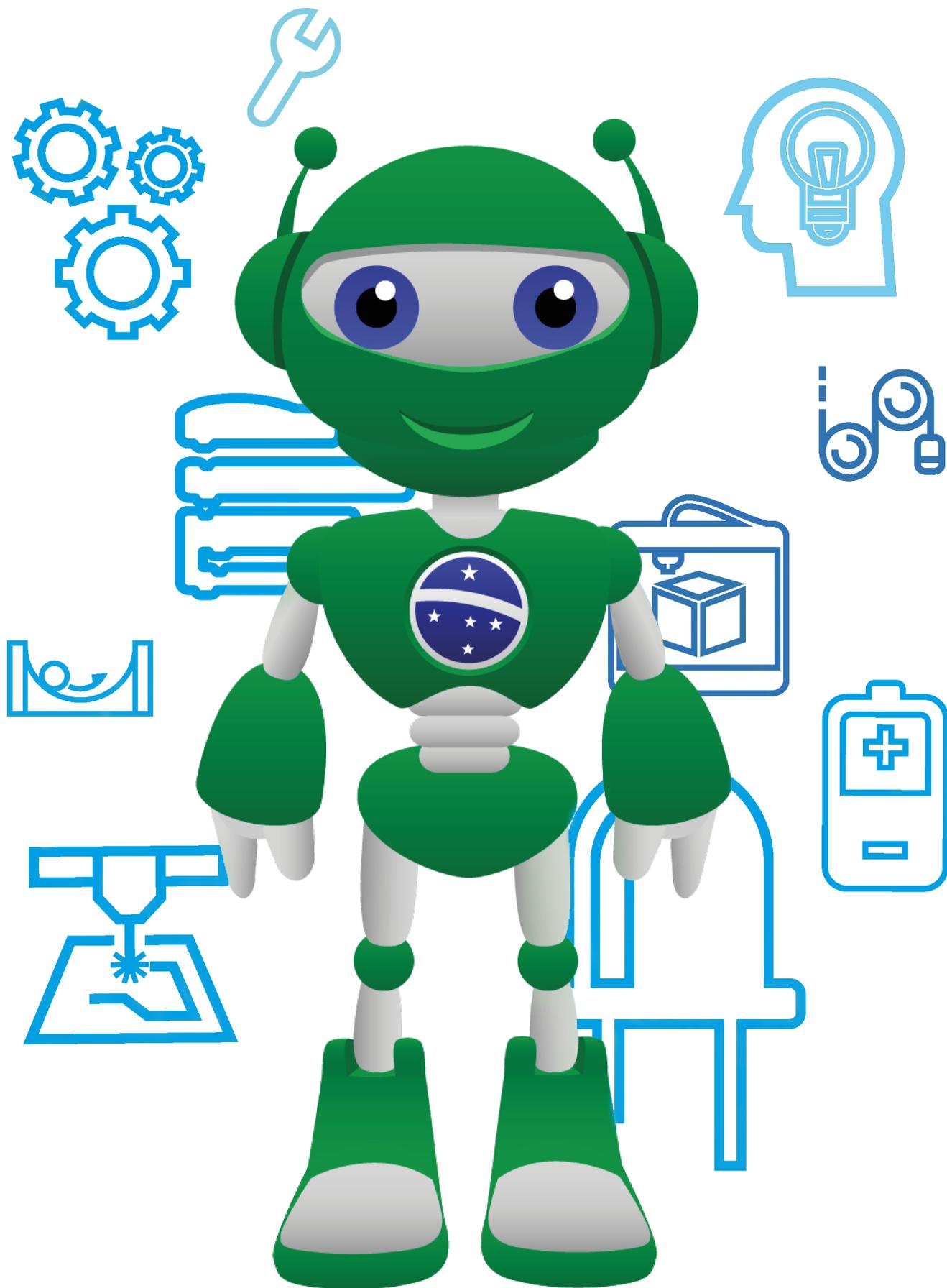
Adilson Carlos Batista
Andrea da Silva Castagini Padilha
Cleiton Rosa
Darice Alessandra Deckmann Zanardini
Edna do Rocio Becker
José Feuser Meurer
Marcelo Gasparin
Michele Serpe Fernandes
Michelle dos Santos
Roberto Carlos Rodrigues

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 e do Esporte do Paraná (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



Diretoria de Tecnologia e Inovação