

ROBÓTICA

Primeiros Passos

Módulo 2



AULA



Aperte para acionar

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

Adilson Carlos Batista

Cleiton Rosa

Darice Alessandra Deckmann Zanardini

Validação de Conteúdo

Cleiton Rosa

Revisão Textual

Adilson Carlos Batista

Orlando de Macedo Júnior

Normalização Bibliográfica

Ricardo Hasper

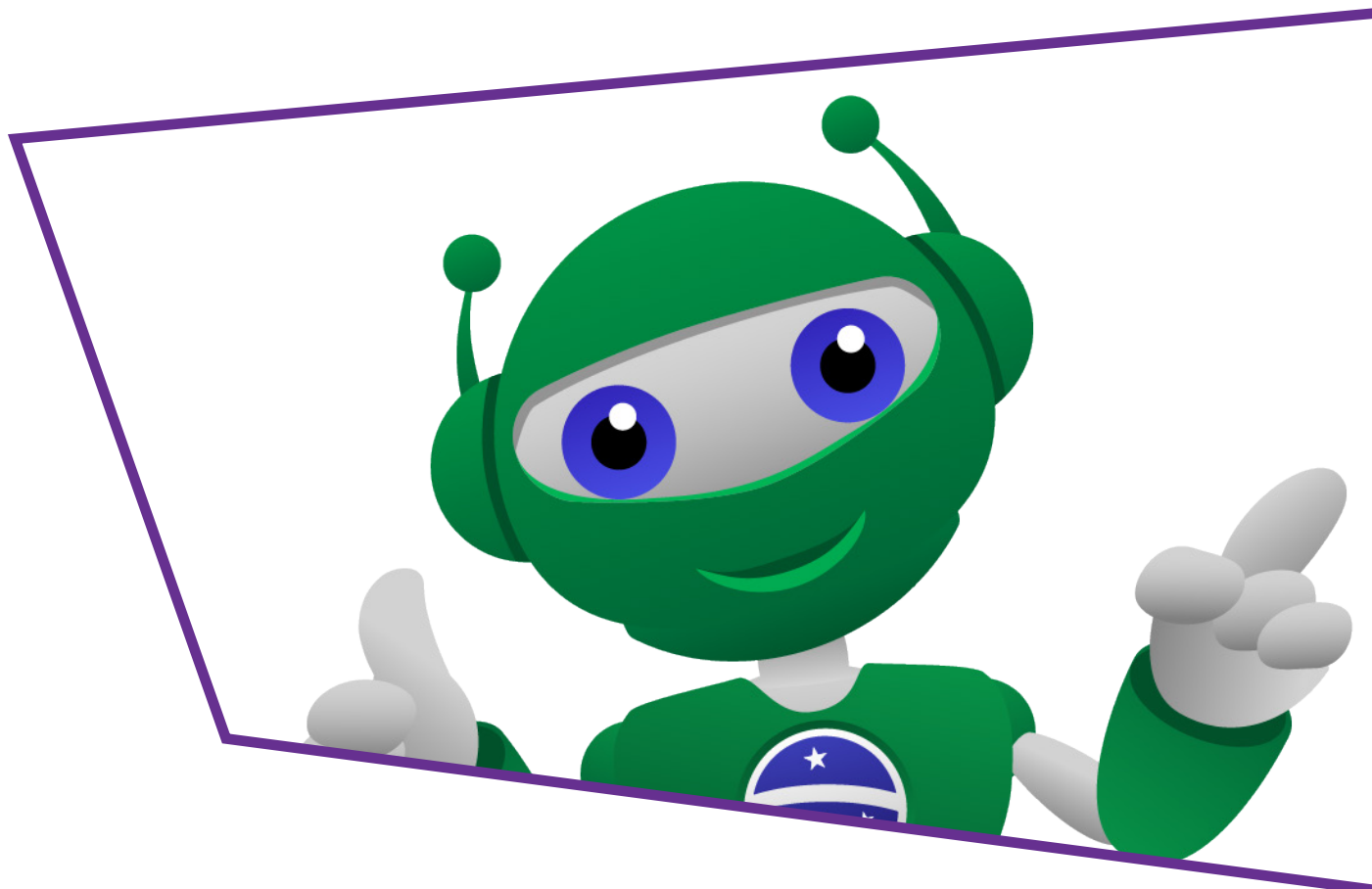
Projeto Gráfico e Diagramação

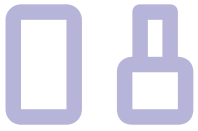
Edna do Rocio Becker

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. Desafios	18
4. Feedback	18
Referências	19





Aperte para acionar



Introdução

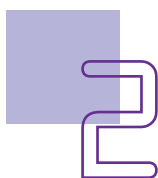
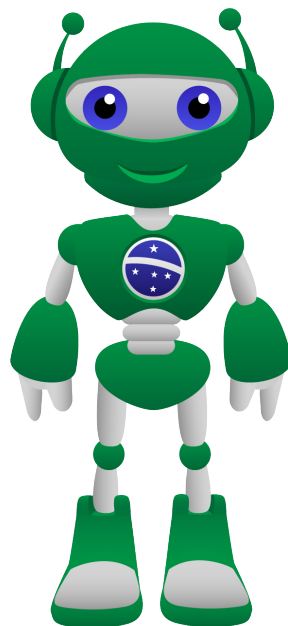
O *push button*, conhecido também como chave tátil, é um dos sensores mais utilizados em prototipagens para controle de outros componentes, através de uma pressão produzida, por exemplo, com o dedo.

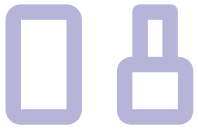
Nesta aula, você vai, além de conhecer esse componente, programá-lo para uma primeira forma de uso: controlar o acionamento de um LED.



Objetivos desta Aula

- Conhecer o componente *push button*;
- Fazer a montagem do LED com o *push button* na protoboard;
- Programar o *push button* para controle do LED.





Competências Gerais Previstas na BNCC

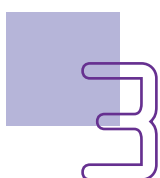
[CG02] - Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas)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

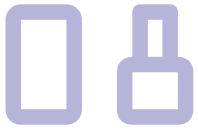
[CG04] - Utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital –, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes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





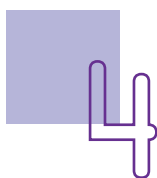
Habilidades do Século XXI a Serem Desenvolvidas

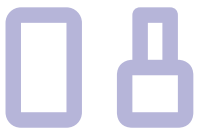
- Pensamento crítico;
- Afinidade digital;
- Resiliência;
- Resolução de problemas;
- Colaboração;
- Comunicação;
- Criatividade.



Lista de Materiais

- 01 placa protoboard;
- 01 placa Arduino;
- 01 *push button*;
- 01 resistor 220 Ohms;
- 01 LED 5mm.
- 04 jumpers macho-macho;
- 01 cabo USB;
- Computador;
- Software mBlock.



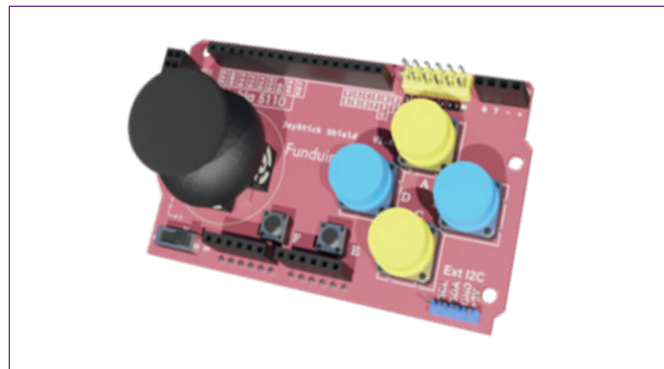


Roteiro da Aula

1. Contextualização:

Você sabe o que é um *push button* (botão de pressão)? Talvez, você não tenha ouvido falar ainda, mas você já teve contato com modelos de *push button* nas aulas que contemplaram o *joystick shield* para controlar os personagens nos jogos criados no mBlock.

Figura 1 – Alavanca e botões *push button* no joystick



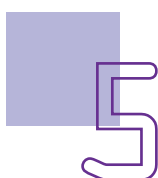
Fonte: Banco de imagens SEED / DTI / CTE, 2022

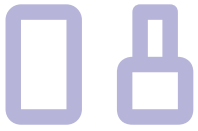
São variados os modelos de *push button*, os quais operam como uma forma de interruptor para abrir e fechar um circuito e, por isso, também são conhecidos como chave tátil.

Figura 2 – Alguns modelos de *push button* ou chave tátil



Fonte: Banco de imagens SEED / DTI / CTE, 2022





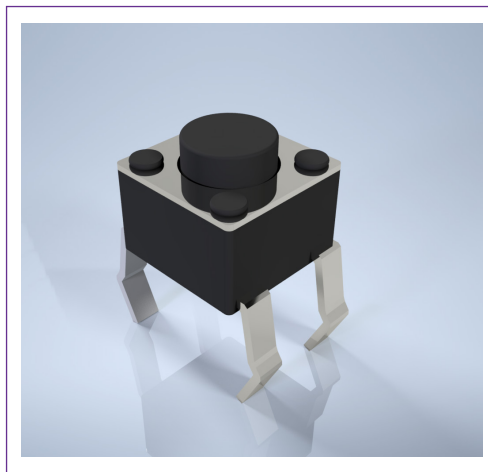
Aperte para acionar

Utilizaremos em nossa programação o modelo de botão 6x6x5 mm com quatro terminais para ligar e desligar um LED. Se sua escola recebeu os kits de Robótica em 2023, o componente é um pouco maior (12x12x7,3), mas isso não interferirá na programação que apresentamos na atividade proposta, pois a funcionalidade do *push button* é a mesma, independente do projeto! Então, vamos para nossa aula!

2. Conteúdo:

O *push button* é utilizado nas áreas de produtos eletrônicos, eletrodomésticos e muito mais devido a sua possibilidade de controle tanto de componentes como também de processos em um circuito elétrico, possibilitando ou interrompendo as operações como a de um som, uma luz etc.

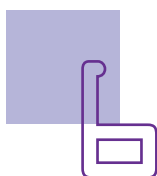
Figura 3 - *Push button* disponibilizado nos kits de Robótica distribuídos em 2022

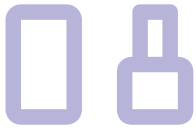


Fonte: Banco de imagens SEED / DTI / CTE, 2023

Especificações e características:

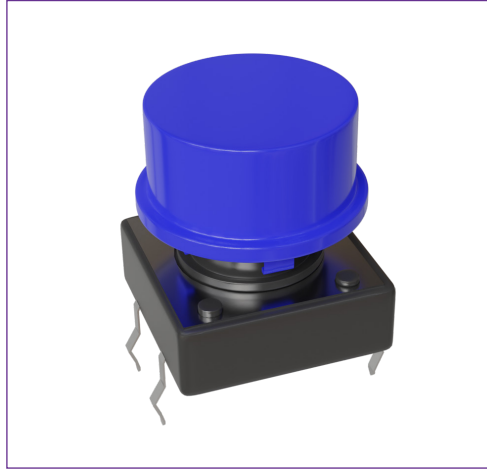
- **Tipo de chave:** táctil com quatro terminais
- **Tensão máxima:** 12 V
- **Corrente máxima:** 50mA
- **Dimensões:** 6x6x5 mm





Aperte para acionar

Figura 4 - *Push button* disponibilizado nos kits de Robótica distribuídos em 2023



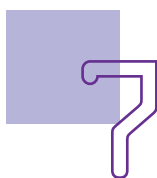
Fonte: Banco de imagens SEED / DTI / CTE, 2023

Especificações e características:

- **Tipo de chave:** táctil com quatro terminais
- **Capas em Plástico**
- **Tensão máxima:** 12 V
- **Corrente máxima:** 50mA
- **Dimensões:** 12x12x7,3 mm

Como funciona o *push button*? Ele tem um toque leve e, ao ser pressionado, pode abrir ou fechar o circuito elétrico com que está conectado. Isso ocorre porque o *push button* “converte” a ação mecânica (de pressionar o botão) em ação no circuito elétrico (permitir ou interromper a passagem de energia) - e por isso dizemos que ele opera como um tipo de interruptor.

Vamos ao nosso protótipo para conferir o *push button* em funcionamento para o controle do acionamento de um LED? Com a programação, a cada toque no *push button* o status do LED será alterado entre ligado e desligado.

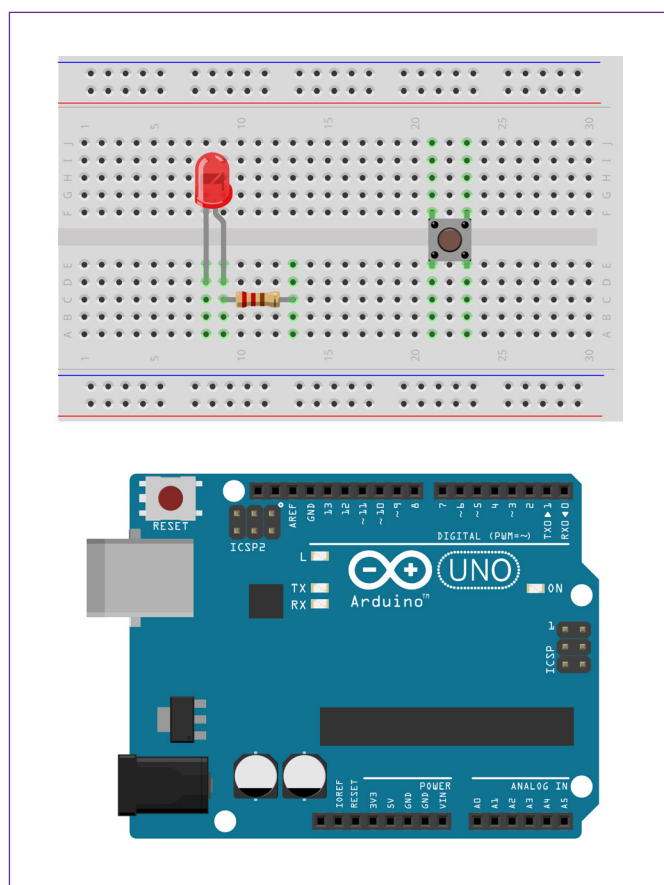




Aperte para acionar

Inicie a montagem dos componentes eletrônicos encaixando, na protoboard, o LED (mantenha seu terminal negativo para o lado esquerdo da protoboard), o *push button* e o resistor de 220 Ohms, conforme posições indicadas pela figura 05. Observe que o *push button* está conectado exatamente sobre a linha central da protoboard para seu perfeito funcionamento e o resistor está com uma das hastes inserida na mesma coluna do terminal positivo do LED (terminal da direita) e uma parte livre da protoboard.

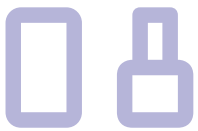
Figura 5 - Inserção dos componentes na protoboard



Fonte: Fritzing

Com um jumper, conecte uma das portas GND do Arduino à coluna da protoboard na qual inserimos o terminal negativo do LED e, com outro jumper, conecte a porta digital 13 do Arduino à coluna da outra extremidade do resistor (considere a haste do lado oposto no qual conectamos o terminal positivo do LED).

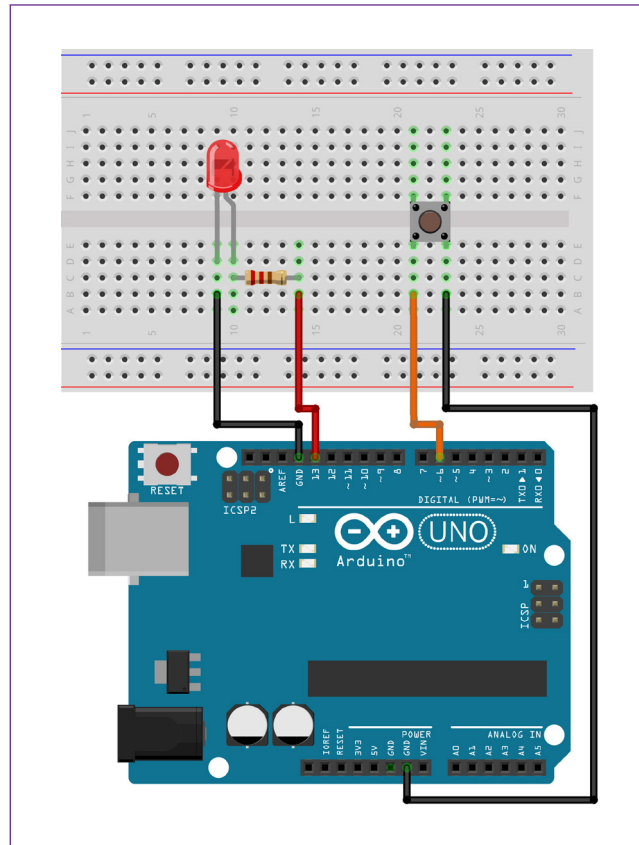




Aperte para acionar

Com mais um jumper, conecte outra porta GND do Arduino a um dos terminais do *push button* e, para finalizar a conexão do Arduino à protoboard, utilize mais um jumper para conectar a porta digital 6 do Arduino ao outro terminal do *push button* (figura 6).

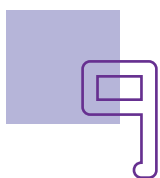
Figura 6 - Conexão dos componentes entre protoboard e Arduino

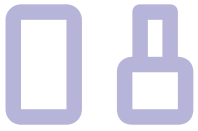


Fonte: Fritzing

A programação para nosso projeto requer as extensões **RP - INPUT PULLUP** e **RP - LEDs** por possuírem funcionalidades que ampliam a programação no mBlock. Consulte, no tutorial **Adicionando Extensões no mBlock**, disponível em rebrand.ly/adicionandoExtensoes_mBlock, sobre como instalá-las.

Iniciaremos nossa programação na aba **Dispositivos**, criando uma variável com o nome **status LED** para definir o estado do LED a partir do acionamento do *push button*. Na programação do Arduino, variáveis reservam locais de memória utilizados durante a execução do projeto para armazenar informações.

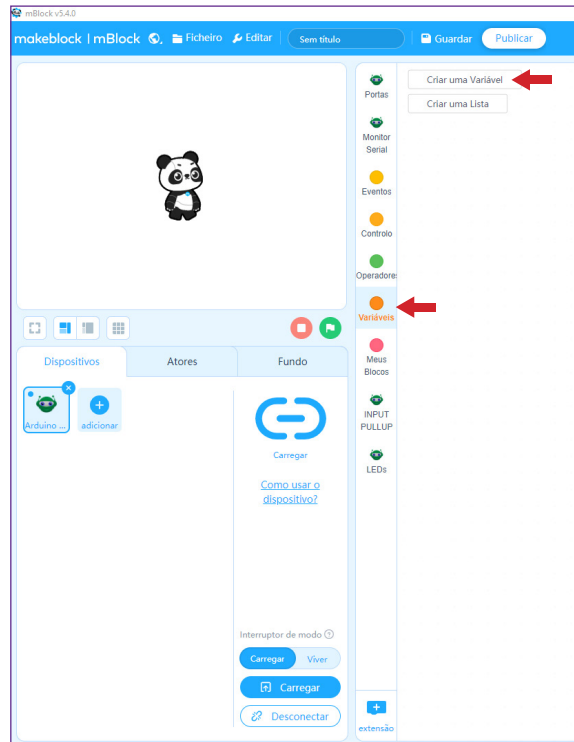




Aperte para acionar

Figura 7 – Menu Variáveis

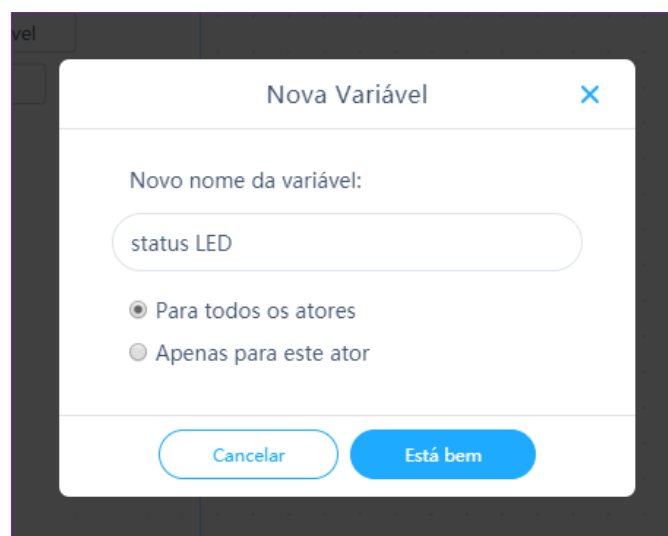
Para isso, no menu de blocos Variáveis, clique na caixa **Criar uma Variável**.



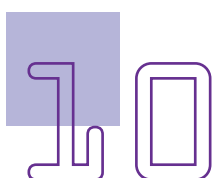
Fonte: mBlock

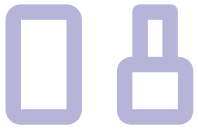
Na janela que abrir, digite o nome da variável que precisa ser criada, de acordo com a necessidade de seu projeto, e clique em **Está bem**.

Figura 8 – Criação de Variável



Fonte: mBlock

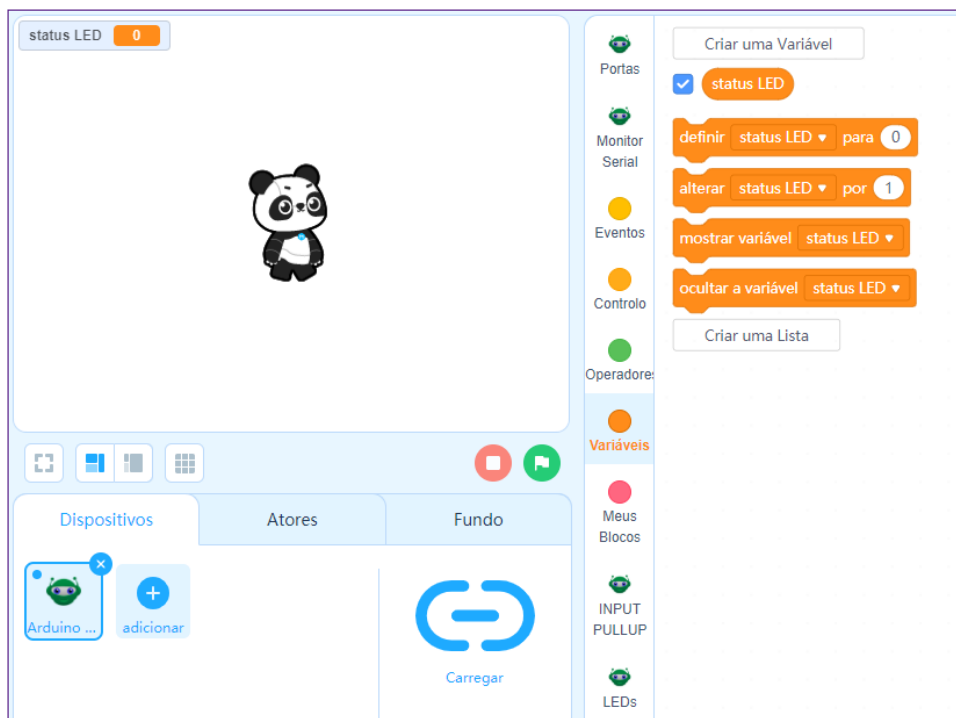




Aperte para acionar

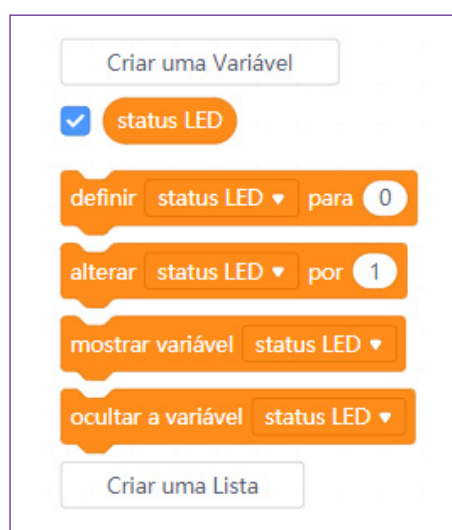
Observe que a criação da variável habilitou blocos para que possamos, nesse projeto, definir e também alterar o status do LED conforme acionamento do *push button*.

Figura 9 - Variável “status LED”

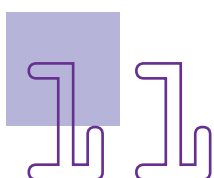


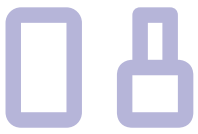
Fonte: mBlock

Figura 10 - Detalhe dos blocos da variável “status LED”



Fonte: mBlock





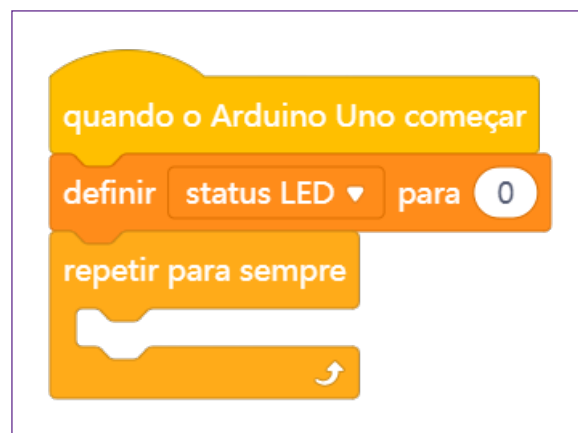
Aperte para acionar

Os LEDs operam com valores booleanos ou sistema binário, ou seja, um sistema em que se tem dois valores ou status possíveis: **0**, que corresponde à condição de desligado (ou baixo, como usamos nas outras programações com mBlock), e **1**, que corresponde à condição de ligado (ou alto, como também usamos nas outras programações com mBlock).

Assim como os demais projetos com o Arduino em que há apenas a montagem do projeto, sem interação com personagens do mBlock, nossa programação começará com o bloco de Eventos <Quando o Arduino Uno começar>, disponível para a aba **Dispositivos**, seguido pelo bloco de Variáveis <definir status LED para 0>, o qual indicará como desligado o status inicial do LED conectado à protoboard.

Como a ideia é que a operação do *push button* para controle do LED seja monitorada constantemente, adicionamos também o bloco <repetir para sempre>.

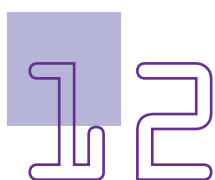
Figura 11 - Programação inicial

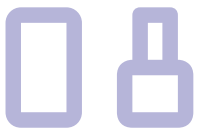


Fonte: mBlock

Agora, toda a programação seguirá dentro da chave <repetir para sempre>. Nela, estará indicado o que acontecerá no circuito cada vez que pressionarmos o *push button*.

Vamos ver, por etapas e com mais detalhes na sequência, a lógica da nossa programação e montagem dos blocos para controle de acionamento do LED pelo *push button*.





Aperte para acionar

Insira, em **<repetir para sempre>**, o bloco de controle **<se __ então>** com o operador de igualdade **<__ = __>** para indicar o status do *push button* que conectamos à porta 6 do Arduino.

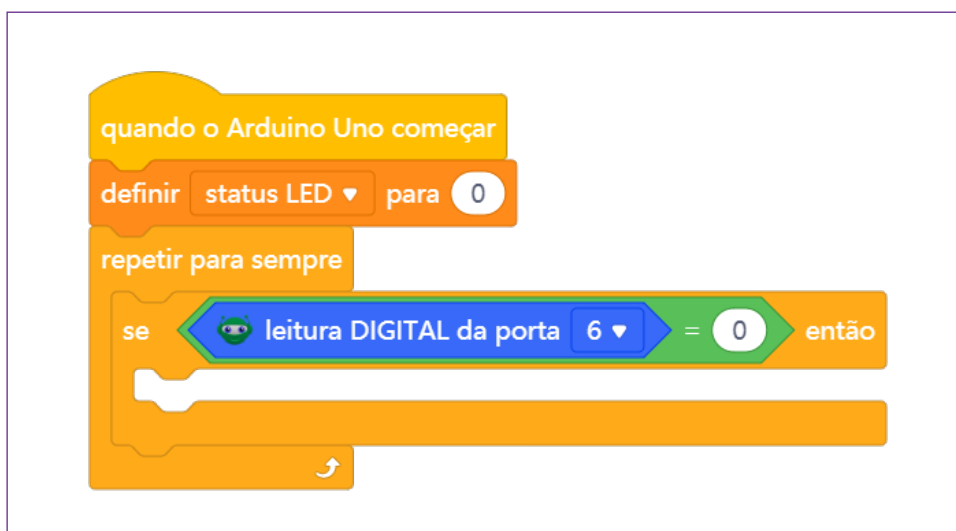
Agora, de um lado da igualdade inserimos o bloco **<leitura DIGITAL da porta 6>**, presente nos blocos da extensão **INPUT PULLUP**, e, do outro lado, o valor **0**. Se a leitura digital da porta de controle do *push button* conectado à porta digital 6 do Arduino for igual a **0**, significa que o botão foi pressionado e então precisaremos verificar qual o status do LED para determinar a ação seguinte quando pressionarmos novamente o botão.

Figura 12 - Inserção do bloco “se... então” e do operador de igualdade



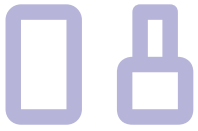
Fonte: mBlock

Figura 13 - Inserção do bloco de leitura do *push button* e valor binário 0



Fonte: mBlock

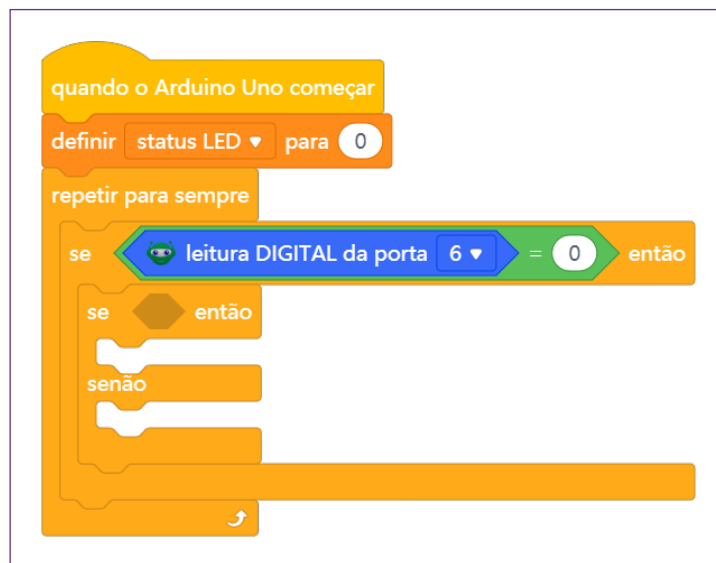




Aperte para acionar

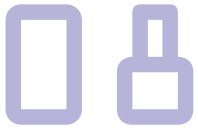
Na sequência, nossa programação fará a conferência do status do LED. Para isso, vamos adicionar no bloco <se __ então> e bloco de controle <se __ então, senão>. Tenha bastante atenção à posição correta dos blocos para que a lógica de verificação do status do LED seja aplicada e permita o controle de seu acionamento pelo push button.

Figura 14 - Inserção do bloco “se... então, senão...”



Fonte: mBlock

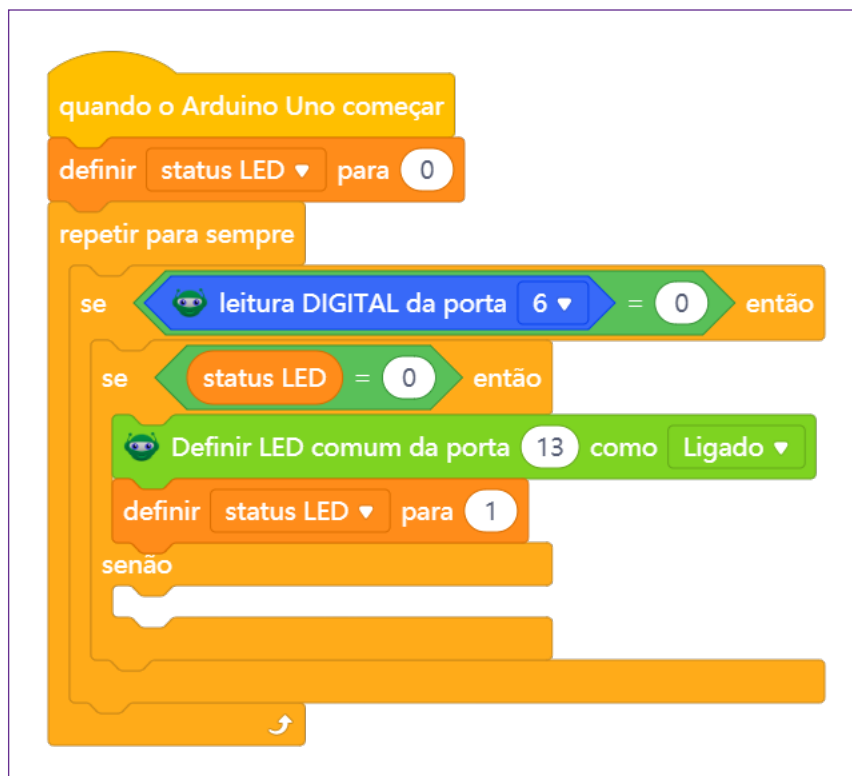




Aperte para acionar

Se a variável <status LED> for igual a **0**, indicado também pelo operador de igualdade <__ = __>, daremos o comando para ligar o LED conectado à porta digital 13 do Arduino através do bloco <Definir LED comum da porta 13 como Ligado>, disponível nos blocos da extensão **LEDs**, e alteramos seu status para **ligado** pela variável <definir status LED para 1>.

Figura 15 - Complemento da programação com ação se o LED estiver desligado



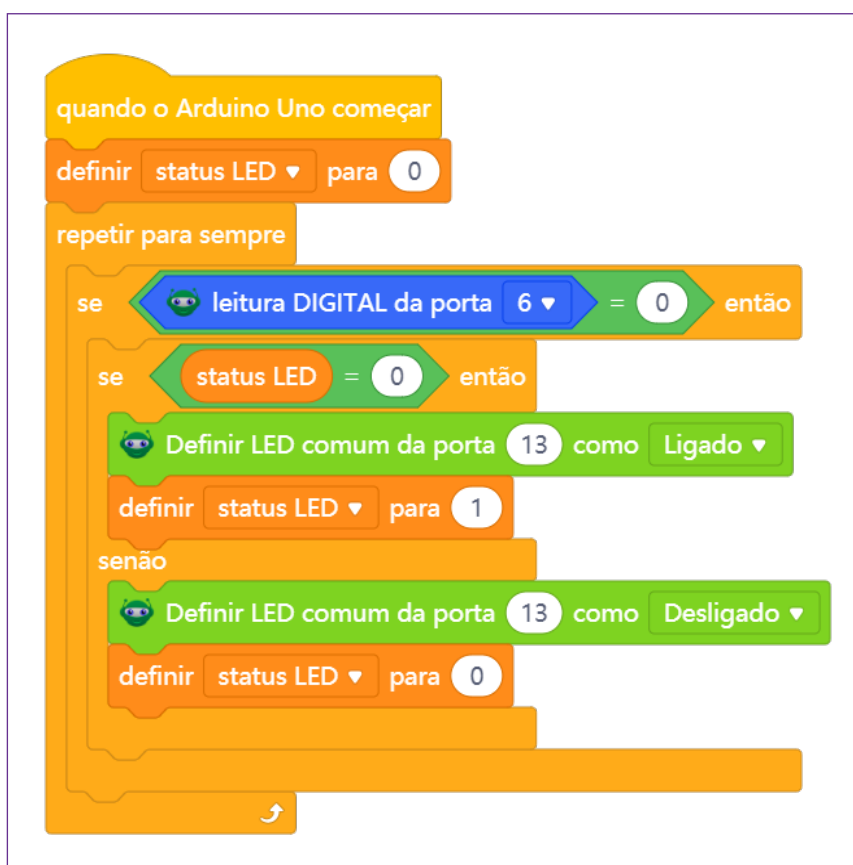
Fonte: mBlock



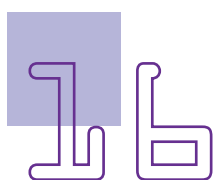
Aperte para acionar

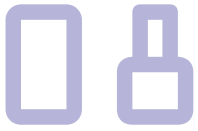
Se o LED já estiver ligado, daremos o comando para desligar o LED através do bloco <**Definir LED comum da porta 13 como Desligado**>, também disponível nos blocos da extensão **LEDs**, e alteramos seu status para **desligado** pela variável <**definir status LED para 0**>.

Figura 16 - Complemento da programação com ação se o LED estiver ligado



Fonte: mBlock

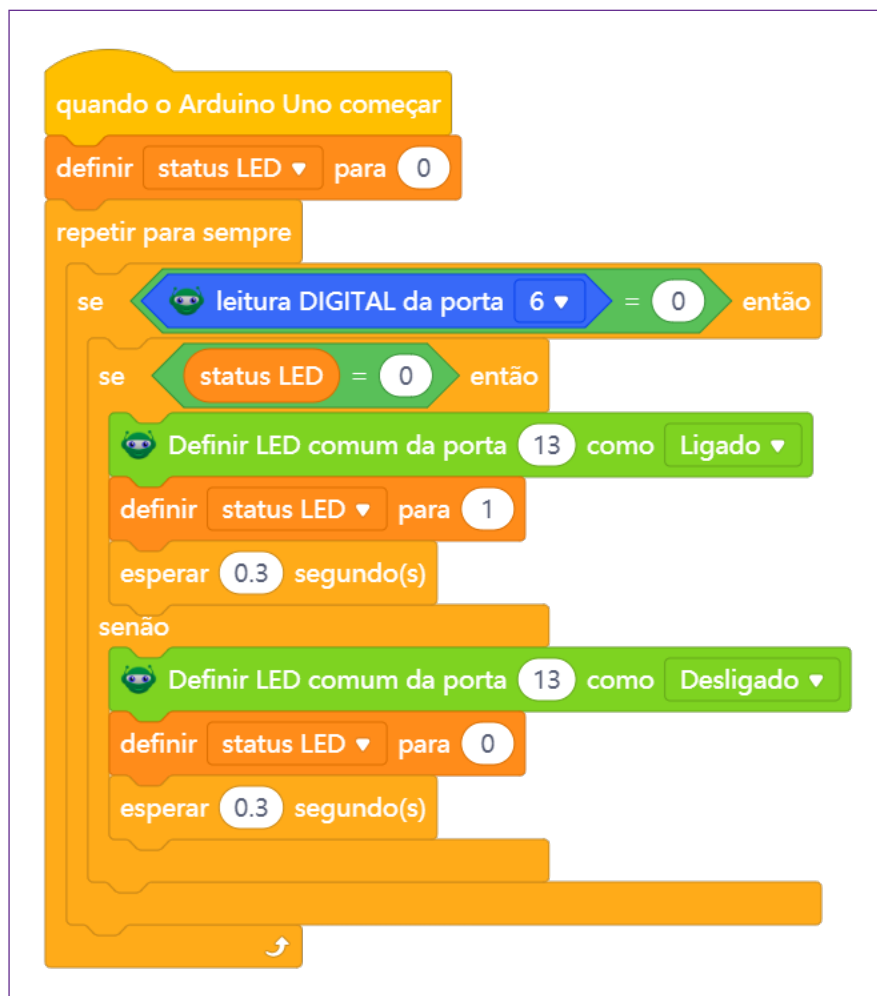




Aperte para acionar

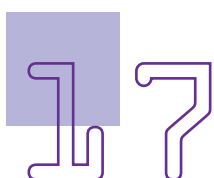
Como o Arduino processa 16 milhões de informações por segundo (16 MHz), entre cada alteração de status LED, após o bloco da variável <definir status LED> vamos inserir um tempo de 300 milissegundos para evitar várias leituras do status do LED quando o botão for pressionado, completando nossa programação.

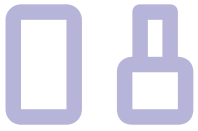
Figura 17 - Programação completa



Fonte: mBlock

Finalizada a programação, carregue-a no Arduino e experimente o funcionamento do nosso protótipo!





Aperte para acionar

3. Desafios

- I. Que tal montar o circuito de uma maneira alternativa, usando outros furos e posições na placa protoboard?
- II. Que tal criar uma animação virtual de um quarto em que a lâmpada do abajur é acesa pelo *push button*?
- III. Que tal inserir outros componentes, como mais LEDs e *buzzer*, para também serem controlados pelo *push button*?

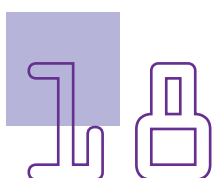
E se ... ?

O projeto não funcionar? Atente-se a alguns dos possíveis erros:

1. Verifique se os jumpers estão na mesma coluna dos terminais dos componentes, fazendo assim a conexão;
2. Verifique se os jumpers estão ligados aos pinos corretos no Arduino;
3. Verifique se o LED não está conectado de modo invertido;
4. Confira as portas digitais informadas na programação.

4. Feedback e finalização

- a. Confira e compartilhe com seus colegas a sua programação, verifique se está tudo funcionando conforme o planejado;
- b. Analise e troque informações com os colegas sobre como você criou sua programação;
- c. Reveja se você entendeu como trabalhar com o *push button* para controle de acionamento de um LED;
- d. Reflita se as seguintes situações ocorreram:
 - i. Você e os seus colegas trocaram ideias no momento da programação do LED com o *push button*?
 - ii. Você teve problema ao fazer a programação? Qual? Como você resolveu?





Referências

BRASIL ESCOLA. **O que é Plano Cartesiano**. Disponível em: <https://brasilecola.uol.com.br/o-que-e/matematica/o-que-e-plano-cartesiano.htm>. Acesso em: 15 set. 2022.

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: 08 set. 2022.

MAKEBLOCK. mBlock. **Download mBlock**. Disponível em: <https://mblock.makeblock.com/en-us/download/>. Acesso em: 15 set. 2022.

MAKEBLOCK. MBlock. **Make with Code**. Disponível em: <https://mblock.makeblock.com/en-us/>. Acesso em: 15 set. 2022.

MAKEBLOCK. mBlock. **Programação em blocos**. Disponível em: <https://ide.mblock.cc/>. Acesso em: 15 set. 2022.

ROBOCORE. **Introdução a chaves e botões**. Disponível em: <https://www.robocore.net/tutoriais/introducao-a-chaves-e-botoes>. Acesso em: 16 fev. 2023.

SILVER, L. Push Button Arduino: 3 modos de ligar botões no Arduino. **Guia Robótica**, 2020. Disponível em: <https://guiarobotica.com/push-button-arduino/>. Acesso em: 15 fev. de outubro de 2021.



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

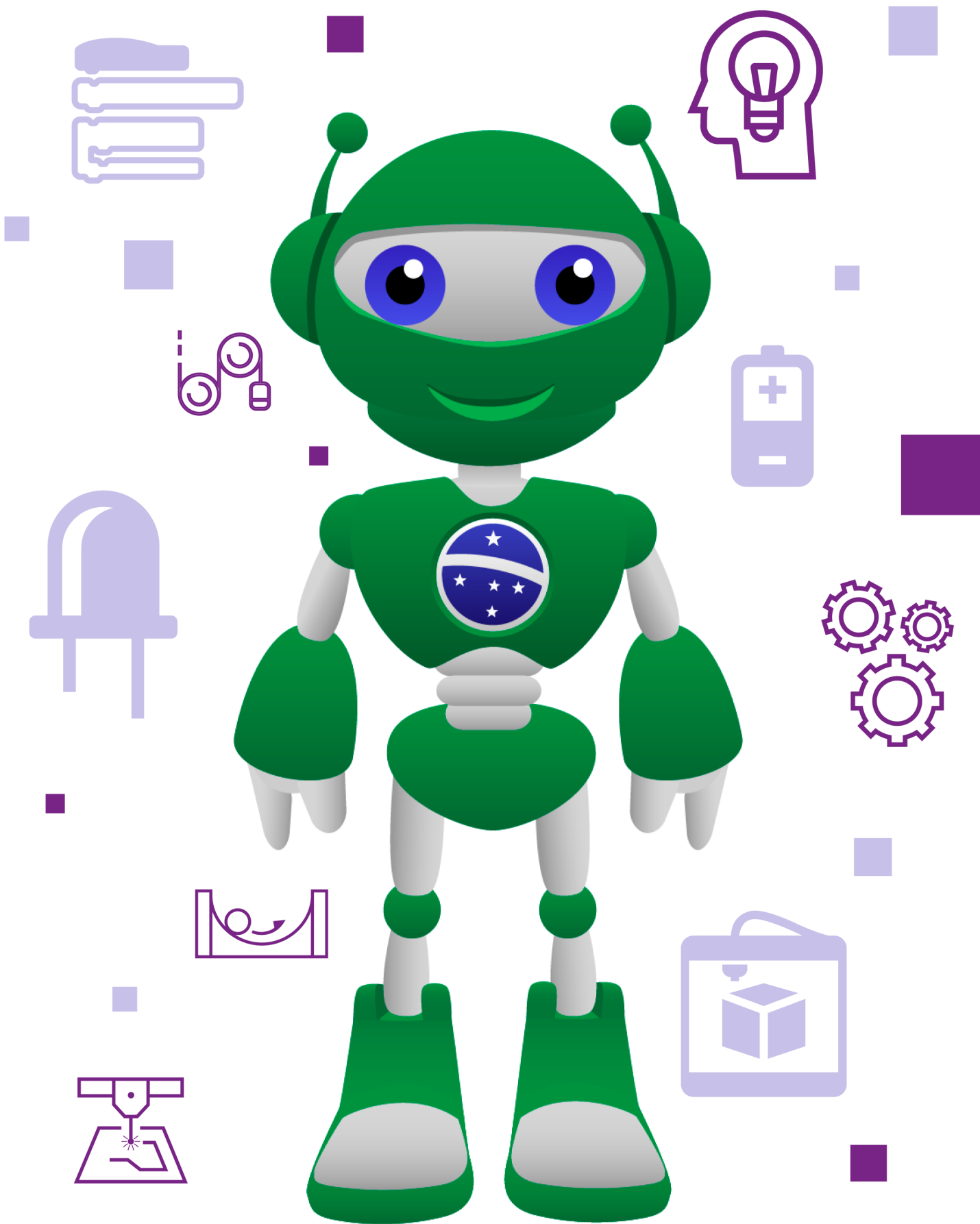
EQUIPE ROBÓTICA PARANÁ

Adilson Carlos Batista
Andrea da Silva Castagini Padilha
Cleiton Rosa
Darice Alessandra Deckmann Zanardini
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