

2ª e 3ª

Série

Robótica

**MATERIAL
DIGITAL**

IA e *Machine Learning* : o que isso tem a ver comigo?

**3º bimestre
Aulas 1 e 2**

**Ensino
Médio**



**GOVERNO DO ESTADO
DE SÃO PAULO**

Conteúdos

- Inteligência artificial;
- *Machine Learning*;
- IA simbólica e IA baseada em dados (visão comparativa e exemplos de IA no cotidiano);
- IA, *Machine Learning*, IoT e microcontroladores.

Objetivos

- Compreender o conceito de IA, *Machine Learning* e suas aplicações;
- Relacionar o uso dessas tecnologias com dispositivos computacionais do cotidiano;
- Analisar como elas estão presentes em serviços, produtos e algoritmos.

Para começar



VIREM E CONVERSEM



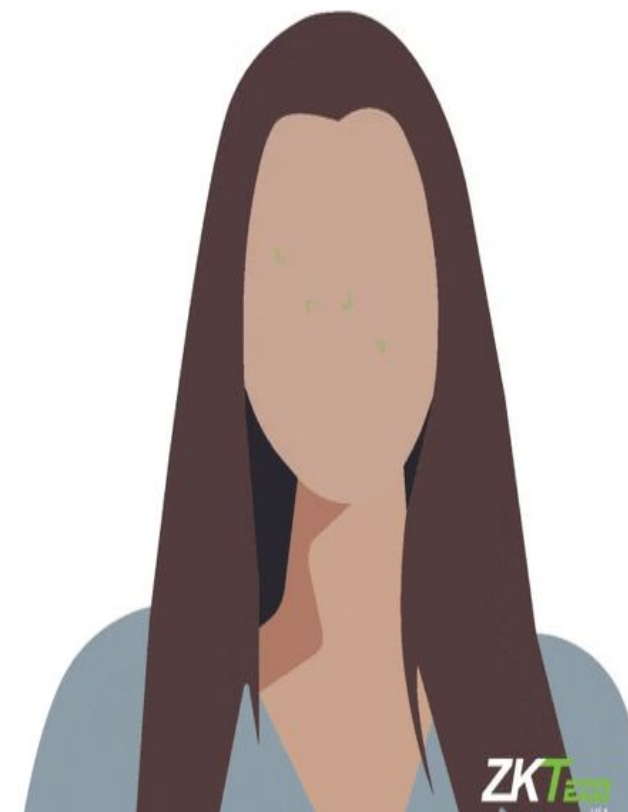
5 minutos

Quando falamos em inteligência artificial, muitas pessoas imaginam sistemas como a ***Skynet**, robôs humanos da ficção científica ou máquinas que pensam sozinhas.

Na prática, a maior parte da inteligência artificial que usamos hoje é muito mais simples e está presente em tarefas do dia a dia. Um exemplo comum é o desbloqueio do celular por reconhecimento facial, que utiliza um tipo de IA para funcionar.

Como uma máquina consegue “perceber” que existe um rosto na frente dela e reagir a isso?

* Inteligência artificial avançada que viu a humanidade como uma ameaça à sua existência. **Alusão à IA do filme *O exterminador do futuro*.**



© Giphy

Como assim, perceber?

De forma simples, podemos dizer que **inteligência artificial** é o nome dado a sistemas criados para realizar tarefas que, até pouco tempo atrás, exigiam inteligência humana – como reconhecer imagens, identificar padrões ou tomar decisões simples.

Existem diferentes formas de construir esses sistemas. Uma delas é a chamada IA simbólica, baseada em regras definidas por pessoas.

Assim como ocorre com aplicações IoT (internet das coisas), não há aprendizado, apenas execução de regras.



© Giphy

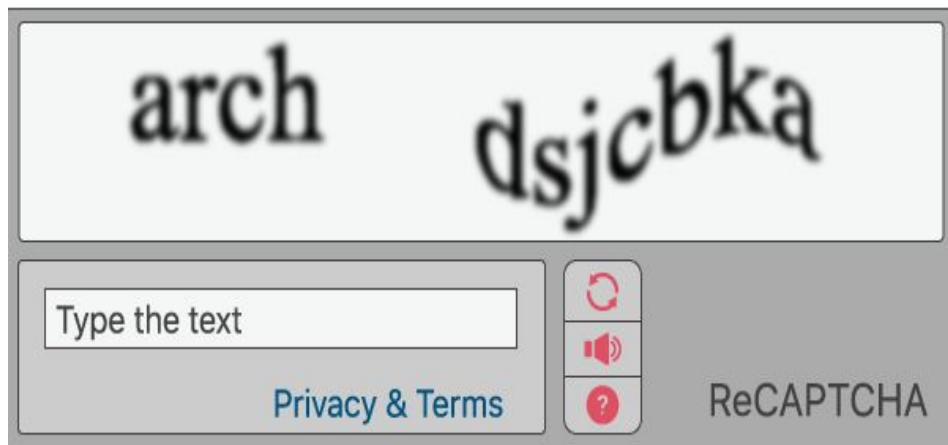
IA simbólica – O ser humano diz exatamente o que a máquina deve fazer

Antes do uso de grandes quantidades de dados, muitos sistemas considerados “inteligentes” funcionavam a partir de regras fixas.

Nesse tipo de IA primitiva, alguém pensa nas situações possíveis e escreve regras do tipo “se apertar um botão, exiba determinado texto”. Ou seja, ela funciona bem para situações simples, mas não lida bem com variações e imprevistos.



© Giphy



Disponível em:

<https://www.cloudflare.com/pt-br/learning/bots/how-captchas-work/>.

Acesso em: 22 jan. 2026.

Você já deve ter se deparado com um destes mecanismos (comumente chamados de *CAPTCHAs*) em sites da internet.

Eles são usados até hoje e constituem um sistema de segurança contra *bots* na internet. Ele atua verificando se a resposta dada pelo internauta bate com a imagem exibida no display.

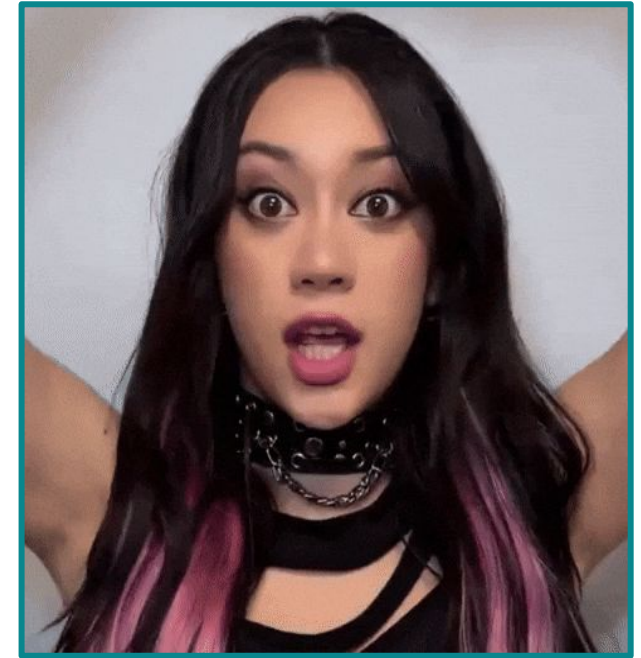
Observe que a IA simbólica não foi criada para imitar a inteligência humana, mas para executar regras definidas por pessoas. No caso do *CAPTCHA*, ele existe somente para **dificultar ações automáticas** em páginas da internet.

* Um **bot** (abreviação de *robot*) é um programa de computador criado para executar tarefas automaticamente na internet, sem precisar de intervenção humana. Ele pode ser usado para coisas **boas** ou **ruins**, dependendo da intenção de quem o programou.

Foco no conteúdo

A **IA baseada em dados** é um modelo em que a máquina não recebe todas as regras prontas. Em vez disso, ela é treinada para identificar padrões a partir de dados detectados por câmeras ou sensores e tomar decisões com base no que foi captado. Sistemas de reconhecimento facial, de voz ou de imagens funcionam dessa forma.

É aqui que entra em cena o ***Machine Learning***, que é um conjunto de técnicas programadas que permitem à máquina esse aprendizado.

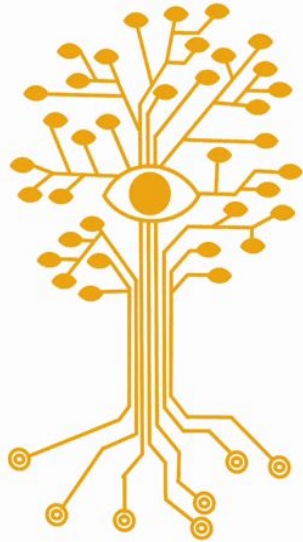


© Giphy

Continua



IA no seu dia a dia



© Giphy

Mesmo sem perceber, você convive com sistemas de inteligência artificial no seu dia a dia.

Quando um aplicativo sugere um conteúdo, seja no YouTube ou nas redes sociais, **algoritmos analisam padrões de comportamento – o seu comportamento.**

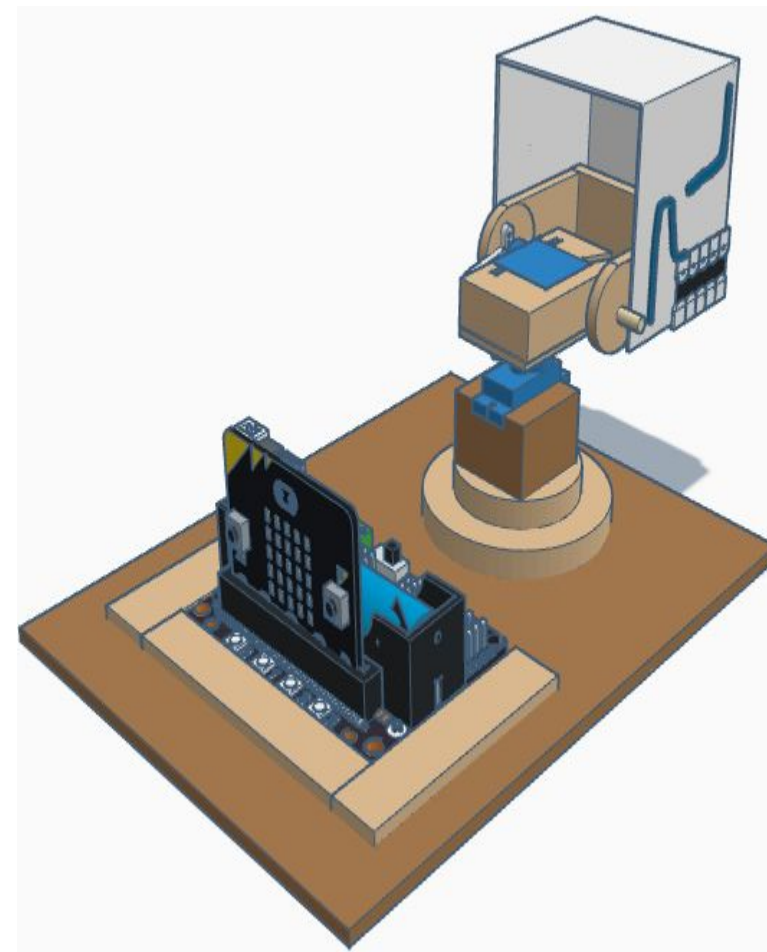
Essas sugestões não são neutras: grandes empresas tentam influenciar escolhas e decisões, o que torna importante **olhar para essas indicações de forma mais crítica.**

Foco no conteúdo

O projeto que será desenvolvido neste bimestre **não tem como objetivo criar ou programar sistemas de inteligência artificial**, mas compreender **como aplicativos que utilizam IA podem ser integrados a um protótipo com micro:bit**.

Antes de colocar a mão na massa, é importante entender **como essa construção acontecerá ao longo das aulas**.

Por isso, apresentamos o planejamento do projeto **aula a aula**, para orientar cada etapa do desenvolvimento e facilitar o acompanhamento do processo.



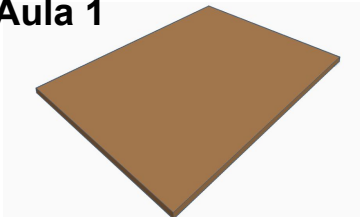
Produzido pela SEDUC-SP com © Tinkercad

Continua



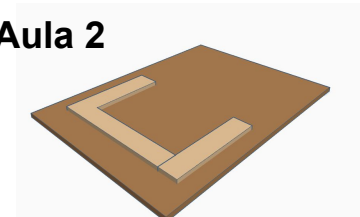
Foco no conteúdo

Aula 1



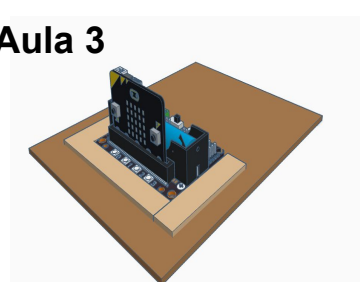
- Inteligência artificial como realizadora de tarefas.
- Compreender os conceitos de IA baseada em dados e IA simbólica.
- Conexão *bluetooth* no micro:bit como ponte entre tecnologias.

Aula 2



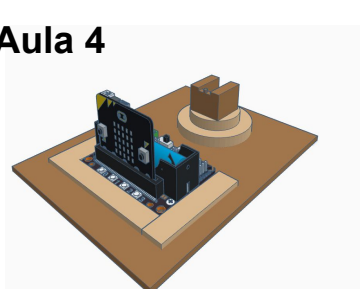
- Compreender como funciona a IoT e sua relação com microcontroladores.
- Analisar como sensores e atuadores permitem a coleta e reação a dados.
- Testar o uso de micro:bit com projetos conectados usando o *bluetooth*.

Aula 3



- Compreender o funcionamento básico de um sistema de aprendizado de máquina.
- Identificar como o micro:bit pode ser usado em projetos de IA.
- Explicar o que significa treinar um modelo e como ele toma decisões.

Aula 4

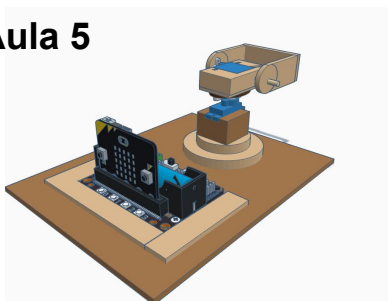


- Entender a cultura STEAM.
- Criar estruturas físicas para protótipos.
- Aplicar a cultura *maker* na improvisação de peças.
- Planejar a montagem de protótipos com sensores e/ou atuadores.



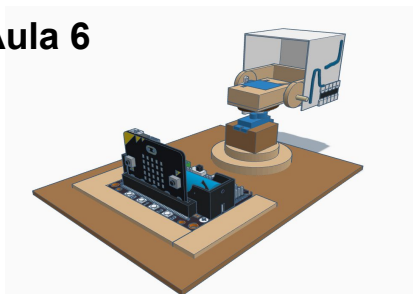
Foco no conteúdo

Aula 5



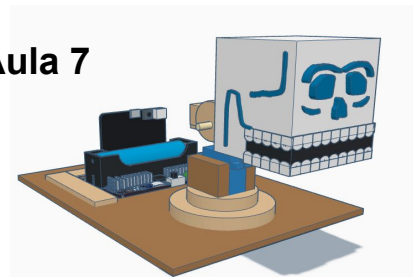
- Criar um protótipo autoral com base em todos os conteúdos vistos.
- Compreender os princípios da cultura *maker* e da abordagem STEAM.
- Experimentar soluções criativas com materiais diversos.
- Projetar peças ou mecanismos funcionais com material reciclado.

Aula 6



- Treinar um modelo básico de IA com micro:bit.
- Testar a integração entre modelo treinado e ação física (output).

Aula 7



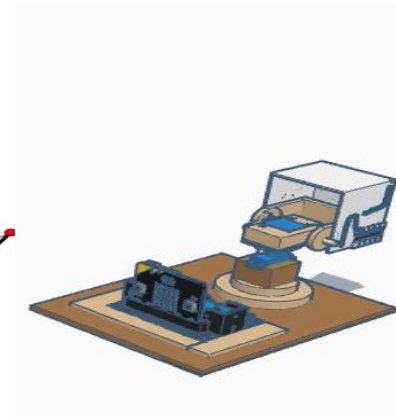
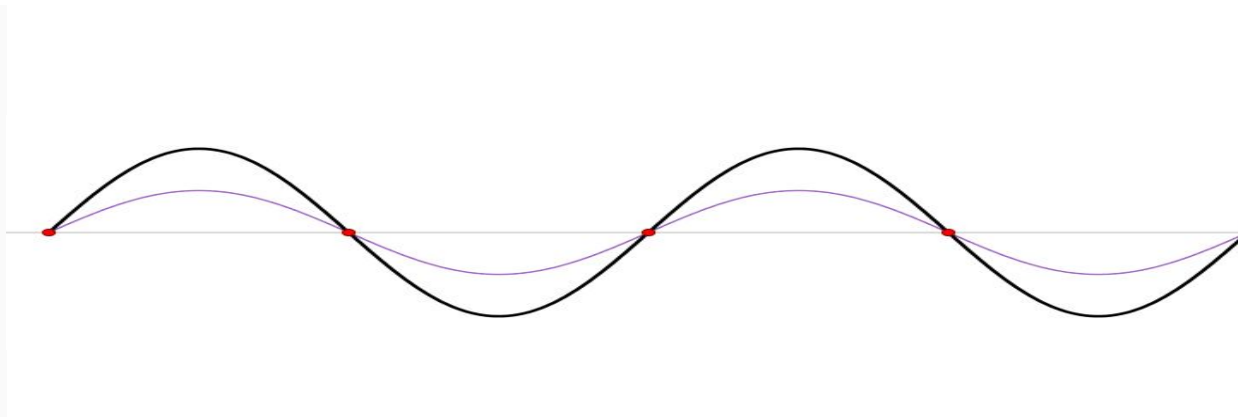
- Comunicar ideias e resultados.
- Avaliar o próprio processo e o do grupo.
- Propor melhorias e novas possibilidades de projeto.

Podem ser feitos ajustes pontuais nos conteúdos ao longo das aulas.

Por que usar *bluetooth* e não o rádio?

Em projetos desse tipo, a inteligência artificial é executada em aplicativos externos, responsáveis por coletar e interpretar informações do ambiente.

Isso significa que o processamento dos dados **não acontece no micro:bit**: ele apenas **reage às decisões tomadas pela IA**.

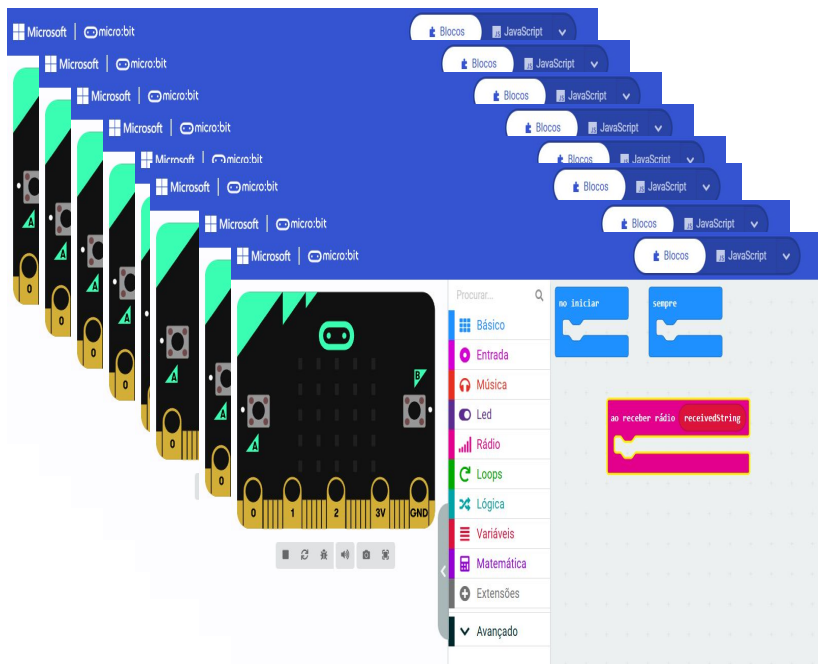


[Esta imagem](#), de autor desconhecido, está licenciada em [CC BY-NC-ND](#). Publicada em https://culturacientifica.com/2018/12/18/ondas-estacionarias/standing_wave_2/. Acesso em 28 de jan. de 2026

Para que essa reação aconteça de forma rápida e contínua, é necessário que o micro:bit mantenha **uma troca constante de informações** com os dispositivos que executam esses aplicativos.

Como esses dispositivos não utilizam o rádio do micro:bit, é preciso recorrer a outra tecnologia de comunicação.

Nesse contexto, o **bluetooth** é a solução mais adequada para realizar essa tarefa.



Por que fazer ajustes?

O MakeCode funciona a partir de um **modelo inicial de programa**. Sempre que você cria um novo projeto, o sistema gera **uma cópia desse modelo**, exclusiva para você.

Cada uma dessas cópias possui **configurações internas** que, até agora, não precisavam ser alteradas. Na imagem, mostramos como são as configurações do modelo-padrão:

Nome

BlueTest

☐ No Pairing Required: Anyone can connect via Bluetooth.

☒ JustWorks pairing (default): Pairing is automatic once the pairing is initiated.

☐ Passkey pairing: Pairing requires 6 digit key to pair.

Salvar **Edit Settings As text**



O que significam estas configurações iniciais:

☐ No Pairing Required: Anyone can connect via Bluetooth.

Pareamento não será exigido: qualquer um pode se conectar via *bluetooth*.

☒ JustWorks pairing (default): Pairing is automatic once the pairing is initiated.

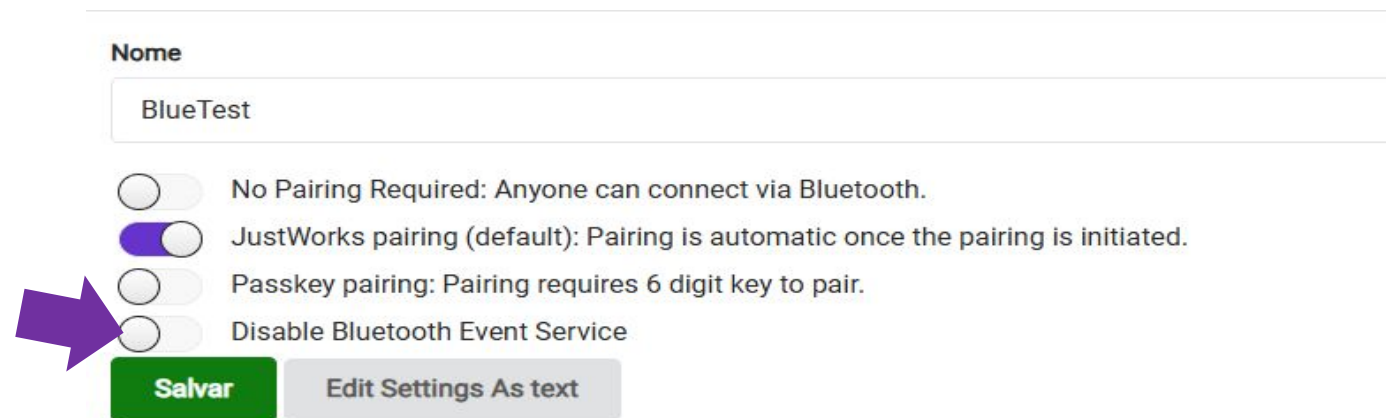
Só funciona com emparelhamento (padrão): o pareamento é automaticamente iniciado quando iniciado na placa – isso ocorre com uma combinação de teclas ao ligar o micro:bit.

☐ Passkey pairing: Pairing requires 6 digit key to pair.

Emparelhamento com senha: requer senha de 6 dígitos para parear.



Quando **adicionamos a extensão *bluetooth***, estas configurações ganham uma nova opção. Observe:



Nome

BlueTest

☐ No Pairing Required: Anyone can connect via Bluetooth.

☒ JustWorks pairing (default): Pairing is automatic once the pairing is initiated.

☐ Passkey pairing: Pairing requires 6 digit key to pair.

☐ Disable Bluetooth Event Service

Salvar Edit Settings As text

Produzido pela SEDUC-SP com © MakeCode

Na aula de hoje, vamos mostrar **como acessar e modificar essas configurações**, para garantir uma conexão mais eficiente e sem autenticação.

Não é preciso se preocupar: qualquer alteração nessas configurações **valerá apenas para este projeto**, sem afetar outros programas da sua biblioteca.

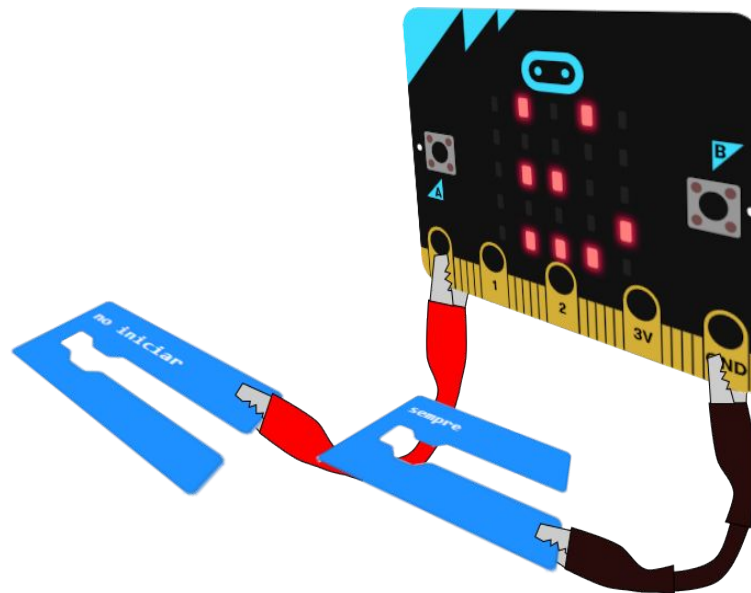
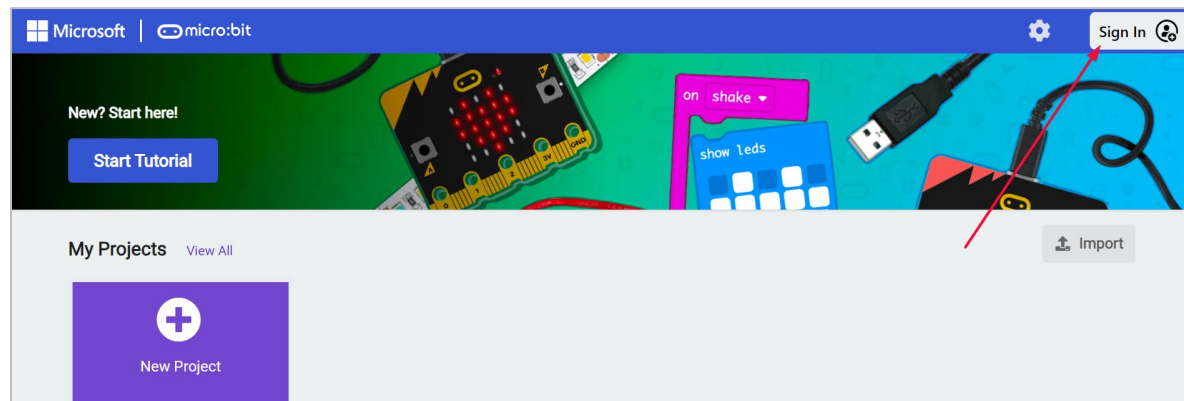


Imagem produzida pela Seduc com auxílio das ferramentas Gimp e Inkscape

- Vamos mostrar como adicionar a extensão *bluetooth* no MakeCode;
- Qual a função dos blocos e como montar um programa;
- Como fazer ajuste fino nas configurações para garantir uma conexão eficiente com o aplicativo externo.

Lembre-se de que, para acessar o MakeCode, você deve seguir o passo a passo abaixo:

1. Acesse a Sala do Futuro para entrar no MakeCode;
2. Ao entrar no MakeCode: use o e-mail institucional **@aluno.educacao** para fazer o login.



Produzido pela SEDUC-SP com © Gimp e © Inkscape



FICA A DICA

Repita esse procedimento em toda aula que o MakeCode for usado, para garantir que você esteja trabalhando no seu login.

Quando você realiza o login, garanta que seus projetos ficarão salvos para acesso futuro seu e do seu professor. Isso será fundamental para o envio do link da atividade do dia ao docente.

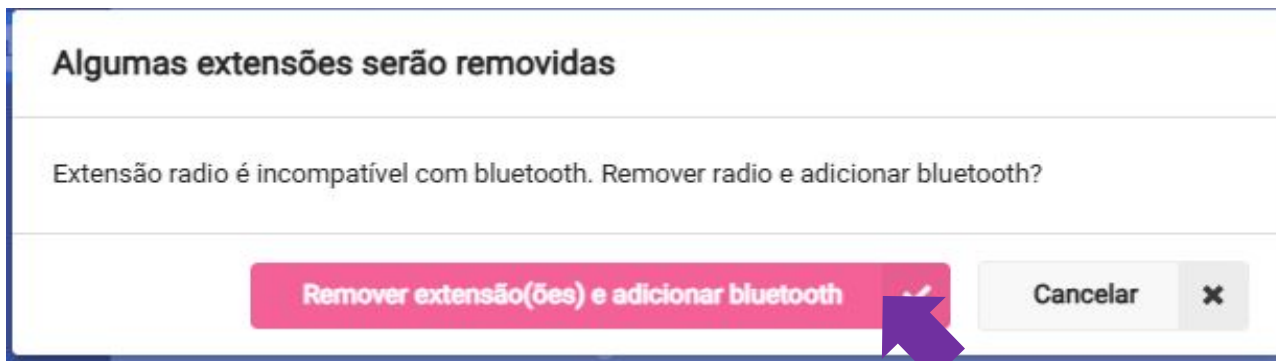
Continua



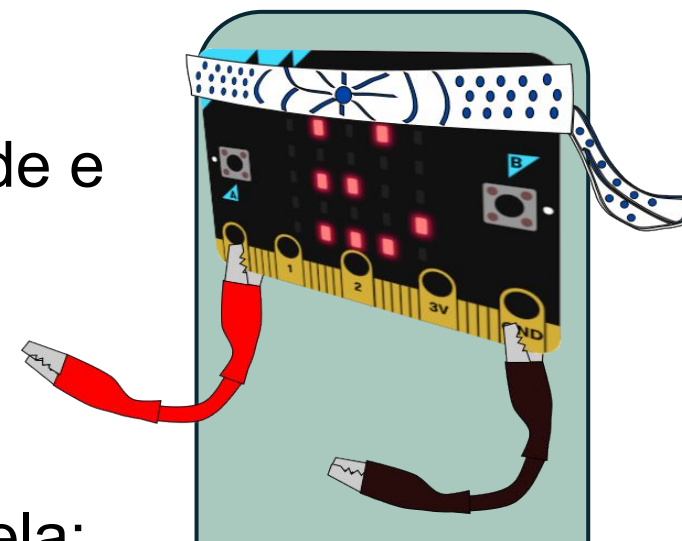
Na prática



1. Crie um programa novo no MakeCode e dê a ele o nome de **BlueTest**;
2. Adicione a extensão **bluetooth**;
3. Ao confirmar a inclusão dessa extensão, aparecerá esta imagem na tela:



4. Confirme clicando na opção “Remover extensão(ões) e adicionar *bluetooth*”.



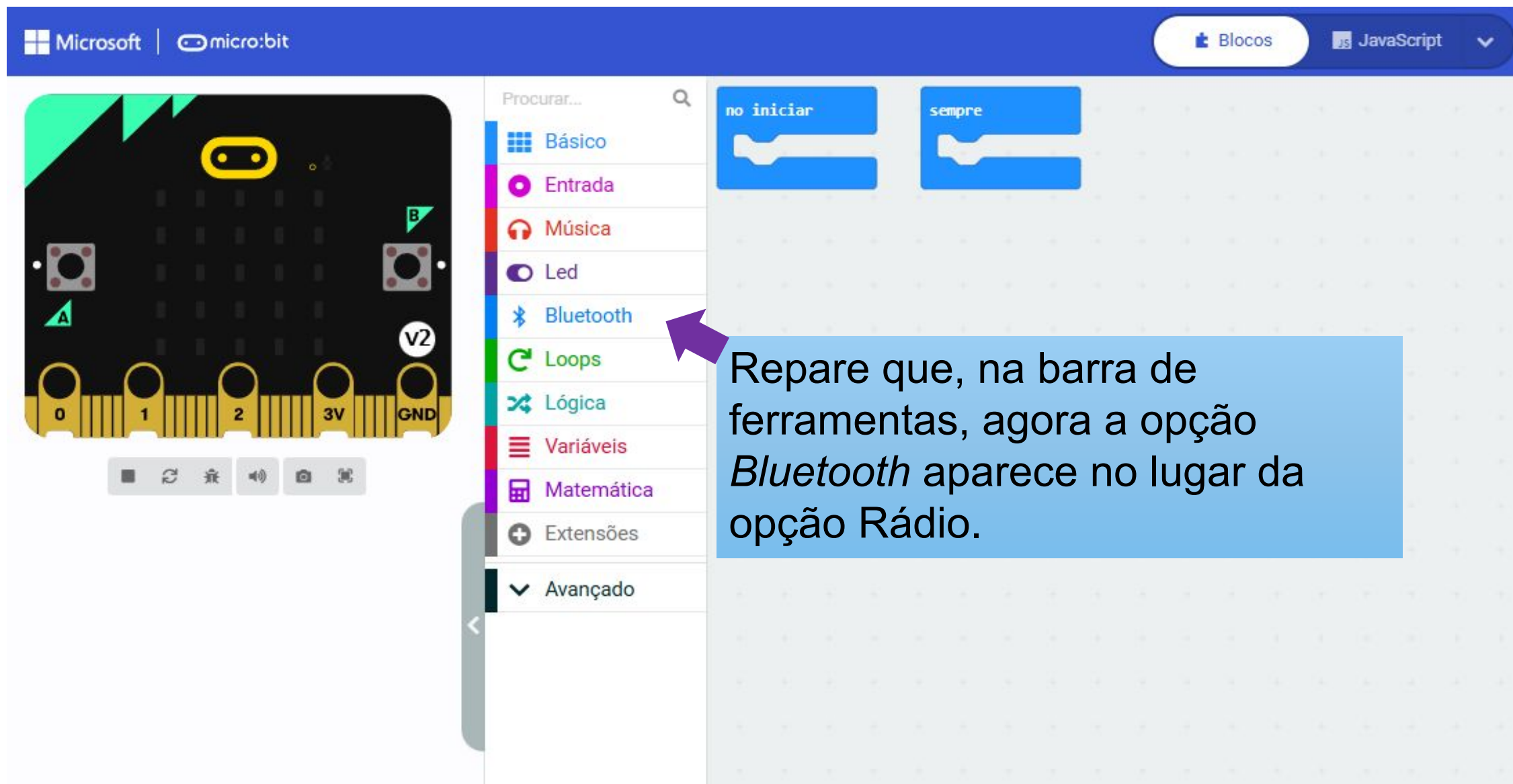
LEMBRE-SE:

Toda vez que aparecer esta imagem, você será convidado a executar sozinho um procedimento no MakeCode.

Produzido pela SEDUC-SP
com © Tinkercad



Na prática



Entendendo a função dos novos blocos



Este bloco liga a conexão *bluetooth* na placa. Deve ser adicionado no bloco **iniciar**.



Usaremos este bloco para exibir uma imagem ou número para indicar que a conexão entre o dispositivo e a placa foi realizada com sucesso.



Usaremos este bloco para exibir uma imagem ou número para indicar que a placa está desconectada.

Entenda que, quando falamos em dispositivos, esses podem ser um computador desktop, um notebook ou um celular.



Na prática

Este bloco atua como o coração do processo de comunicação entre o **micro:bit** e o aplicativo externo.

A Scratch code block for the 'uart bluetooth' extension. It is a blue block with the text 'uart bluetooth ler até' in white, followed by a dark blue dropdown menu showing 'new line ()' with a downward arrow.

É ele que recebe as mensagens enviadas pelo aplicativo **aos poucos, como se fossem partes de uma frase.**

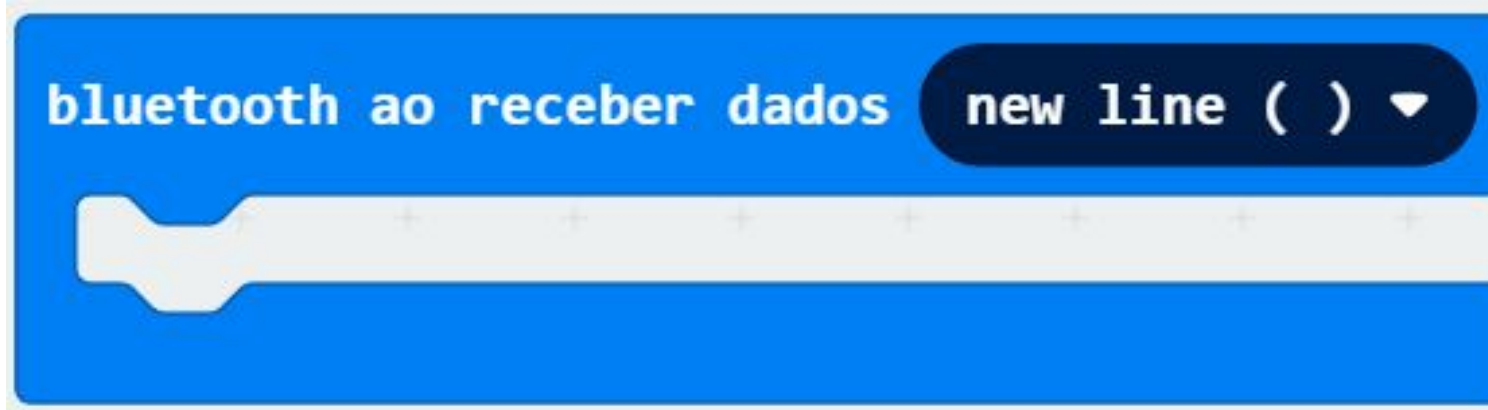
O bloco “**new line ()**” faz com que o programa **aguarde o final da mensagem**, identificado por um sinal que indica que a informação está completa.

Quando o final da mensagem é identificado, ela é **armazenada em uma variável** e pode ser usada pelo programa.

Em seguida, o micro:bit fica pronto para **receber a próxima linha de dados**, mantendo a comunicação contínua.



Na prática



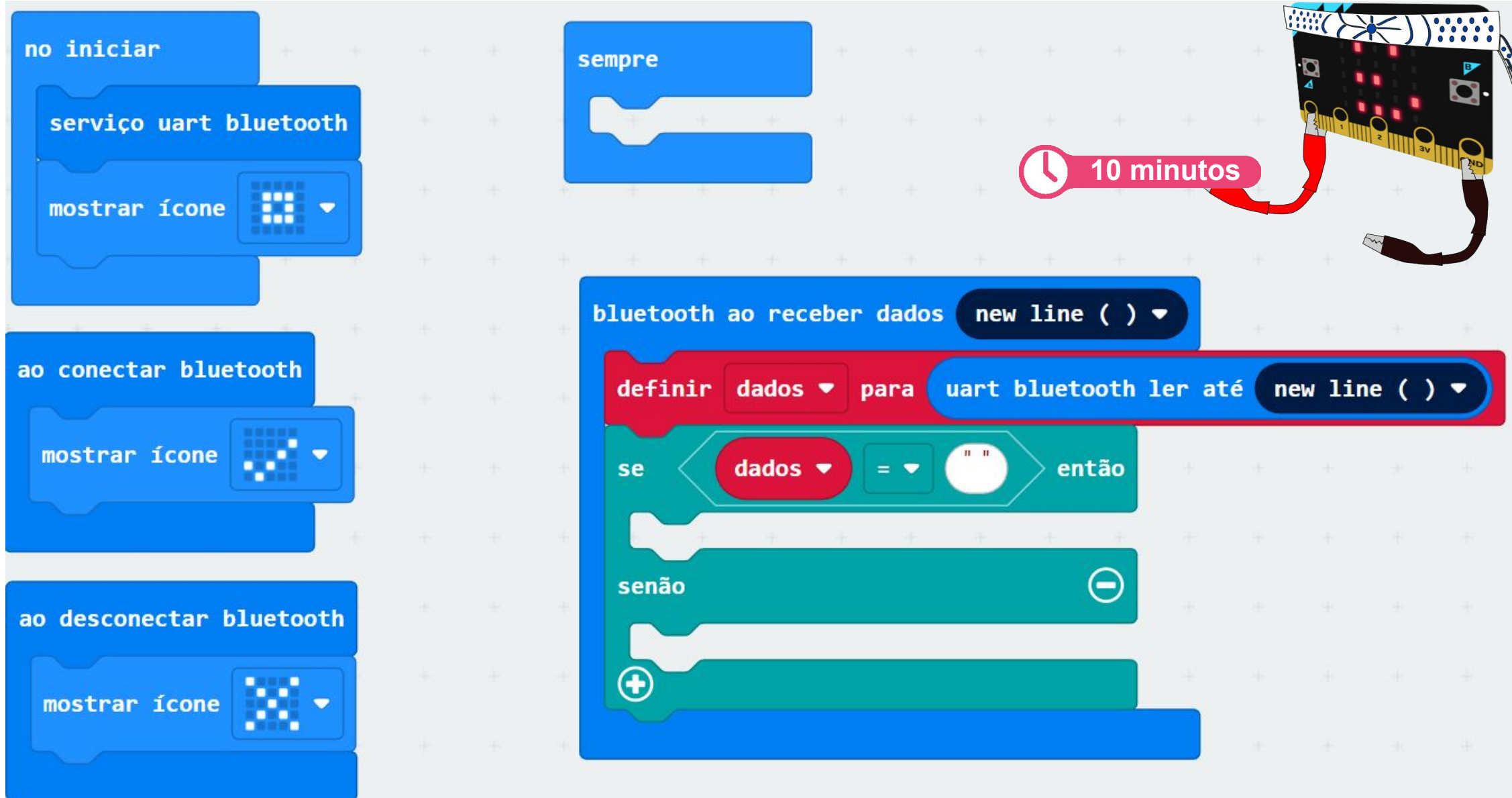
Usaremos este bloco para estabelecer como o micro:bit vai reagir aos dados recebidos. Sua lógica de funcionamento é muito parecida com a do bloco “**ao receber rádio receivedString**”.



Produzido pela SEDUC-SP com © MakeCode

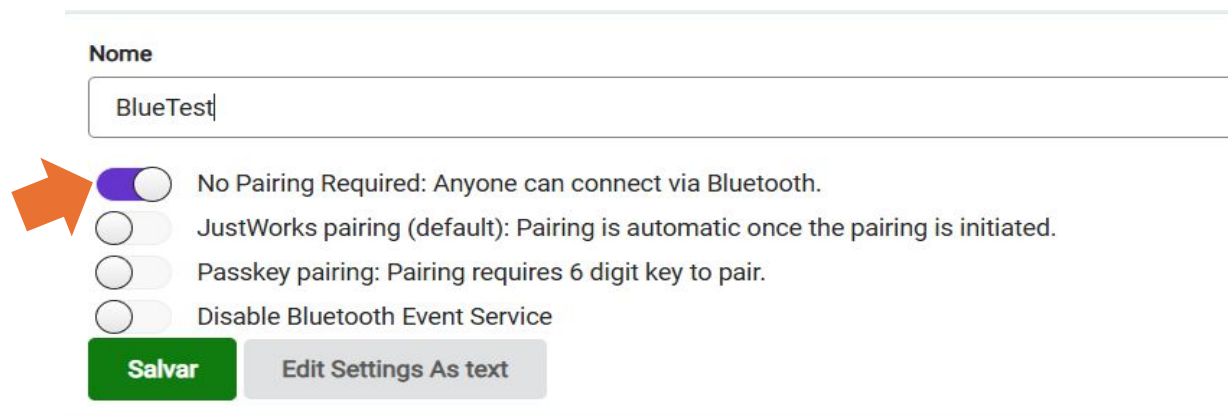
Continua





Com o esqueleto do código pronto, vamos ajustar as configurações do projeto

- Com o programa BlueTest aberto, clique no ícone de engrenagem que está localizado no canto superior direito da tela;
- Nas opções, clique em “Configurações do Projeto”;
- Ative a primeira opção:



Nome

BlueTest

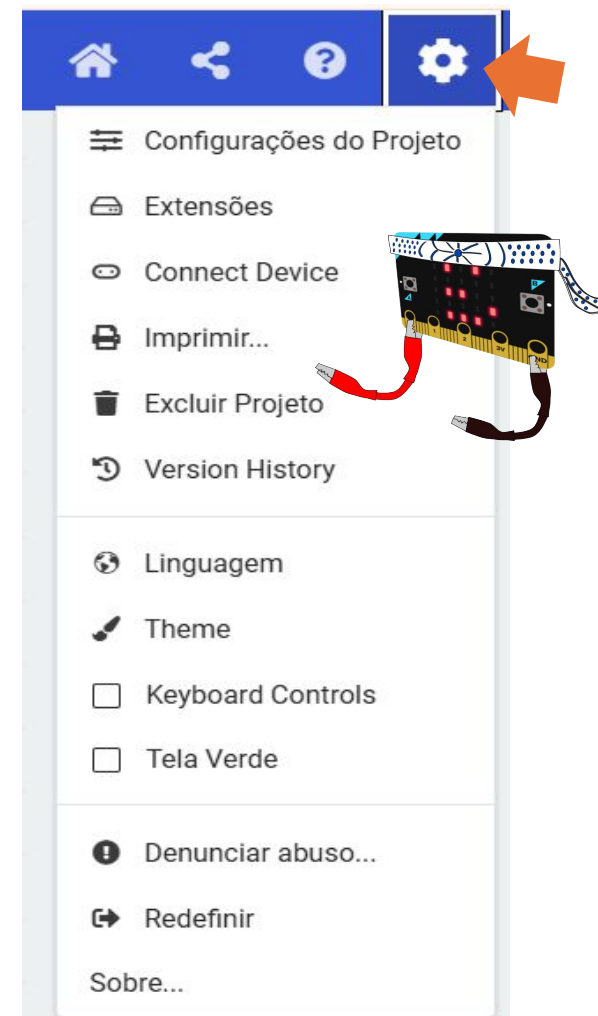
☒ No Pairing Required: Anyone can connect via Bluetooth.

☐ JustWorks pairing (default): Pairing is automatic once the pairing is initiated.

☐ Passkey pairing: Pairing requires 6 digit key to pair.

☐ Disable Bluetooth Event Service

Salvar Edit Settings As text



Agora vem o “pulo do gato”

- Clique em “***Edit Settings As text***”.

Nome

BlueTest|

☒ No Pairing Required: Anyone can connect via Bluetooth.

☐ JustWorks pairing (default): Pairing is automatic once the pairing is initiated.

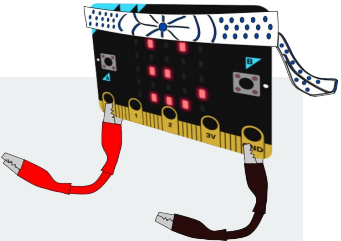
☐ Passkey pairing: Pairing requires 6 digit key to pair.

☐ Disable Bluetooth Event Service

Salvar Edit Settings As text

O que aparecerá na tela é o código (“esqueleto” padrão) de toda cópia do modelo de programa do MakeCode.

Para que a conexão sem pareamento com *bluetooth* funcione perfeitamente no micro:bit V2 (nossa placa), é necessário adicionar uma linha a esse código.



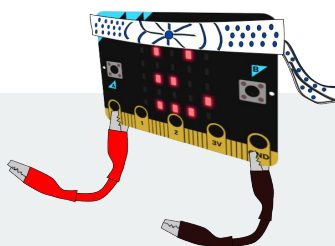
```
1 {
2   "name": "BlueTest",
3   "description": "",
4   "dependencies": {
5     "core": "*",
6     "microphone": "*",
7     "bluetooth": "*"
8   },
9   "files": [
10    "main.blocks",
11    "main.ts",
12    "README.md"
13  ],
14  "preferredEditor": "tsprj",
15  "yotta": {
16    "config": {
17      "microbit-dal": {
18        "bluetooth": {
19          "open": 1,
20          "whitelist": 0,
21          "security_level": null
22        }
23      }
24    }
25  }
26 }
27
```



Na prática

Na seção “preferredEditor”...

```
1  {
2    "name": "BlueTest",
3    "description": "",
4    "dependencies": {
5      "core": "*",
6      "microphone": "*",
7      "bluetooth": "*"
8    },
9    "files": [
10     "main.blocks",
11     "main.ts",
12     "README.md"
13   ],
14   "preferredEditor": "tsprj",
15   "yotta": {
16     "config": {
17       "microbit-dal": {
18         "bluetooth": {
19           "open": 1,
20           "whitelist": 0,
21           "security_level": null
22         }
23       }
24     }
25   }
26 }
27
```



```
"preferredEditor": "blocksprj",
"yotta": {
  "config": {
    "microbit-dal": {
      "bluetooth": {
        "open": 1,
        "pairing_mode": 0,
        "whitelist": 0,
        "security_level": null
      }
    }
  }
}
```

Adicione a linha “pairing_mode”: 0 depois da linha “open”: 1; feita essa alteração, é só voltar para visualização em blocos.

Pronto! Ajuste concluído.

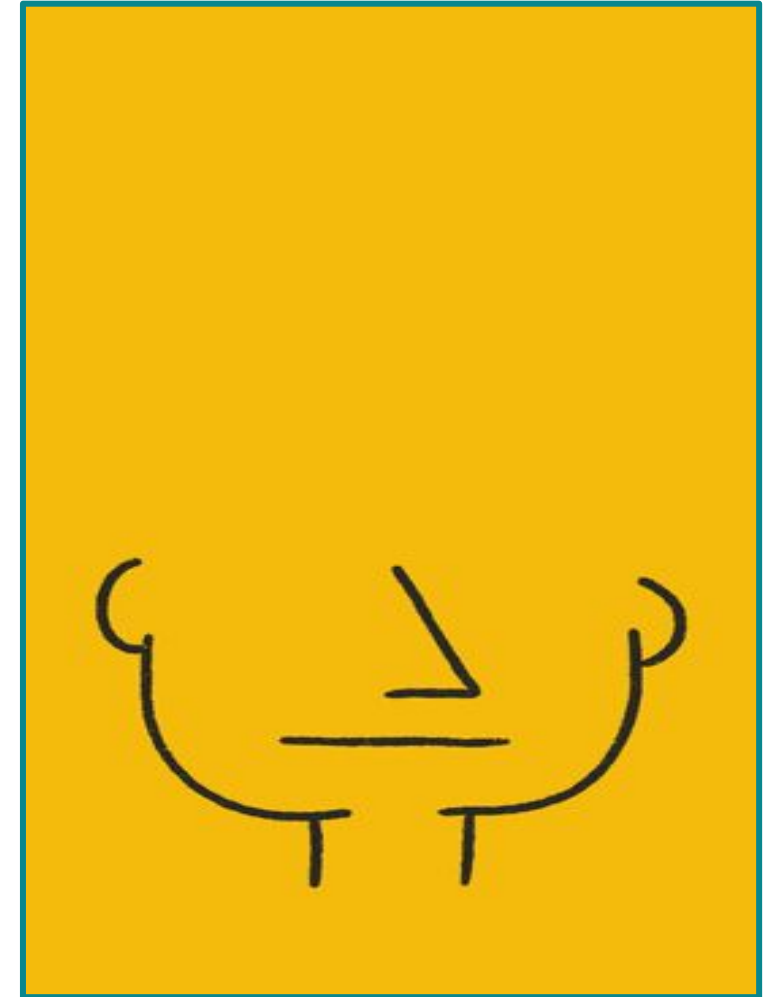


Na próxima aula, vamos aprender como usar e testar esse código usando um aplicativo de IA. Então, não se esqueça de salvar este programa.

Encerramento

Analizando o código que criamos hoje, explique com suas próprias palavras: **o que acontece no micro:bit...**

1. quando o *bluetooth* conecta?
2. quando o *bluetooth* desconecta?
3. quando uma mensagem chega pelo aplicativo?



© Giphy



© Giphy

- Que existem diferentes formas de construir sistemas de IA;
- IA simbólica funciona com regras fixas e *Machine Learning* funciona a partir de exemplos e padrões;
- O micro:bit não “faz IA”, mas é essencial para transformar decisões em ações físicas.

Referências

BRASIL. Conselho Nacional de Educação. Câmara de Educação Básica. Parecer CNE/CEB nº 2/2022, de 17 de fevereiro de 2022. **Normas sobre Computação na Educação Básica** – Complemento à Base Nacional Comum Curricular (BNCC). Brasília, DF: CNE/CEB, 2022.

KAHNEMAN, D. **Thinking, fast and slow**. New York: Farrar, Straus and Giroux, 2011.

LEMOV, Doug. **Aula nota 10 3.0**: 63 técnicas para melhorar a gestão da sala de aula / Doug Lemov; tradução: Daniel Vieira, Sandra Maria Mallmann da Rosa; revisão técnica: Fausta Camargo, Thuinie Daros. 3. ed. Porto Alegre: Penso, 2023.

LÉVI-STRAUSS, C. **The Savage Mind**. University of Chicago Press, 1966.

SILVA, R. C. da; ZEJNILOVIC, L.; OLIVEIRA, P. Improvisation and bricolage: Similarities and differences... in The Routledge Companion to Improvisation in Organizations.

MAYER, R. E. **Multimedia learning**. Cambridge: Cambridge University Press, 2001.

PAPERT, S. **Mindstorms**: children, computers, and powerful ideas. New York: Basic Books, 1980.

ROSENSHINE, B. “Principles of instruction: research-based strategies that all teachers should know”. In: **American Educator**, v. 36, n. 1, Washington, 2012. p. 12-19. Disponível em: <https://www.aft.org/ae/spring2012>. Acesso em: 21 ago. 2025.



Referências

SÃO PAULO (Estado). Secretaria da Educação. **Currículo Paulista**: etapa Ensino Médio, 2019. Disponível em:
<https://efape.educacao.sp.gov.br/curriculopaulista/wp-content/uploads/2020/08/CURR%C3%8DCULO%20PAULISTA%20etapa%20Ensino%20M%C3%A9dio.pdf>. Acesso em: 12 ago. 2024.

SWELLER, J. **Cognitive load theory**. New York: Springer, 1988.

VYGOTSKY, L. S. **Mind in society**: the development of higher psychological processes. Cambridge: Harvard University Press, 1978.

Identidade visual: imagens © Getty Images

Para professores

Slide 2

Habilidade:

(EM13CO02) Explorar e construir a solução de problemas por meio de refinamentos, utilizando diversos níveis de abstração desde a especificação até a implementação.

(EM13CO07) Compreender as diferentes tecnologias, bem como equipamentos, protocolos e serviços envolvidos no funcionamento de redes de computadores, identificando suas possibilidades de escala e confiabilidade.

Professor(a),

Ao longo dos últimos anos, estudantes e professores vêm construindo, passo a passo, habilidades relacionadas à programação, à prototipagem e à cultura *maker*.

Sabemos que esse percurso não é linear e que cada turma chega às aulas de Robótica com **níveis diferentes de familiaridade e segurança**.

Reconhecemos também o papel ativo dos professores, que se dedicam a estudar, experimentar e adaptar propostas para que a aprendizagem aconteça de forma significativa.

Este material foi desenvolvido para **dialogar com essa trajetória**, funcionando como apoio para revisão, aprofundamento e ampliação de conceitos e práticas da Robótica educacional.

O objetivo não é começar do zero, mas **avançar a partir do que já sabemos**, respeitando ritmos diferentes e valorizando o percurso construído até aqui.

Sugerimos três orientações-guia que valem para **todas as aulas**:

- 1. Não tente “explicar tudo”.** O material já faz isso. Procure **costurar sentido**, e não se aprofunde em questões técnicas sem necessidade.
- 2. Sempre fale em termos de comportamento do sistema**, nunca use expressões do tipo “como a IA pensa”. Isso evita antropomorfização e confusão conceitual.
- 3. Use exemplos do cotidiano dos estudantes antes de exemplos técnicos.** O micro:bit entra como *materialização*, não como ponto de partida.

Robótica educacional não é apenas programação, nem apenas montar protótipos.

No currículo, a Robótica educacional se apoia em **três pilares complementares**:

Mecânica – estrutura, movimento, articulações, forças, limitações físicas;

Eletrônica – sensores, atuadores, energia e sinais;

Programação – lógica, controle, tomada de decisão.

O objetivo da disciplina **não é formar técnicos especialistas**, mas desenvolver:

- pensamento computacional;
- modelagem de problemas do mundo real;
- integração entre ideias abstratas e objetos concretos.

Portanto, uma aula pode enfatizar mais um pilar do que o outro sem deixar de ser Robótica.

Na ausência de materiais estruturados, nossos protótipos seguem o princípio da robótica com sucata e são feitos para dar sentido à programação.

Temos que evitar que professores e estudantes **associem equivocadamente robótica apenas à programação**, já que, no contexto escolar, a **criação de protótipos** introduz conceitos de **mecânica de forma indireta**.

Assim, tanto professores que não são da área de exatas como estudantes aprendem sobre limites físicos dos materiais, movimento, peso, atrito e estrutura mecânica **sem formalização científica e matemática pesadas**.

Sem protótipo, a programação vira exercício isolado de computação física e o conteúdo da aula passa a ser confundido com sua coirmã (programação), ministrada pela Alura.

Se o estudante está programando algo que se move, responde ou se transforma no mundo físico, ele está aprendendo Robótica educacional – mesmo que, naquele momento, o foco não seja eletrônica ou código textual.

Os pilares da Robótica aparecem de forma progressiva não simultânea no conteúdo do Ensino Médio.

O desenho atual das aulas prevê que, para **1ª e 2ª Séries**, os estudantes trabalharão com programação em blocos + prototipagem → introdução integrada de lógica e mecânica.

Já a partir da 3ª Série:

aprofundamento em **eletrônica**, sensores, comunicação e sistemas mais complexos.

Essa organização é **pedagogicamente intencional, pois visa:**

- reduzir carga cognitiva sobre estudantes e professores;
- respeitar a diversidade de formação dos professores;
- garantir que todos os estudantes construam repertório comum antes do aprofundamento técnico.

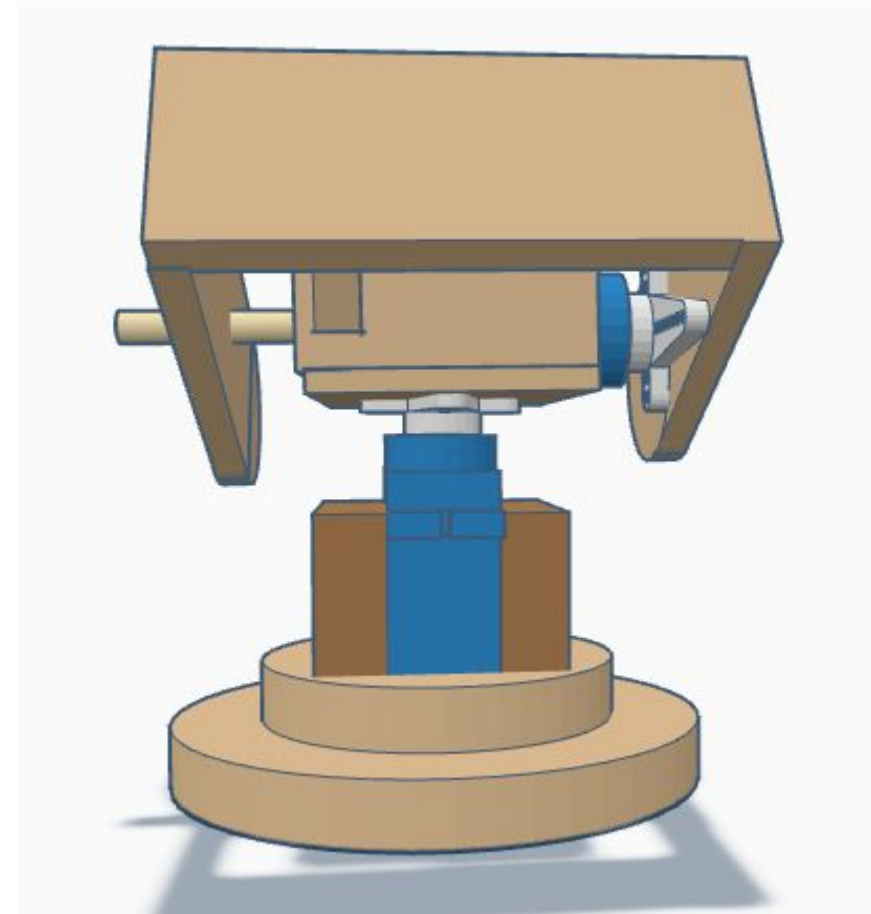
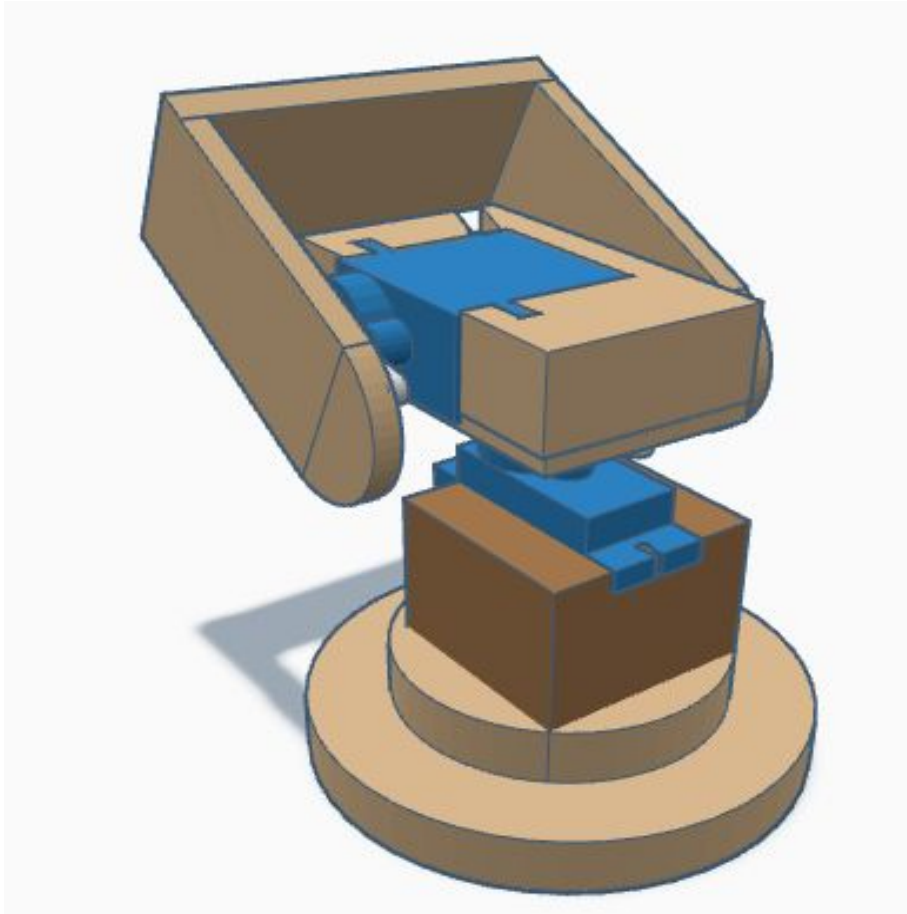
Risco comum ao abordar os diferentes tipo de IA:

Não trate a IA simbólica como “IA errada” ou ultrapassada. Apresente-a como **adequada para certos problemas**, mas limitada para outros.

Nesta aula, defendemos que esse tipo de sistema funciona muito bem quando o mundo é previsível, que o problema é quando ele deixa de ser. Fazemos isso no sentido de preparar o terreno para falar sobre IA baseada em dados sem hierarquizar moralmente.

Slides 9, 10, 11 e 12

Possibilidade de projeto – apenas para ilustrar como é possível reutilizar o que o estudante já aprendeu.



Tarefas de Robótica

Caro(a) professor(a),

Algumas aulas possuem tarefas indicadas. No [documento de passo a passo](#), encontram-se instruções para postagem da **atividade de aula** para seus estudantes (se houver). Orientamos que a postagem seja feita **antes ou durante a aula** para que o estudante possa **registrar** a entrega da atividade **durante a aula**.

O objetivo deste envio é que o estudante **registre** na Sala do Futuro, a atividade realizada em sala de aula, para acompanharmos o **engajamento** com as aulas de Robótica, e possibilitar a você, docente, avaliar a **aprendizagem e a evolução do estudante**.

Orientamos também que a atividade seja postada sem prazo de término especificado. Assim, caso estejam com dificuldades em acessar a Sala do Futuro ou a internet no dia, o estudante poderá finalizar a tarefa posteriormente.

Destaque



Importante: nem todas as aulas do bimestre possuem tarefas!

Para saber para quais aulas estão previstas tarefas, consulte a [planilha de apoio do componente!](#)

Para professores

Olá, docente! 🙌 Este material contém algumas ferramentas e recursos que visam tornar a aula mais interativa, acessível e interessante.

Recomendamos que utilize sempre o modo apresentação do Power Point.

Este material foi organizado para que você consiga desenvolver a aula apoiado no PDF; contudo, a experiência será mais rica e mais profunda com os recursos que o Power Point apresenta.

Outro recurso importante é o Complemento à BNCC de Computação. Recomendamos a leitura!

Além do Material Digital, disponibilizamos materiais com um passo a passo de **como fazer a codificação, o download da programação na placa e/ou montar o protótipo** para apoiar a condução e o planejamento da aula.

Os links para os vídeos estão disponíveis no repositório (CMSP) e no YouTube.

Destaque



Apoie-se em nossos recursos! 😊

[!\[\]\(c50c8b7b2cc2cf9ff925edec0ee94c0d_img.jpg\) Tutoriais 6º Ano](#)

[!\[\]\(6a9b39b98eb945faa14c645ec99e4eaa_img.jpg\) Tutoriais 7º Ano](#)

[!\[\]\(9c2e8d1b5bd77cb5c9f83b7a9cff79fd_img.jpg\) Tutoriais 8º Ano](#)

[!\[\]\(e3275251d0893157c3584e20c81dc3ba_img.jpg\) Tutoriais 9º Ano](#)

[!\[\]\(f60b7a900783ac3fd531bfd9c111be6d_img.jpg\) Tutoriais 1ª Série do Ensino Médio](#)

[!\[\]\(f1c5da15572e3e09d343161be98f508d_img.jpg\) Tutoriais 2ª e 3ª Séries do Ensino Médio](#)

[!\[\]\(235bfe13ebf007ce2eea9e689707fac7_img.jpg\) Lista de Reprodução: Kit de Robótica](#)

[!\[\]\(eabd9f9ababee93effadc3b380fe65fd_img.jpg\) Lista de Reprodução: Orientações adicionais](#)

[!\[\]\(83bbbd261710c59db0214aa27b2edc0d_img.jpg\) Manual: Kit de Robótica](#)

Caso não consiga acessar algum dos links acima, eles também estão listados na seguinte planilha on-line: [Links e Recursos de Robótica](#).



**GOVERNO DO ESTADO
DE SÃO PAULO**