

Guião de apoio: Robô controlado por gestos de IA (Arduino + PictoBlox)

Montagem do Robô

1. Materiais Necessários:

- *Arduino Uno R3*
- *Driver de Motor L298N*
- 2 motores *DC* + chassi com rodas
- *Protoboard + jumpers*
- Módulo *Bluetooth HC-05*
- Bateria 9V + conector

2. Montagem do Robô:

1. **Montagem do Chassi:** Fixar as rodas ao chassi e ligar os motores.

2. Instalação dos Componentes:

- Instalar o *driver L298N* ao *Arduíno*
 - IN1 → pino 8
 - IN2 → pino 9
 - IN3 → pino 10
 - IN4 → pino 11
- Ligar o módulo *Bluetooth HC-05*:
 - TX → RX *Arduino*
 - RX → TX *Arduino*

3. **Ligar a bateria apenas no final, por segurança.**

Nota: Desliga a bateria antes de mexer nas ligações.

3. Teste de Conectividade

- Usar o *PictoBlox* para enviar sinais simples ao *Arduino* e verificar se os motores funcionam corretamente.

4. Treinar o modelo de IA

1. Criar conta ou iniciar sessão em *PictoBlox*.
2. Ativar as extensões de IA e *Machine Learning*.
3. Ligar ao *Teachable Machine*.

4. Criar um novo projeto de imagens.
5. Usar a *webcam* ou câmara do telemóvel para recolher exemplos de gestos (mínimo 50 imagens por gesto, com boa iluminação).
 - Exemplo:
 - Mão para a frente → robô avança
 - Mão inclinada para a esquerda → robô vira
6. Treinar o modelo e exportar:
 - *Export Model* → *Upload my Model* → copiar o link do *Hosted Model*.
7. No *PictoBlox*, abrir a extensão *Machine Learning* e colar o link no bloco *Load a Model*.

5. Programar o robô no *PictoBlox*

1. Carregar o modelo treinado:

- Usar o bloco *load image model from url () and set name ()*.

2. Associar gestos a ações:

- Programar os comandos para o robô avançar, virar, parar, etc.
- A ligação ao *Arduino* é feita por *Bluetooth*.

3. Criar blocos personalizados:

- *Go Straight, Turn Left, Turn Right, Brake*.
- Usar a aba “Meus Blocos” para criar funções para cada movimento.

4. Iniciar o programa:

- Usar o bloco *when clicked* para começar a execução.
- Ativar a câmara e o modelo de IA.

5. Ciclo principal:

- Usar o bloco *Forever* para verificar continuamente os gestos.
- Para cada gesto identificado, executar o bloco correspondente (*Go Straight, Turn Left, etc.*).

6. Testes e resolução de problemas

- Testar cada gesto e confirmar se o robô responde corretamente.
- Se não funcionar:
 - Verificar ligação *Bluetooth*.
 - Confirmar se o modelo foi bem treinado (mínimo 50 imagens por gesto).
 - Rever a associação dos blocos às ações.

Nota: Este guião pode ser adaptado conforme o nível dos participantes. Para mais detalhes técnicos ou códigos, consultar o anexo.

1. Treinar e usar o modelo de IA no PictoBlox

1. Aceder às ferramentas

- Acede ao **PictoBlox** e cria uma conta ou faz *login* → <https://pictoblox.ai>
- Vai à aba **Extensões** e ativa **IA e Machine Learning**.
- O **PictoBlox** liga-se ao **Teachable Machine**, onde vais treinar o modelo. → <https://teachablemachine.withgoogle.com/>

2. Criar o modelo de gestos

- No **Teachable Machine**, cria um novo projeto:
Image Project → *Standard image model*.
- Usa a **webcam** do computador ou a **câmara do telemóvel** para recolher exemplos de gestos.

Exemplos de gestos:

- **Mão para a frente** → o robô avança.
- **Mão inclinada para a esquerda** → o robô vira.
- Recolhe pelo menos **50 imagens** por gesto
Quanto mais imagens e melhor a iluminação, maior a precisão.

3. Treinar e exportar o modelo

- Clica em **Train Model** para iniciar o treino.
- Após o treino, exporta o modelo:
Export Model → **Upload my Model**
Copia o **link do Hosted Model**.

4. Carregar o modelo no PictoBlox

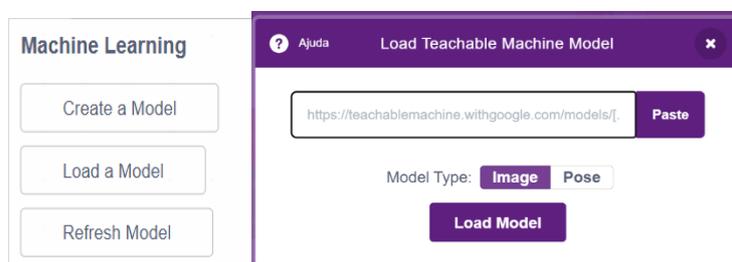
- No **PictoBlox**, abre a extensão **Machine Learning**.
- Usa o bloco:
load image model from url () and set name ()
- Cola o **link** do modelo treinado na caixa indicada.
- Se aparecer erro ao carregar, verifica a ligação à Internet e se o **link** está completo.

5. Reconhecimento em tempo real

- Com o modelo carregado, o **PictoBlox** começa a reconhecer os gestos em tempo real através da câmara.
- Estes gestos serão usados para controlar o robô, associando cada gesto a uma ação (avançar, virar, parar).

2. Ligar o modelo ao *PictoBlox* e programar o robô

- Depois de treinar o modelo no **Teachable Machine** <https://teachablemachine.withgoogle.com/>, copia o **link do modelo treinado**.
- No **PictoBlox**, <https://pictoblox.ai>, abre a extensão **Machine Learning** → escolhe a opção **Load a Model**.
- Cola o **link** do modelo na caixa indicada e carrega em **Load Model**.
 - O bloco chama-se exatamente **load image model from url () and set name ()**
 - Nota: Se aparecer erro ao carregar, confirma a ligação à internet e se o **link** do modelo está completo.



- Agora o **PictoBlox** reconhece os teus gestos em tempo real através da câmara.
- Em seguida, programa no **PictoBlox** os comandos que o robô vai executar. A ligação ao **Arduino** é feita por **Bluetooth**, para que cada gesto controlado seja transformado numa ação (avançar, virar, parar).

Etapas seguintes

1. Treino do Modelo:

- Utilizar uma webcam ou câmara de um *smartphone* para capturar diferentes gestos.
- Definir os gestos para ações específicas (avançar, parar, girar).
- Testar a precisão do modelo e ajustar conforme necessário.

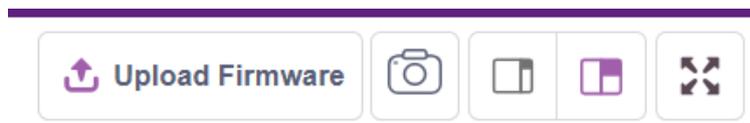
2. Programação do Arduino:

- Enviar comandos via **Bluetooth** para controlar a direção dos motores.
- Garantir que o robô está a responder de forma fluida aos gestos reconhecidos.

3. Programar os movimentos do robô no PictoBlox

Passo 1 – Selecionar a placa evive

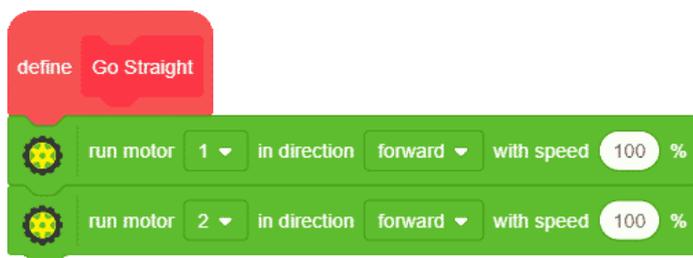
- No *PictoBlox*, vai ao menu acima do “palco” e seleciona a placa evive.
 - Se for a primeira vez, clica em **Upload Firmware para carregar o firmware**.



Nota: O carregamento pode demorar alguns segundos. Só depois consegues usar os blocos de motores.

Passo 2 – Criar blocos personalizados

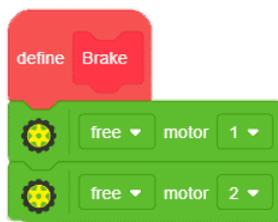
- Vai à aba **Meus Blocos** → clica em **Criar um bloco**.
- Primeiro, faz a função para fazer o robô andar para frente, escrevendo **Go Straight** no nome e clica em OK.
- Verás um bloco de chapéu na área de programação.
- Coloca o **motor de execução (1) na direção (1) com velocidade (1) %** do bloco **atuadores**.
- Duplica o bloco e coloca-o abaixo do outro bloco e escolhe **2** no primeiro menu suspenso.



- Da mesma forma, faz os blocos personalizados para virar à esquerda, virar à direita e travar, um por um.
- Duplica os dois blocos de motor e coloca-os abaixo dos blocos personalizados (ver imagem abaixo – **bloco Turn Left e bloco Turn Right**).
- Faz algumas alterações nesses blocos.
 - No bloco **Turn Left**, seleciona **reverse** no segundo menu suspenso do primeiro bloco.
 - No bloco **Turn Right**, seleciona **forward** no segundo menu suspenso do segundo bloco.

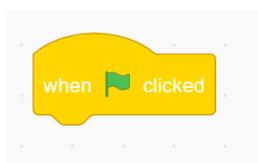


- No bloco **Brake**, arrasta e solta o bloco **free motor 1** da aba atuadores.
- Duplica-o e escolhe 2 do segundo bloco.

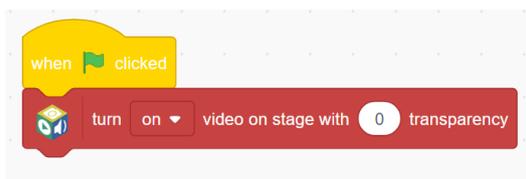


Passo 3 – Iniciar o programa

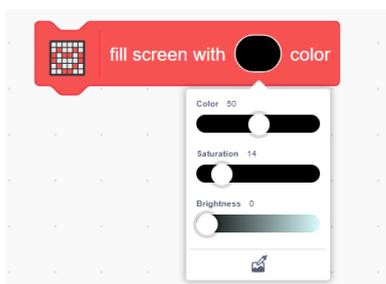
- Vai à aba **Events** → escolhe o primeiro bloco **when clicked** para que a programação comece a partir do momento que a bandeira verde seja clicada.



- De seguida, na aba **Machine Learning**, com o modelo do *Teachable Machine* carregado, arrasta o bloco **turn on video on stage with 0 transparency**, para encaixar no bloco da bandeira verde.

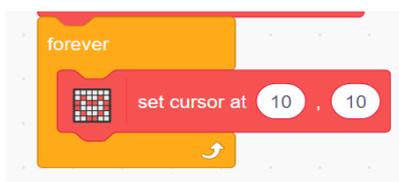


- Na aba **display** seleciona **fill screen with color**, arrasta o cursor **brigness** (brilho) para 0 e a cor ficará preta.



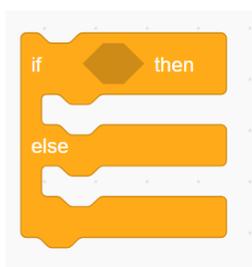
Passo 4 – Criar o ciclo principal

- Na aba **Control**, arrasta o bloco **Forever** de modo a encaixar no bloco anterior.
- Dentro deste bloco adiciona o bloco **set cursor at** que fica na aba **display** e seleccione 10, 10.

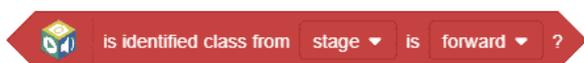


Passo 5 - Programar condições para os gestos

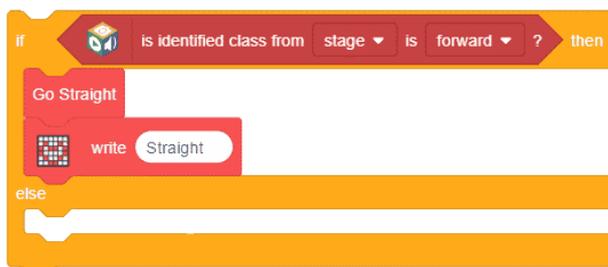
- Ainda dentro do bloco **Forever**, coloca um bloco **if-else** da aba **control**.



- Arrasta o bloco **is identified class from (stage) is (forward)** e associa ao gesto treinado.
- Seleciona sempre **stage** e depois o gesto correto.
- Se for **forward**, chama o bloco **Go Straight** e usa **write()** para mostrar **Straight** no ecrã.

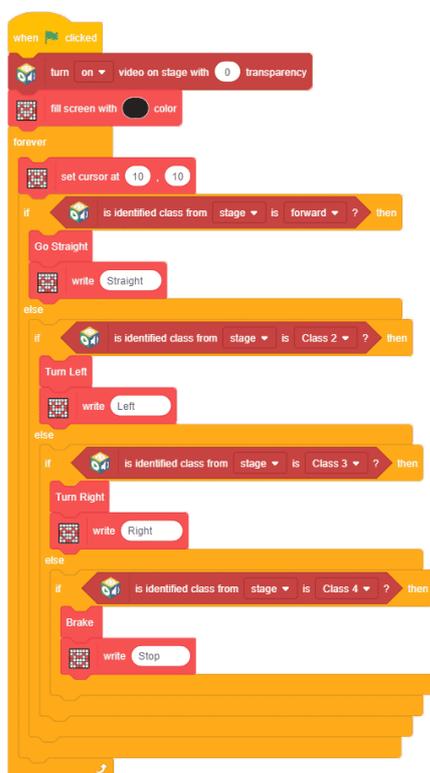


- Coloque o bloco **Go Straight** no braço **if**.
- Configure a imagem seguinte diretamente no ecrã do **evive** usando o bloco **write()** da aba de exibição.



Passo 6 – Adicionar outros gestos

- Duplica o bloco **if** e coloca-o no braço **else**.
- Selecciona **left** no bloco **if** e substitui o bloco **go straight** pelo bloco **turn left**.
- Escreve **Left** no **display** do **evive**.
- Repete o mesmo processo para a direita e parar.



7. Executar o programa

- O programa agora está completo, clica na **bandeira verde** para executá-lo.
- Testa cada gesto e confirma se o robô responde corretamente.
- Com isso, o teu robô controlado por gestos de IA está pronto.

4. Testes

- Se o robô não responder corretamente, verifica: (a) se o **Bluetooth** está ligado, (b) se o modelo foi bem treinado (min. 50 imagens por gesto), (c) se os blocos foram associados às ações certas.



Os conteúdos abordados neste documento encontram-se sob a licença Creative Commons. Utilização Não Comercial. BY - Os créditos devem ser dados ao autor. NC – Não são permitidos usos comerciais. SA – As adaptações devem ser partilhadas nos mesmos termos.

AUTOR(ES)	DATA
Center of Digital Inclusion (CDI) Portugal	Outubro/2025