

# Robótica Educacional

Módulo 3



Aula

28

PalmaBot - III

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**

Darice Alessandra Deckmann Zanardini

**Validação de Conteúdo**

Cleiton Rosa

**Revisão Textual**

Kellen Pricila dos Santos Cochinski

**Projeto Gráfico e Diagramação**

Edna do Rocio Becker

2024

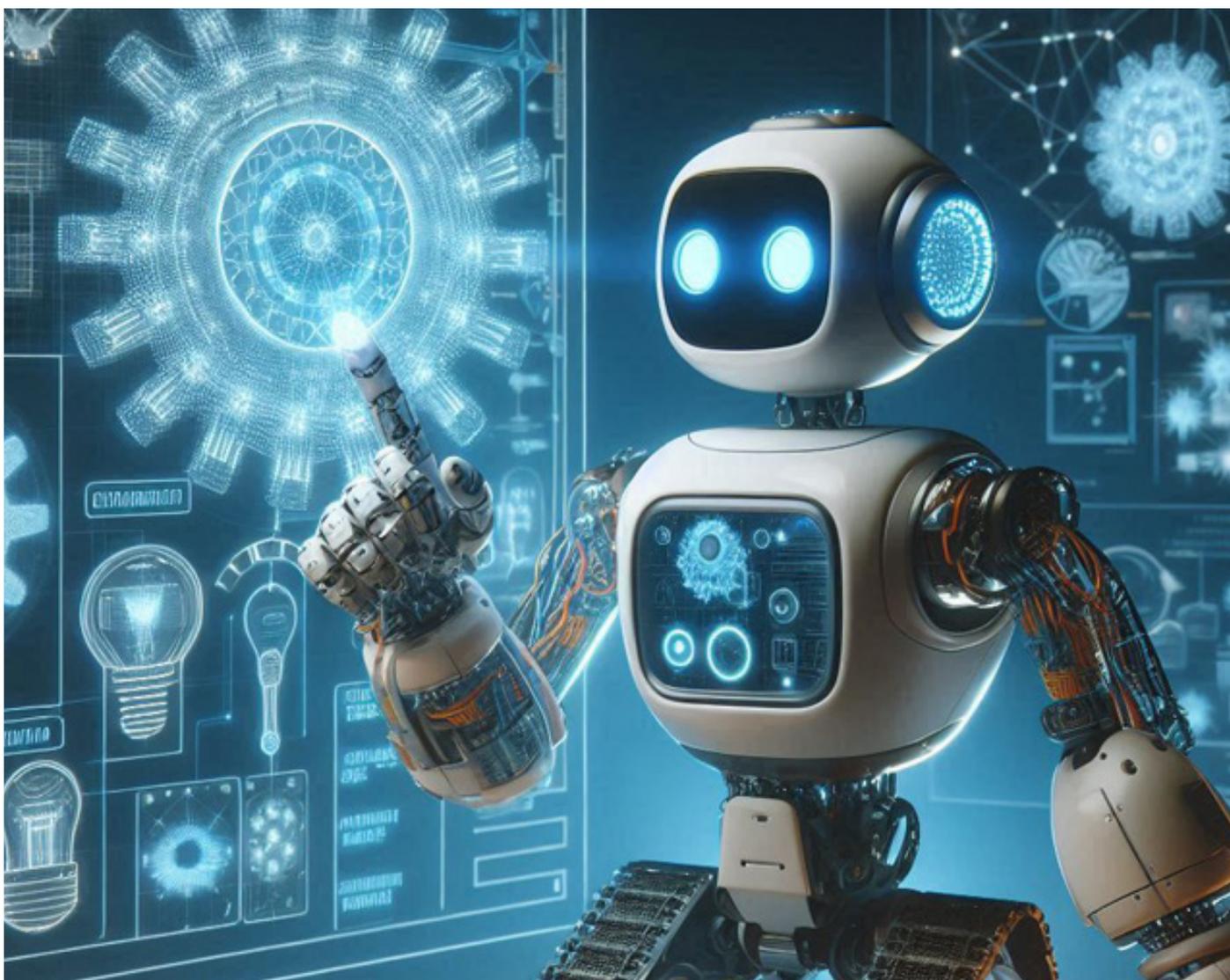
# Sumário

<b>Introdução</b>	<b>2</b>
<b>Objetivos desta aula</b>	<b>3</b>
<b>Roteiro da aula</b>	<b>3</b>
1. Contextualização	4
2. Feedback e finalização	10
<b>Referências</b>	<b>10</b>

## Introdução

Na jornada incrível pela Robótica, nos inspiramos, desenvolvemos e aprimoramos projetos. Nas **Aulas 26 e 27 – PalmaBot [Parte I e II]**, desenvolvemos um robô capaz de interagir com o ambiente através da detecção de palmas, utilizando um Arduino Uno, ponte H L298N e um sensor de som sobre o chassi 2WD.

Com o desenvolvimento da programação do robô até a montagem física e os primeiros testes, tivemos mais um exemplo que explora a Robótica como uma ferramenta para criar soluções inovadoras, aprimorando projetos e permitindo novas formas de aplicação e controle. A interação do PalmaBot com o ambiente, neste caso através do som, é fundamental para a autonomia do robô.

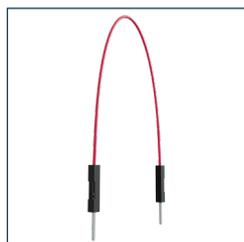
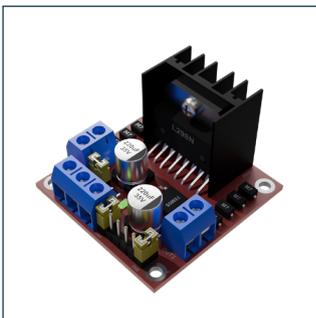


## Objetivos desta aula

- Promover exibição de recursos atribuídos ao robô com interação por som.
- Percorrer labirinto ou utilizando palmas para guiar o robô.

## Lista de materiais

- Notebook ou computador;
- 1 Arduino Uno;
- Kit chassi 2WD;
- Ponte H L298N;
- Sensor de som;
- Jumpers;
- Fonte de alimentação/bateria.



## Roteiro da aula

### 1. Contextualização

No desenvolvimento do projeto PalmaBot, os sensores são apresentados como os meios pelos quais o robô percebe o ambiente, coletando informações que permitem a tomada de decisões. Neste projeto, o foco inicial é no sentido da audição e criamos funções específicas para controle de movimento e detecção de palmas pelo sensor de som.

- Funções de movimento:
  - **moverFrente()**: move o robô para frente.
  - **moverRe()**: move o robô para trás.
  - **moverDireita()**: gira o robô para a direita.
  - **moverEsquerda()**: gira o robô para a esquerda.
  - **pararRobo()** para o robô.
- Funções de controle:
  - **iniciarEscuta()**: inicia o modo de escuta após detectar a primeira palma.
  - **contarPalmas()**: conta o número de palmas detectadas, utilizando de-bounce para evitar contagens duplicadas.
  - **executarMovimento()**: executa ações com base na contagem de palmas, chamando as funções de movimento correspondentes.

No decorrer da programação, aplicamos a função **millis()** destacando-a como essencial para o controle de tempo sem bloquear a execução do programa e, pelos testes iniciais na última aula, verificamos a resposta do robô aos comandos de palmas para, se necessário, ajustar a sensibilidade do sensor no ambiente em que ele se movimentará.

Que tal conferir o desempenho do seu robô percorrendo um labirinto? Nesta aula, desafiamos você e seus colegas a colocarem seus robôs em uma disputa emocionante!

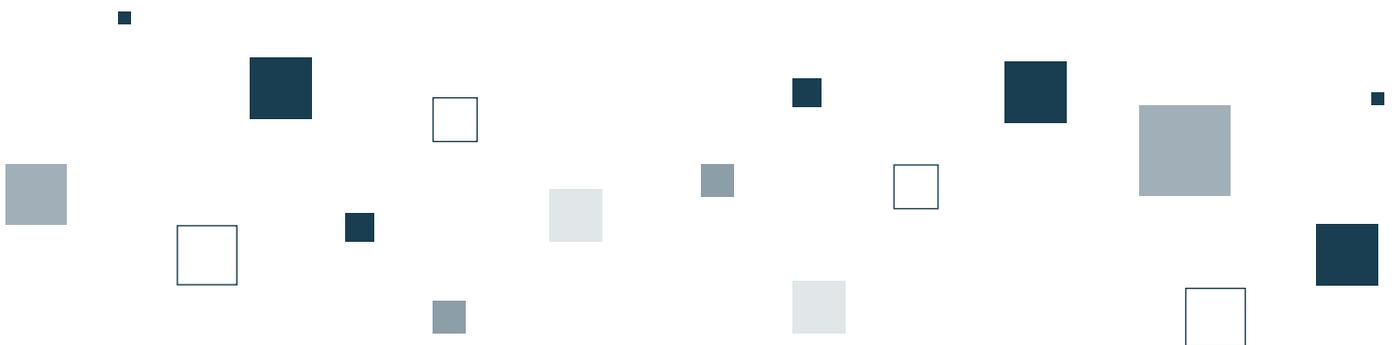
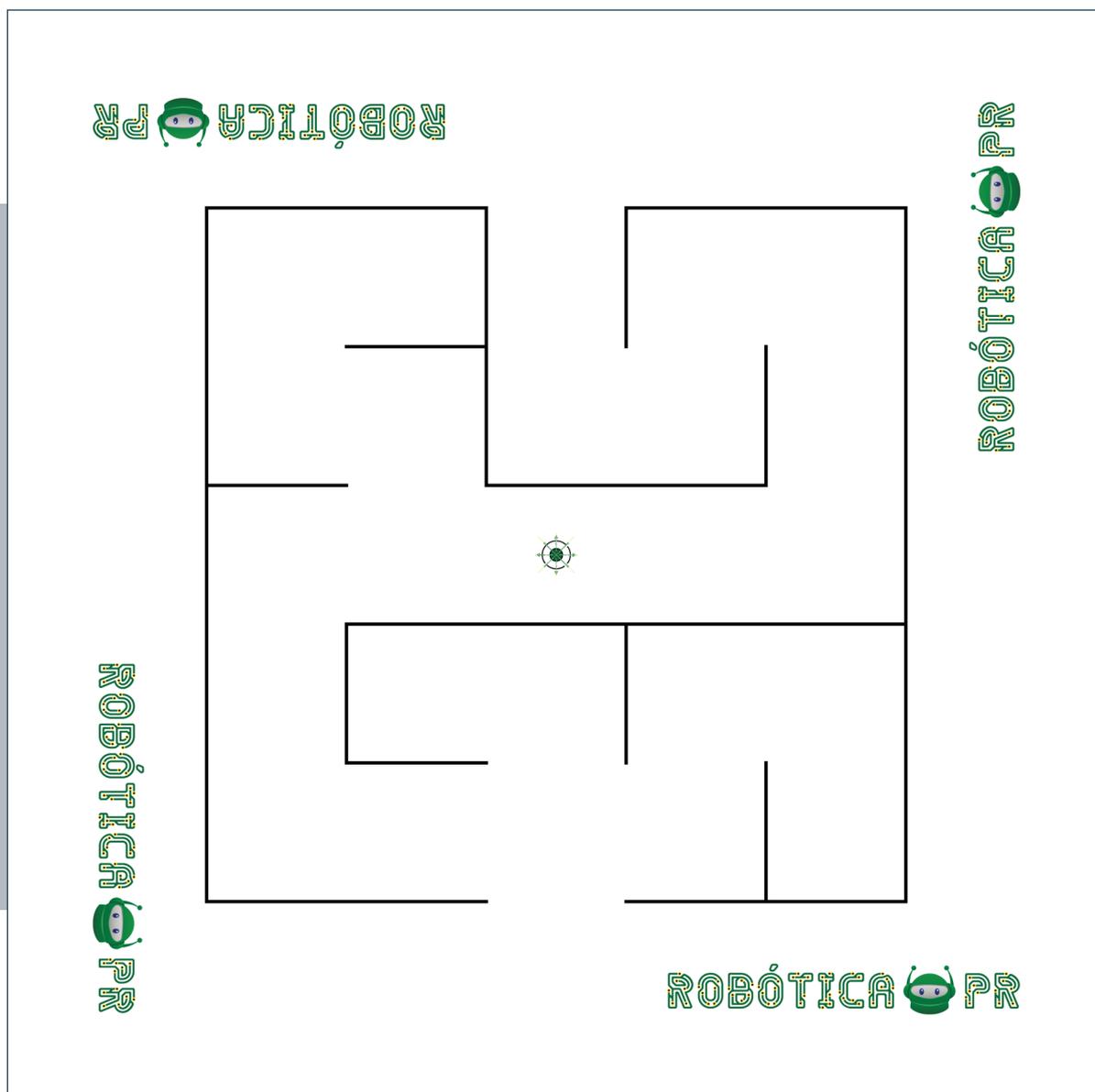


Figura 1 – [Modelo de labirinto](#)

Fonte: Seed/DTI/CTE.

O [modelo de labirinto, disponível em PDF para impressão](#), possui 2 m x 2 m e é uma inspiração – você e seus colegas podem também criar seus próprios labirintos para disputas incríveis!

O labirinto pode ser construído com materiais simples e de baixo custo, como papelão, madeira, cartolina ou outros materiais reciclados. A escolha do material dependerá da disponibilidade e do nível de dificuldade desejado para o desafio – por exemplo, fazer com que o PalmaBot percorra o labirinto no menor tempo possível, utilizando apenas os comandos de palma para controlar seus movimentos.

O **design** de cada labirinto pode apresentar um percurso com diferentes níveis de dificuldade. Isso pode incluir:

- Trechos curvos que exigem precisão nos movimentos do robô.
- Passagens com largura limitada, desafiando a capacidade do robô de se manter no caminho correto.
- Espaços maiores que permitem ao robô executar movimentos mais amplos e rápidos.

O **nível de dificuldade** do labirinto pode ser ajustado de acordo com o nível de habilidade dos estudantes: para iniciantes, um labirinto mais simples, com menos curvas e corredores largos, pode ser mais adequado; para estudantes mais experientes, um labirinto complexo, com múltiplos caminhos e obstáculos, pode ser um desafio maior.

## Inspire-se!

[Maze Generator](#) é uma ferramenta online gratuita que permite criar labirintos personalizados de maneira rápida e fácil. Conforme a definição de parâmetros, a ferramenta gera labirintos automaticamente com diferentes tamanhos, complexidades e estilos.

Os labirintos gerados pelo [Maze Generator](#) são chamados de labirintos “perfeitos” por possuírem exatamente uma solução e sem seções isoladas. O labirinto gerado é exibido na tela para inspirar a criação do seu labirinto para o PalmaBot e você também pode redefinir seus parâmetros e baixar a imagem como um arquivo PDF, SVG ou PNG, conforme instruções do site para a geração.

Essa ferramenta pode ser útil para o projeto, pois permite criar labirintos personalizados para testar o robô de maneira prática. Você e seus colegas podem desenhar labirintos diferentes e comparar o desempenho do robô em cada um deles.



Na construção do labirinto, considere também as **dimensões**, que devem ser adequadas ao tamanho do PalmaBot, permitindo que ele se mova livremente sem ficar preso. É importante considerar a largura dos corredores e a altura das paredes para evitar dificuldades excessivas, lembrando que a **montagem** do labirinto deve ser feita de forma que as paredes fiquem firmes e estáveis, evitando que o robô as derrube acidentalmente.

Defina também as **regras**! A principal e mais importante: os robôs devem ser controlados por palmas, sem intervenção manual durante o percurso. Considere, com seus colegas, a possibilidade de definir um tempo limite para conclusão do labirinto ou mesmo penalidades que podem ser aplicadas por erros de percurso ou por tocar as paredes do labirinto.

Antes do desafio, é **importante testar** o labirinto com PalmaBot para garantir que

o percurso seja viável e que o robô consiga completar o trajeto. Isso permite identificar possíveis problemas e fazer ajustes no design do labirinto e nas próprias regras, se necessário!

Para **avaliar** o desempenho dos robôs, você e seus colegas podem considerar o tempo que cada equipe leva para completar o labirinto. Vocês também podem focar no número de palmas utilizadas para completar o labirinto e na precisão dos movimentos, avaliando a capacidade do robô de seguir o percurso sem tocar nas paredes ou sair do caminho definido, e até a criatividade nas soluções implementadas.

Para deixar a avaliação mais emocionante, **penalidades** podem ser aplicadas por cada erro de percurso ou toque nas paredes e, analisando os erros detectados, estimule a criatividade e eficiência das modificações no código para melhorar o desempenho do robô no labirinto.

### MODELOS DE COMPETIÇÃO

Competição por tempo	A equipe que completar o labirinto no menor tempo é a vencedora.
Competição por pontos	Atribuir pontos para cada etapa do percurso completada com sucesso e subtrair pontos por erros ou penalidades. A equipe com a maior pontuação ao final do tempo limite é a vencedora.
Competição mista	Combinar tempo e pontos, onde o tempo é o critério principal, mas pontos extras são concedidos por precisão e inovação.

É importante definir regras claras e transparentes antes do início da competição para garantir que todas as equipes compreendam os critérios de avaliação e as expectativas. Além disso, o desafio deve promover a criatividade, o trabalho em equipe e a resolução de problemas, incentivando os alunos a aplicarem os conhecimentos adquiridos nas **Aulas 26 e 27 – PalmaBot [Partes I e II]**.

Encare essa jornada e aproveite o momento com seus colegas e seu PalmaBot!

Figura 2 - Representação do PalmaBot



Fonte: Seed/DTI/CTE.

**Desafio:**

Que tal criar diferentes níveis de dificuldade no labirinto para desafiar as equipes e permitir que mostrem suas habilidades em diferentes cenários?

Experimentando mais o robô controlado por palmas percorrendo o labirinto, que tal modificar o código para adicionar novas funcionalidades, como diferentes sequências de palmas para diferentes movimentos? Como vimos, o controle por palmas é um recurso fascinante que pode ser aplicado em variados projetos, tanto criativos quanto práticos, e mostra a diversidade de aplicações da Robótica.

Que tal também propor às equipes a obtenção de bônus por completar o percurso utilizando funcionalidades extras implementadas no código, como desvio de obstáculos com sensores adicionais para auxiliar na navegação do labirinto e ampliar o desempenho do robô durante o percurso?

Que tal colocar também seu PalmaBot para desviar de obstáculos?

**E se...**

- Os movimentos do robô estiverem descontrolados ou não se movimentarem como o esperado? Verifique os motores, a regulação do sensor de som e a alimentação do protótipo. Lembre-se de garantir que as baterias do PalmaBot estejam com carga suficiente e tensão entre 6V e 12V para o robô passar pelos testes e percorrer o labirinto!
- O desempenho do robô não for satisfatório? Desenvolva estratégias para otimizar o percurso do robô no labirinto, considerando a precisão dos comandos de palma e a velocidade de resposta do robô.

## 2. Feedback e finalização

O projeto PalmaBot oferece uma excelente oportunidade para explorarmos os conceitos de robótica, programação e interação com o ambiente. A combinação de programação detalhada, montagem prática e testes, como vimos nas **Aulas 26 e 27**, permite um aprendizado prático e a reflexão sobre as aplicações e o potencial da robótica na sociedade. As questões levantadas sobre controle autônomo versus controle remoto estimulam o pensamento crítico e a busca por soluções inovadoras, promovendo também momentos lúdicos, como o desafio de percorrer um labirinto, o qual promove a criatividade, o trabalho em equipe e a resolução de problemas. Como foi a experiência, as dificuldades e o aprendizado durante a jornada?

Para finalizar, as equipes podem incluir uma apresentação sobre as estratégias utilizadas, os desafios encontrados e as soluções implementadas para melhorar o desempenho do robô, apresentando também um relatório técnico detalhando o processo de desenvolvimento, as escolhas de design e as otimizações realizadas.

A documentação de projetos é essencial tanto para o registro quanto para o compartilhamento e aprimoramento futuro e você e seus colegas podem, cada vez mais, promover outros desafios e visualizar novas aplicações aos projetos da Robótica.

Parabéns por seguirem na jornada!



### REFERÊNCIAS

ARDUINO. **Documentação de Referência da Linguagem Arduino**. Disponível em: <https://www.arduino.cc/reference/pt/>. Acesso em: 27 mai. 2024.

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

**EQUIPE ROBÓTICA PARANÁ**

- Adilson Carlos Batista
- Ailton Lopes
- Andrea da Silva Castagini Padilha
- Cleiton Rosa
- Darice Alessandra Deckmann Zanardini
- Edna do Rocio Becker
- Kellen Pricila dos Santos Cochinski
- Marcelo Gasparin
- Michele Serpe Fernandes
- Michelle dos Santos
- Roberto Carlos Rodrigues
- Sandra Aguera Alcova Silva
- Viviane Dziubate Pittner

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 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](https://creativecommons.org/licenses/by-nc-sa/4.0/)

