



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA CENTRAL
CURSO BACHARELADO EM ENGENHARIA MECÂNICA

PAULO ROBERTO ARAÚJO LEAL

OMNIBOT: ROBÔ OMNIDIRECIONAL E SOLUÇÃO IOT PARA CONTROLE E
MONITORAMENTO REMOTO

TERESINA

2025

PAULO ROBERTO ARAÚJO LEAL

OMNIBOT: ROBÔ OMNIDIRECIONAL E SOLUÇÃO IOT PARA CONTROLE E
MONITORAMENTO REMOTO

Trabalho de Conclusão de Curso (monografia)
apresentado como exigência parcial para
obtenção do diploma do Curso de Bacharelado
em Engenharia Mecânica do Instituto Federal
de Educação, Ciência e Tecnologia do Piauí,
Campus Teresina Central.

Orientador: Prof. Me. Francisco Marcelino
Almeida de Araújo.

TERESINA

2025

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Leal, Paulo Roberto Araújo

L435o Omnibot : robô omnidirecional e solução IOT para controle e monitoramento remoto / Paulo Roberto Araújo Leal. - 2025.
62 f.: il. color.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia Mecânica)
- Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus
Teresina Central, 2025.

Orientador : Prof Me. Francisco Marcelino Almeida de Araújo.

1. Omnidirecional. 2. ROS. 3. Micro-ROS. 4. IOT. I.Título.

CDD - 620

Elaborado por Tanize Maria Sales CRB 3/1024

PAULO ROBERTO ARAÚJO LEAL

OMNIBOT: ROBÔ OMNIDIRECIONAL E SOLUÇÃO IOT PARA CONTROLE E
MONITORAMENTO REMOTO

Trabalho de Conclusão de Curso (monografia)
apresentado como exigência parcial para
obtenção do diploma do Curso de Bacharelado
em Engenharia Mecânica do Instituto Federal
de Educação, Ciência e Tecnologia do Piauí,
Campus Teresina Central.

Aprovado (a) em: 11 / 12 / 2025 .

BANCA EXAMINADORA

Prof. Me. Francisco Marcelino Almeida de Araújo (Orientador)
Instituto Federal de Educação, Ciência e Tecnologia do Piauí (IFPI)

Prof. Me. Edivaldo Feitosa Pereira
Instituto Federal de Educação, Ciência e Tecnologia do Piauí (IFPI)

Prof. Dra. Rosilda Maria Alves
Instituto Federal de Educação, Ciência e Tecnologia do Piauí (IFPI) e Secretaria
Municipal de Educação de Teresina

Prof. Me. Márcio Antônio Dias Pereira
Instituto Federal de Educação, Ciência e Tecnologia do Piauí (IFPI)

A Deus.

Aos meus pais.

Aos mestres.

A todos que acreditaram em mim.

AGRADECIMENTOS

Agradeço à Deus, por tudo.

A minha família, pelo incentivo, pelo apoio e por sempre me mostrarem que a educação é o melhor caminho.

Aos amigos e colaboradores de pesquisa do LABIRAS, por terem me acolhido desde o início desta jornada, bem como por todas as oportunidades e ensinamentos que não seriam alcançados sem o apoio de vocês. Em especial, ao professor Francisco Marcelino e ao colega Matheus Dantas.

Aos professores, que muito contribuíram com minha formação acadêmica, agradeço por todas as lições e ensinamentos, as conversas, as orientações e a oportunidade de poder aprender com vocês. Foi uma honra tê-los como mestres.

"O começo da sabedoria é a declaração 'eu não sei'. A pessoa que não pode fazer essa declaração é aquela que nunca aprenderá nada."
(Thrall, o Guardião da Terra, 2006)

RESUMO

A robótica vem se estabelecendo como um campo essencial, impulsionada pelos progressos tecnológicos e pela demanda crescente por automação em vários setores. Nesse contexto, a robótica móvel é um dos segmentos que mais se destaca, não apenas pelo crescimento da robótica na indústria, mas também pelos esforços significativos na robótica educacional, que adota uma abordagem mais acadêmica e voltada para a pesquisa. Sob esse viés, este trabalho apresenta o desenvolvimento de uma plataforma robótica móvel omnidirecional denominada OmniBot. O objetivo é disponibilizar uma ferramenta didática que possa ser utilizada em sala de aula e em laboratório para o ensino e pesquisa de robótica no curso de Bacharelado em Engenharia Mecânica do Instituto Federal do Piauí (IFPI). O projeto foi concebido como uma solução modular, de baixo custo e fácil replicação, integrando conceitos de robótica móvel, Internet das Coisas (IoT) e controle distribuído. O robô foi construído utilizando motores JGA25-370 com *encoders*, rodas *mecanum*, microcontrolador ESP32 e interface de controle remoto BitDogLab (Raspberry Pi Pico W), que se comunica com o Robot Operating System 2 (ROS 2) por meio do micro-ROS. A estrutura mecânica foi modelada em Autodesk Fusion 360 e fabricada por impressão 3D, permitindo o encaixe modular de sensores e controladores para diferentes cenários de aplicação. Com isso, o OmniBot foi testado em laboratório e apresentou êxito conforme esperado para sua movimentação em todas as direções, além de comunicação estável com o ROS 2 e controle remoto responsivo. Por fim, foi elaborada uma documentação técnica contendo todas as informações necessárias para montagem, operação e modificação do robô, reforçando seu caráter educacional e aberto.

Palavras-chave: robótica; omnidirecional; ros; micro-ros; iot; engenharia.

ABSTRACT

Robotics has been establishing itself as a vital field, driven by technological advances and the growing demand for automation in various sectors. In this context, mobile robotics is one of the most prominent segments, not only due to the growth of robotics in industry but also due to significant efforts in educational robotics, which adopts a more academic and research-oriented approach. From this perspective, this paper presents the development of an omnidirectional mobile robotics platform called OmniBot. The goal is to provide a teaching tool that can be used in the classroom and laboratory for teaching and researching robotics in the Bachelor's degree program in Mechanical Engineering at the Federal Institute of Piauí (IFPI). The project was conceived as a modular, low-cost, and easily replicable solution, integrating concepts from mobile robotics, the Internet of Things (IoT), and distributed control. The robot was built using JGA25-370 motors with encoders, mecanum wheels, an ESP32 microcontroller, and a BitDogLab remote control interface (Raspberry Pi Pico W), which communicates with the Robot Operating System 2 (ROS 2) via micro-ROS. The mechanical structure was modeled in Autodesk Fusion 360 and manufactured using 3D printing, allowing for modular attachment of sensors and controllers for different application scenarios. The OmniBot was then tested in the laboratory and demonstrated success, as expected, with its movement in all directions, stable communication with ROS 2, and responsive remote control. Finally, technical documentation was prepared containing all the necessary information for assembling, operating, and modifying the robot, reinforcing its educational and open-source nature.

Keywords: robotics; omnidirectional; ros; micro-ros; iot; engineering.

LISTA DE FIGURAS

Figura 1 – "Tortoise" robô móvel	18
Figura 2 – Os quatro tipos de rodas.	20
Figura 3 – Veículos Omnidirecionais.	21
Figura 4 – Roda rodízio (esquerda) e roda direcional (direita).	21
Figura 5 – Roda universal (esquerda) e roda mecanum (direita).	22
Figura 6 – Rodas <i>Mecanum</i>	22
Figura 7 – Mecânica de um robô omnidirecional.	23
Figura 8 – Veículo Omnidirecional KUKA OmniMove.	23
Figura 9 – Exemplos de movimentação de um robô com rodas mecanum.	24
Figura 10 – Eixos do veículo e suas dimensões.	24
Figura 11 – Velocidade em relação ao eixo fixo.	26
Figura 12 – Comunicação entre <i>nodes</i>	30
Figura 13 – Arquitetura ROS 1/ROS 2.	30
Figura 14 – Robôs quadrúpedes para exploração e mapeamento em minas subterrâneas.	31
Figura 15 – Robô VIPER da NASA.	31
Figura 16 – Arquitetura do micro-ROS.	32
Figura 17 – Layout do PCB BitDogLab: (a) frente e (b) parte traseira.	33
Figura 18 – Análise comparativa dos recursos do BitDogLab, Micro:bit e Arduino para uso educacional.	34
Figura 19 – Vista frontal do BitDogLab montado com todos os seus componentes.	35
Figura 20 – Visão ampliada dos terminais de conexão do BitDogLab que podem ser conectados com cliques jacaré.	35
Figura 21 – Microcontrolador ESP-32.	35
Figura 22 – Roomba montado com ESP 32, bateria e a base para adição de sensores variados.	37
Figura 23 – DJI ® Ryze Tello Combo Boost.	38
Figura 24 – Plataforma de Pesquisa e Ensino OMNINO.	39
Figura 25 – Finalistas do AWS DeepRacer 2024.	39
Figura 26 – Finalistas do AWS DeepRacer 2025.	40
Figura 27 – Robô AWS DeepRacer do IFPI.	40
Figura 28 – Impressora 3D PRO-GTMAX Core H5.	41

Figura 29 – Motores JGA25-370 com <i>encoders</i> acoplados.	42
Figura 30 – Plataformas robóticas móveis terrestres do LABIRAS.	43
Figura 31 – Diagrama com a proposta de arquitetura de conexão entre os dispositivos. . .	45
Figura 32 – Estrutura de projeto ROS 2.	46
Figura 33 – Dimensões da placa do Create 3.	47
Figura 34 – Modelagem 3D do OmniBot no Autodesk Fusion 360 a) Vista interna b) Vista externa.	48
Figura 35 – Montagem física do robô OmniBot.	49
Figura 36 – Layout da placa PCB de controle do OmniBot a) Vista superior b) Vista inferior. .	49
Figura 37 – Diagrama de esquemático conexões eletrônicas do OmniBot.	50
Figura 38 – Montagem interna do OmniBot, mostrando o arranjo dos componentes. . . .	50
Figura 39 – Modelagem do case desenvolvido para a BitDogLab.	51
Figura 40 – Case da BitDogLab impresso em 3D a) Vista frontal b) Vista posterior. . . .	51
Figura 41 – Configuração dos parâmetros do robô.	52
Figura 42 – rqt mostrando o <i>node</i> e <i>topics</i> gerados pelo micro-ROS.	52
Figura 43 – rqt mostrando toda a arquitetura de conexão dos <i>nodes</i> e <i>topics</i>	53
Figura 44 – Informações do display OLED da BitDogLab do funcionamento em a) Modo diferencial b) Modo holonômico.	54
Figura 45 – rqt mostrando toda a arquitetura de conexão dos <i>nodes</i> e <i>topics</i> para operação simultânea de 3 dispositivos robóticos.	55
Figura 46 – Informações do display OLED da BitDogLab do funcionamento durante a operação simultânea de 3 dispositivos robóticos em a) Modo diferencial b) Modo holonômico.	55

LISTA DE TABELAS

Tabela 1 – Especificações do ESP8266 e ESP32.	36
---	----

LISTA DE SIGLAS

API	<i>Application Programming Interfaces</i>
BLE	<i>Bluetooth Low Energy</i>
DDS	<i>Data Distribution Service</i>
GPIO	<i>General Purpose Input/Output</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IFPI	Instituto Federal de Educação, Ciência e Tecnologia do Piauí
IoT	<i>Internet of Things</i>
LABIRAS	Laboratório de Inovação e Pesquisa Aplicada
MC	Matriz Cinemática
MR	Matriz de Rotação
NASA	<i>National Aeronautics and Space Administration</i>
Open-CV	<i>Open Source Computer Vision</i>
OSRF	<i>Open Source Robotics Foundation</i>
PCL	<i>Point Cloud Library</i>
POSIX	<i>Portable Operating System Interface</i>
PCB	<i>Printed Circuit Board</i>
PPC	Projeto Pedagógico de Curso
ROS	<i>Robot Operating System</i>
RTOS	<i>Real Time Operating System</i>
SOC	<i>System On Chip</i>
SRAM	<i>Static Random Access Memory</i>
STEAM	<i>Science, Technology, Engineering, Arts e Mathematics</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagram Protocol</i>
VIPER	<i>Volatiles Investigation Polar Exploration Rover</i>
XRCE	<i>Extremely Resource Constrained Environments</i>

LISTA DE SÍMBOLOS

- ° Graus
- δ Letra grega minúscula Delta
- θ Letra grega minúscula Teta
- ω Letra grega minúscula Ômega
- ® Marca registrada

SUMÁRIO

1	INTRODUÇÃO	15
2	OBJETIVOS	17
2.1	OBJETIVO GERAL	17
2.2	OBJETIVOS ESPECÍFICOS	17
3	FUNDAMENTAÇÃO TEÓRICA	18
3.1	ROBÔS MÓVEIS	18
3.1.1	Locomoção por rodas	19
3.1.2	Dispositivos omnidirecionais	20
3.1.2.1	Rodas de um veículo omnidirecional	21
3.1.2.2	Roda Mecanum	22
3.1.2.3	Cinemática de um robô móvel de acionamento omnidirecional com <i>mecanum wheels</i>	24
3.2	ROS	26
3.2.1	Conceitos básicos	28
3.2.2	Casos de uso	30
3.2.3	Micro-ROS	32
3.3	BITDOGLAB	33
3.4	ESP-32	34
4	TRABALHOS RELACIONADOS	37
5	METODOLOGIA	41
5.1	PESQUISA BIBLIOGRÁFICA	41
5.2	FERRAMENTAS E MATERIAIS	41
5.3	SOFTWARE	43
5.4	HARDWARE	44
5.5	PADRONIZAÇÃO	45
5.6	DOCUMENTAÇÃO	47
6	RESULTADOS	48
6.1	ESTRUTURA FÍSICA E MONTAGEM	48
6.2	OPERAÇÃO DO ROBÔ	51
6.3	OPERAÇÃO DE OUTRAS PLATAFORMAS ROBÓTICAS	54
7	CONCLUSÃO	56

8	TRABALHOS FUTUROS	57
	REFERÊNCIAS	58

1 INTRODUÇÃO

Para a civilização ocidental o conceito de evolução humana está diretamente associado ao grau de desenvolvimento tecnológico adquirido ao longo do tempo [...]. Portanto, a motivação de se criar máquinas que possam substituir o homem na realização de tarefas, é uma característica da própria cultura ocidental” (ROMANO; DUTRA, 2002).

Com o advento da Quarta Revolução Industrial, a otimização e a flexibilidade tornaram-se variáveis importantes em sistemas de produção. A virtualização, a descentralização e a construção de redes, possibilitadas pelas tecnologias emergentes, têm transformado o ambiente de produção tradicional (FRAGAPANE *et al.*, 2022).

Os avanços tecnológicos exercem impacto direto sobre a indústria, que busca reduzir custos por meio da adoção de diferentes modelos de produção. Nesse contexto, os robôs industriais, caracterizados como máquinas reprogramáveis capazes de executar diversos movimentos e adaptar-se às necessidades das tarefas que executam, são amplamente empregados em processos de automação.

O termo robô foi originalmente utilizado em 1921 pelo dramaturgo checo Karen Capek, na sua peça de teatro "Os Robôs de Rossum" como referência a um autômato que acaba rebelando-se contra o ser humano. A palavra robô é derivada de "robota" de origem eslava que significa "trabalho forçado". O termo robótica foi criado por Asimov para designar a ciência que se dedica ao estudo destas máquinas (SCIAVICCO; SICILIANO, 1995).

Robôs podem ser definidos ainda como mecanismos atuadores programados com um grau de autonomia para performar locomoção, manipulação ou posicionamento sendo dotados de um sistema de controle. Alguns exemplos de estruturas mecânicas robóticas são manipuladores, plataformas móveis e robôs vestíveis (STANDARTIZATION, 2012).

A robótica tem se consolidado como um campo fundamental, impulsionada pelos avanços tecnológicos e pela crescente demanda por automação em diversos setores, em consoante com Oliveira, Santos e Ferreira (2024), tais como automação industrial, monitoramento ambiental e veículos autônomos. Em 2020, de acordo com Robotics (2020), foi registrado o recorde de 2.7 milhões de robôs industriais operando em fábricas ao redor do mundo, sendo que, na América do Sul, o Brasil é o país líder com estoque operacional com quase 15.300 unidades. Diante disso, a robótica móvel é um dos segmentos da robótica que mais está ganhando notoriedade, pois além do crescimento do espaço da robótica na indústria, muitos esforços estão sendo colocados na robótica educacional, com uma abordagem mais acadêmica e de pesquisa.

Conforme o Robot Institute of America, um robô móvel é um manipulador multifuncional e reprogramável projetado para cumprir diferentes tarefas por intermédio de movimentos integrados (CONSIDINE; CONSIDINE, 2012). Diante disso, Leao *et al.* (2021) expõe que o uso de robôs móveis integra ambientes industriais e centros de pesquisa, possibilitando maior segurança para os profissionais envolvidos, bem como diminuir o risco de acidentes e possíveis ocorrências de atrasos. Além do mais, novas tecnologias, dispositivos e programação aplicada, elevam as possibilidades de robôs realizarem uma gama de tarefas com elevada precisão, conforme apresentado por Martins, Neto e Farias (2021).

Ainda assim, o uso de robôs convencionais apresenta limitações, pois muitos dependem de espaços amplos para operar com eficiência, geralmente possuindo mobilidade restrita e necessidade de trajetórias pré definidas. Isto posto, esse fatores acabam tornando-os menos eficientes e inadequados para ambientes dinâmicos, onde mudanças frequentes na disposição do espaço podem ocorrer, ou em cenários compactos e congestionados, em consoante com Parada-Salado *et al.* (2018) e Franco *et al.* (2017). Diante disso, uma alternativa eficiente para mitigar esses desafios se torna importante.

Para mais, à medida que a escala e o escopo dos sistemas robóticos atuais crescem, projetar e desenvolver o software que os comanda está se tornando cada vez mais difícil, conforme apresentado por Quigley *et al.* (2009). Na robótica, de acordo com Albonico *et al.* (2023) e Macenski *et al.* (2022), majoritariamente, por razão da alta complexidade envolvida, é comum dispor de frameworks e suítes de software, entre os quais o *Robot Operating System* (ROS) se tornou um padrão.

Sendo assim, esse trabalho propõe o desenvolvimento de um robô omnidirecional com rodas mecanum wheels e uma solução de *Internet of Things* (IoT) para controle e monitoramento desse dispositivo robótico elaborados no ambiente ROS.

Todavia, a utilização dessa plataforma robótica não fica limitada somente aos tópicos apresentados, já que no Projeto Pedagógico de Curso (PPC) do curso de Engenharia Mecânica do IFPI é previsto a disciplina de robótica como uma das 23 disciplinas que podem ser abordadas no grupo de optativas de projeto. Nele, são previstos por semana 3 horas de aula teórica e 1 hora de aula prática (LOPES *et al.*, 2018). Desse modo, tanto o material desenvolvido como a plataforma podem ser utilizados pelos alunos durante as aulas práticas da graduação.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Desenvolver uma plataforma robótica móvel terrestre omnidirecional com rodas *mecanum* como ferramenta para pesquisa e ensino de robótica industrial para os alunos do curso de bacharelado em engenharia mecânica do IFPI, e uma solução IoT para controle e monitoramento remoto desse robô.

2.2 OBJETIVOS ESPECÍFICOS

- Projetar e elaborar o modelo físico do robô;
- Desenvolver um *hardware* IoT para controle e monitoramento remoto desse robô através do ROS 2;
- Descrever ROS 2 como *framework* de robótica para indústria;
- Operar o robô real utilizando ROS 2;
- Receber dados dos sensores embarcados do robô;
- Adaptar o hardware para receber novos sensores;
- Padronizar a forma que ele recebe e envia as mensagens;
- Documentar a utilização do robô e o funcionamento do mesmo.

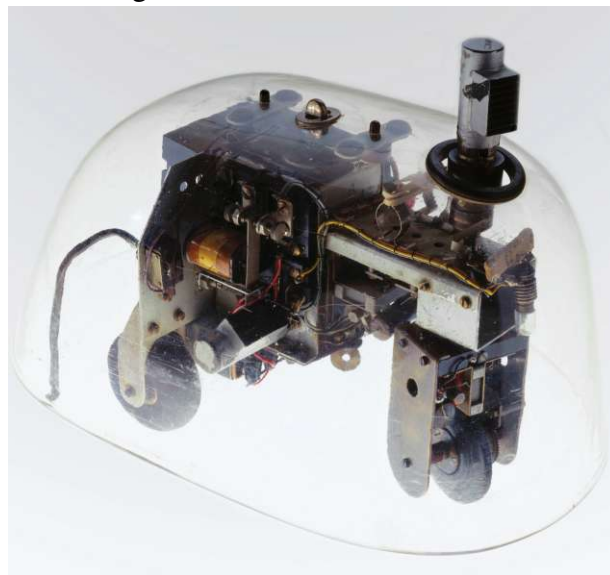
3 FUNDAMENTAÇÃO TEÓRICA

Para fundamentar as decisões de projeto adotadas no desenvolvimento da solução apresentada neste trabalho, torna-se imprescindível compreender previamente alguns conceitos teóricos essenciais referentes a robôs móveis e ao ROS.

3.1 ROBÔS MÓVEIS

Segundo Niloy *et al.* (2021), robôs capazes de deslocar-se de maneira autônoma e de realizar escolhas inteligentes a partir da percepção do ambiente e dos objetos ao seu redor são classificados como robôs móveis autônomos. Um dos primeiros exemplares desse tipo de sistema foi o robô *Tortoise*, desenvolvido por W. Grey Walter, representado na Figura 1.

Figura 1 – "Tortoise" robô móvel



Fonte: Marsh (2020).

O robô *Tortoise* foi concebido com dois tipos de sensores distintos. O primeiro consiste em uma célula fotoelétrica, dispositivo capaz de gerar um sinal elétrico em resposta à exposição à radiação eletromagnética. Esta célula era rotativa e atuava como o “olho” do robô, realizando varreduras do ambiente até localizar uma fonte de luz. Ao detectar a fonte, o robô ajustava seu movimento, aproximando-se caso a intensidade fosse moderada ou afastando-se caso a luminosidade fosse excessiva. O segundo sensor é um interruptor de contato externo, sensível ao toque, que provoca o recuo do robô ao encontrar obstáculos. Além disso, o *Tortoise* possuía um mecanismo de retorno automático à estação de carregamento quando o nível de bateria se

encontrava baixo.

No estudo intitulado "*Critical design and control issues of indoor autonomous mobile robots: A review*", Niloy *et al.* (2021) ressaltam que robôs móveis devem ser capazes de operar em qualquer tipo de ambiente, incluindo terrenos acidentados, perigosos, irregulares, com obstáculos ou áreas restritas. A locomoção constitui um elemento essencial para o deslocamento do robô dentro de seu espaço de trabalho, sendo influenciada não apenas pelas características do ambiente ou do trajeto a ser percorrido, mas também por fatores críticos como controlabilidade, manobrabilidade, eficiência e estabilidade em diferentes tipos de terreno, conforme explicitado por Rubio, Valero e Llopis-Albert (2019). Robôs móveis de interior, em consoante com Dubey *et al.* (2015), são projetados para se deslocar a fim de cumprir tarefas específicas, podendo ser equipados com pernas, rodas ou até mesmo asas, sendo que desses mecanismos, os sistemas baseados em rodas demonstram maior eficiência, conforme será detalhado na seção subsequente. Neste trabalho, a atenção concentra-se em um robô móvel de interior com propulsão baseada em rodas.

3.1.1 Locomoção por rodas

Em consoante com Sa (2016), a locomoção por rodas constitui um dos mecanismos mais empregados em robôs móveis, sendo reconhecida por sua simplicidade de implementação, uma vez que é amplamente utilizada em diversos tipos de veículos. Esse método apresenta vantagens significativas, como estabilidade, facilidade de controle e design mecânico relativamente simples, o que torna sua aplicação eficiente e de fácil integração em sistemas robóticos.

Conforme Sa (2016), a escolha do tipo de rodas para um robô móvel está intimamente ligada à disposição e à geometria das rodas na estrutura do robô, sendo crucial considerar esses aspectos durante o planejamento do sistema de locomoção. As decisões relacionadas a esse projeto influenciam diretamente três características essenciais do robô: estabilidade, capacidade de manobra e capacidade de controle.

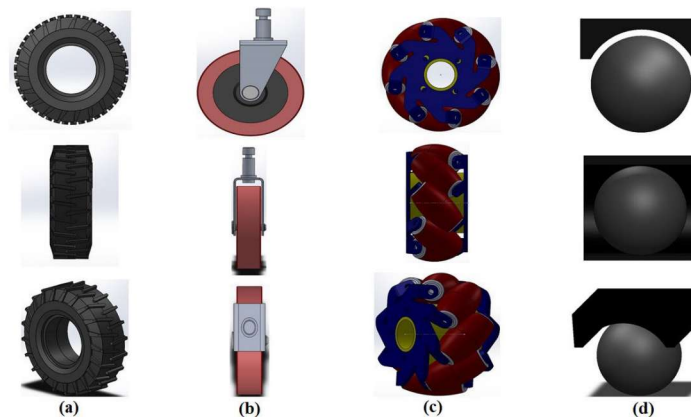
Nesse contexto, podem ser identificados 4 tipos distintos de locomoção baseada em rodas, a saber:

- a) **Roda padrão:** apresenta dois graus de liberdade, com rotação em torno do eixo e do ponto de contato com o solo.
- b) **Roda rodízio:** possui dois graus de liberdade, com o movimento de giro ocorrendo em torno da junta de direção deslocada.

- c) **Roda omnidirecional:** dispõe de três graus de liberdade, permitindo rotação em torno dos rolos, do eixo da roda e do ponto de contato com o solo.
- d) **Roda esférica:** caracteriza-se por uma maior área de contato em comparação à roda rodízio, que possui formato cilíndrico.

A Figura 2 ilustra as quatro categorias de rodas comumente empregadas em robôs móveis. Conforme o ambiente de operação, o arranjo de rodas pode combinar diferentes tipos ou utilizar múltiplas unidades de um mesmo tipo. Tanto o tipo de roda quanto sua geometria configuram características essenciais para o desempenho do robô, influenciando diretamente sua estabilidade, manobrabilidade e controlabilidade.

Figura 2 – Os quatro tipos de rodas.



Fonte: Siegwart, Nourbakhsh e Scaramuzza (2011).

3.1.2 Dispositivos omnidirecionais

Com base no apresentado por Sa (2016), este tipo de robô possui máxima capacidade de manobra no plano, o que implica que ele é capaz de se deslocar em qualquer direção sem a necessidade de se redirecionar, possuindo uma boa manobrabilidade.

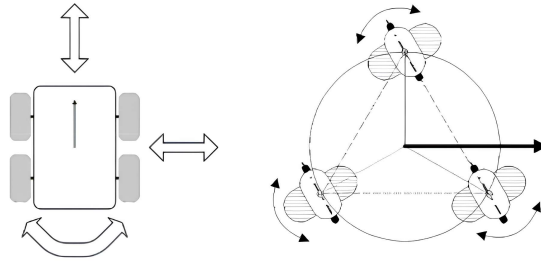
A manobrabilidade é a habilidade de manobrar em qualquer direção gradualmente, de acordo com Dubey *et al.* (2015), e é uma das principais preocupações no projeto de robôs.

Dessa forma, esses robôs possuem a capacidade de se mover com três graus de liberdade (3DOF), ou seja, seu posicionamento será determinado a partir de três valores, possuindo a capacidade de transladar em duas direções e rodar em relação ao seu centro de gravidade.

Portanto, um veículo omnidirecional, conforme exposto na Figura 3, tem seu posicionamento determinado por três dimensões: duas para indicar sua posição no plano e uma para

indicar sua rotação em torno de seu eixo vertical, que é ortogonal ao plano de deslocamento.

Figura 3 – Veículos Omnidirecionais.



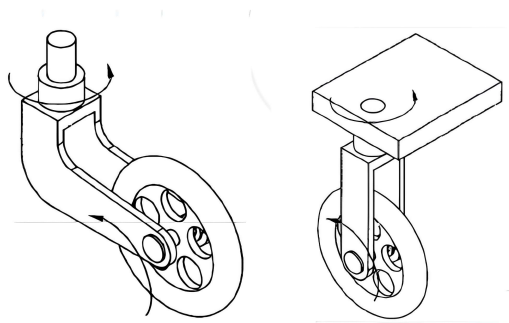
Fonte: Elaborado pelo autor (2025).

3.1.2.1 Rodas de um veículo omnidirecional

Para a construção de um veículo omnidirecional é fundamental considerar os tipos de rodas que estarão sendo utilizadas, existindo dois tipos que permitem obter este tipo de mobilidade com rodas sem restrições, as rodas de desenho convencional e as rodas de desenho especial. (SA, 2016).

As rodas convencionais com capacidade omnidirecional, ilustradas na Figura 4, são empregadas em robôs móveis e são categorizadas em duas categorias: as rodas direcionais e as rodas com rodízios.

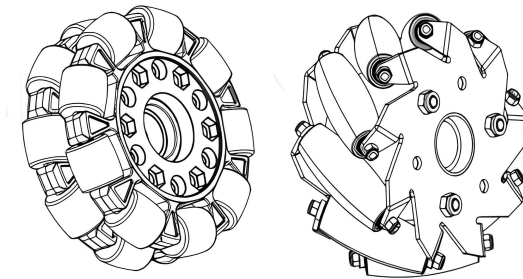
Figura 4 – Roda rodízio (esquerda) e roda direcional (direita).



Fonte: Sa (2016).

Por outro lado, as rodas com desenho especial, além de terem maior capacidade de carga, também têm maior tolerância a irregularidades do terreno. Essas rodas são classificadas como rodas especiais por possuírem tração em uma direção e movimento passivo na outra. Na Figura 5, elas são apresentadas e podem ser categorizadas em duas categorias: as universais, também chamadas de suecas, e as mecanum.

Figura 5 – Roda universal (esquerda) e roda mecanum (direita).



Fonte: Sa (2016).

3.1.2.2 Roda Mecanum

Desenvolvida por Bengt Ilon na década de 1970 (DICKERSON; LAPIN, 1991), as *Mecanum Wheels*, apresentadas na Figura 6, são um componente de destaque na construção de robôs omnidirecionais. É a partir dos princípios fundamentais da física, mais especificamente relacionados à distribuição de forças e ao momento angular, que decorre sua capacidade de movimento em diversos sentidos.

Figura 6 – Rodas *Mecanum*.

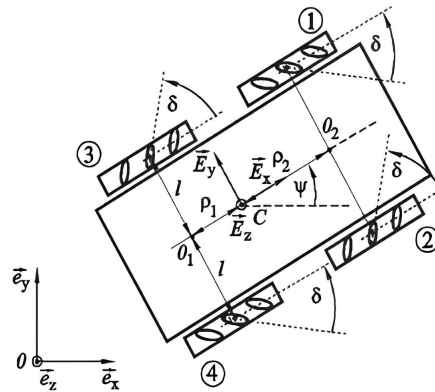


Fonte: Retirada da Internet: <https://www.robotshop.com/products/mecanum-wheels-4-pieces-compatible-with-lego-blocks> (2025).

A característica mais marcante dessas rodas está na disposição diagonal dos rollers (rolos) em torno da sua circunferência. Esses roletes, em vez de colocados perpendicularmente ao eixo são colocados a um ângulo de 45° do eixo da roda (δ), tendo seu eixo de rotação deslocado em relação ao eixo de rotação da roda, conforme exposto no esquema da Figura 7. Em virtude dessa

configuração singular, as rodas têm uma habilidade especial de gerar forças de propulsão em ângulos diversos, o que possibilita executar movimentos complexos e omnidirecionais (TĂTAR *et al.*, 2014).

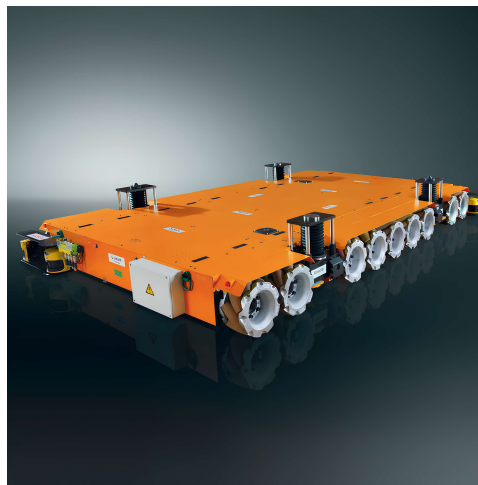
Figura 7 – Mecânica de um robô omnidirecional.



Fonte: Elaborado pelo autor com base em Zeidis e Zimmermann (2019).

Ela é adequada a espaços limitados devido à capacidade de manobra que proporciona ao veículo, por isso, é empregada em equipamentos de transporte em indústrias que exigem essa característica. Sa (2016) aponta que veículos como o KUKA OmniMove (Figura 8) e o Airtrax Cobrar já empregam essa tecnologia no setor industrial, atendendo de forma inovadora o mercado de transporte de cargas industriais e diminuindo os custos logísticos.

Figura 8 – Veículo Omnidirecional KUKA OmniMove.

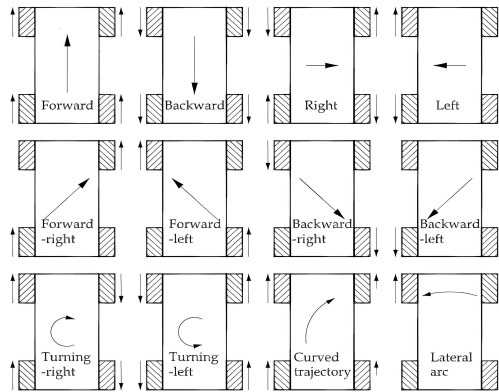


Fonte: Retirada da Internet: <https://www.directindustry-china.cn/prod/kuka-ag/product-17587-1714586.html> (2025).

Através do ajuste das velocidades de rotação individual de cada roda, é possível exercer controle tanto sobre a intensidade quanto sobre a direção das forças geradas por esses dispositi-

vos, sendo possível determinar os deslocamentos do robô através da movimentação das rodas, conforme apresentado na Figura 9.

Figura 9 – Exemplos de movimentação de um robô com rodas mecanum.

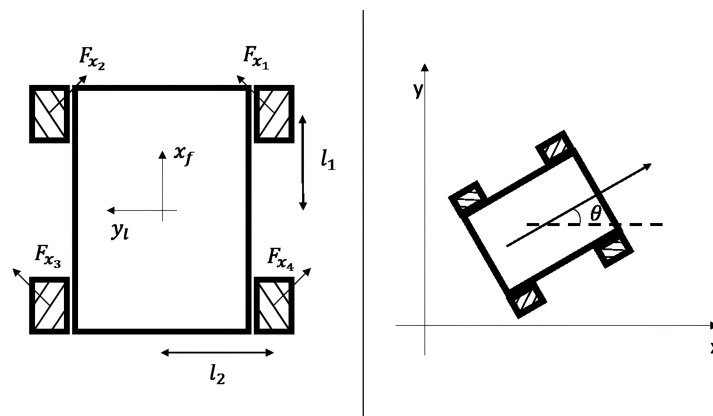


Fonte: Sa (2016).

3.1.2.3 Cinemática de um robô móvel de acionamento omnidirecional com *mecanum wheels*

Essa configuração tem seu posicionamento determinado a partir de três valores, às coordenadas X e Y para a posição e um ângulo θ para a rotação em relação ao eixo vertical, conforme apresentado na Figura 10. Com isso, conforme Sa (2016), o robô é capaz de mover-se livremente no plano, sem precisar realizar manobras ou se reorientar.

Figura 10 – Eixos do veículo e suas dimensões.



Fonte: Sa (2016).

Na robótica, a cinemática desempenha um papel fundamental para definir posição, orientação e velocidade dos robôs. A equação da cinemática direta permite prever o movimento do robô sabendo a velocidade angular de cada roda.

Dessa forma, considerando o veículo representado na Figura 10, os sistemas de coordenada apresentados são: o fixo (x, y), que é utilizado em relação ao espaço onde o veículo irá andar e o móvel (x_f, y_l), que é o sistema de coordenada do veículo.

Isto posto, a velocidade de rotação de cada roda é dada pela Matriz Cinemática (MC) abaixo:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & 1 & (l_1 + l_2) \\ 1 & -1 & -(l_1 + l_2) \\ 1 & 1 & -(l_1 + l_2) \\ 1 & -1 & (l_1 + l_2) \end{bmatrix} \begin{bmatrix} v_f \\ v_l \\ v_\theta \end{bmatrix} \quad (3.1)$$

Sendo:

ω_i - Velocidade angular da roda i

v_f - Velocidade frontal

v_l - Velocidade lateral

v_θ - Velocidade de rotação

Dado que a velocidade de rotação do motor está associada à velocidade de rotação das rodas através da relação de transmissão do motor N, a velocidade de rotação pode ser calculada através da seguinte equação:

$$\omega_{m_i} = \omega_i * N \quad (3.2)$$

Sendo:

ω_{m_i} - Velocidade angular do motor i

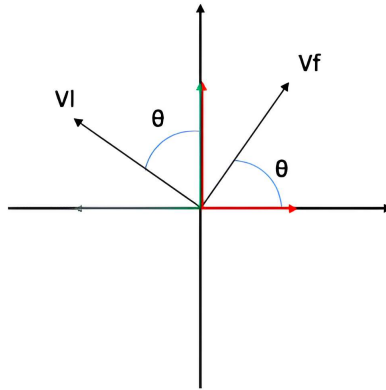
N - Relação de transmissão do motor

Primeiramente, para saber o posicionamento em um eixo fixo dado pelo esquema na Figura 10 é preciso calcular a velocidade do veículo nos eixos das coordenadas x e y.

Na Figura 11, é possível observar a projeção das velocidades frontal e lateral nos eixos x e y, resultando nas equações das velocidades v_x e v_y .

$$v_y = v_f * \sin \theta + v_l * \cos \theta \quad (3.3)$$

Figura 11 – Velocidade em relação ao eixo fixo.



Fonte: Sa (2016).

$$v_x = v_f * \cos \theta - v_l * \sin \theta \quad (3.4)$$

Sendo:

v_x - Velocidade na direção do eixo x

v_y - Velocidade na direção do eixo y

Passando para forma matricial, tem-se a matriz de rotação (MR).

$$\begin{bmatrix} v_y \\ v_x \end{bmatrix} = \begin{bmatrix} \sin \theta & \cos \theta \\ \cos \theta & -\sin \theta \end{bmatrix} \begin{bmatrix} v_f \\ v_l \end{bmatrix} \quad (3.5)$$

Integrando a velocidade é obtido o posicionamento em relação à x e y.

$$\int v_x = x + x_0 \quad (3.6)$$

$$\int v_y = y + y_0 \quad (3.7)$$

3.2 ROS

O *Robot Operating System* (ROS), ou Sistema Operacional Robótico, constitui um kit de desenvolvimento de software de código aberto voltado para aplicações em robótica. Ele fornece uma plataforma de software padronizada que auxilia desenvolvedores de diversos setores,

abrangendo desde atividades de pesquisa e prototipagem até etapas de implantação e produção (ROBOTICS, 2022). O ROS é um sistema de código aberto e meta-operacional, oferecendo serviços típicos de um sistema operacional, conforme destacado por Macenski *et al.* (2022), Quigley *et al.* (2009) e Koubaa *et al.* (2017):

- abstração de hardware;
- controle de dispositivo de baixo nível;
- implementação de funcionalidade comumente usada;
- troca de mensagens entre processos;
- gerenciamento de pacotes.

O ROS configura-se como um sistema operacional voltado para plataformas robóticas, permitindo a comunicação entre softwares desenvolvidos em diferentes linguagens de programação e a integração simplificada de diversos dispositivos de entrada e saída. Além disso, o ROS agrega esforços de múltiplos desenvolvedores na área de robótica, de acordo com o exposto por (QUIGLEY; GERKEY; SMART, 2015). Atualmente, existem mais de 7000 pacotes desenvolvidos para as diferentes distribuições do ROS (LALANCETTE; POUBEL, 2022). Ao longo de mais de uma década, o projeto ROS consolidou um amplo ecossistema de software para robótica, sustentando uma comunidade global de milhões de desenvolvedores e usuários engajados na melhoria contínua dessa plataforma (FOOTE, 2022).

Embora pesquisas em robótica já fossem realizadas antes do surgimento do ROS, grande parte do software desenvolvido era específica para cada robô, apresentando limitações quanto à reutilização, mesmo quando de código aberto. Em comparação com os sistemas robóticos existentes, o ROS apresenta desempenho superior nos seguintes aspectos (JOSEPH, 2017):

- **Desenvolvimento colaborativo:** O ROS, sendo de código aberto e gratuito para uso em pesquisa e indústria, permite que desenvolvedores ampliem suas funcionalidades por meio da adição de pacotes. A maioria desses pacotes opera sobre uma camada de abstração de hardware, facilitando sua reutilização em diferentes plataformas robóticas.
- **Suporte a linguagem:** O *framework* de comunicação do ROS pode ser implementado em diversas linguagens modernas. Entre as suportadas nativamente estão C++ e Python, além de bibliotecas mantidas pela comunidade que oferecem suporte a linguagens como Ada, C, Node.js, Rust, entre outra

- **Integração de bibliotecas:** O ROS dispõe de interfaces para diversas bibliotecas robóticas de terceiros, como *Open Source Computer Vision* (Open-CV), *Point Cloud Library* (PCL), Open-NI, Open-Rave e Orocos, permitindo aos desenvolvedores utilizá-las com facilidade em seus projetos.
- **Integração de simuladores:** O ROS é compatível com simuladores de código aberto, como Gazebo e Webots, e mantém boa integração com simuladores proprietários, como CoppeliaSim, possibilitando testes avançados em ambientes virtuais.
- **Teste de código:** O ROS inclui uma estrutura embutida de teste chamada *rostop*, que permite verificar a qualidade do código e identificar *bugs*.
- **Escalabilidade:** A arquitetura do ROS é projetada para ser escalável, permitindo a execução de tarefas de alto custo computacional em robôs que podem operar na nuvem ou em *clusters* heterogêneos.
- **Customização:** Por ser totalmente de código aberto e gratuito, o ROS pode ser adaptado conforme os requisitos específicos de cada robô. É possível utilizar apenas determinados componentes, como a plataforma de mensagens, ou ajustar o sistema para otimizar o desempenho de um robô específico.
- **Comunidade:** O ROS é um projeto orientado à comunidade, liderado principalmente pela *Open Source Robotics Foundation* (OSRF). O amplo suporte da comunidade facilita o início de novos desenvolvimentos em aplicações robóticas e promove a troca contínua de conhecimento entre desenvolvedores.

Diante desse contexto, considerando as diversas vantagens apresentadas, o ROS consolidou-se como um dos principais *frameworks* para robótica, sendo amplamente empregado no desenvolvimento de sistemas robóticos e de automação em escala global.

3.2.1 Conceitos básicos

No ROS, um *node* representa uma unidade de código executável que se comunica com outros nodes por meio de *messages*, formando um sistema integrado. O conceito central é a comunicação via *messages* entre nodes, sendo cada *node* responsável por uma função específica, o que facilita a reutilização e o desenvolvimento modular do software (PYO *et al.*, 2017).

O *package* constitui a unidade básica do ROS. As aplicações são estruturadas a partir de *packages*, que contêm arquivos de configuração para execução de *nodes* ou de outros *packages*, além de todas as dependências e bibliotecas necessárias para o funcionamento do sistema (PYO *et al.*, 2017). Um exemplo é o ROS Kinect, que foi uma das distribuições mais populares e integra uma biblioteca com 2772 *packages* (LALANCETTE; POUBEL, 2022).

A *message* é um conjunto de dados utilizado na comunicação entre *nodes*, podendo incluir variáveis como inteiros, *floats* e booleanos. *Messages* podem ser estruturadas, contendo outras *messages* em seu interior.

Os *topics* são elementos essenciais do gráfico ROS, funcionando como um barramento para a troca de *messages* entre *nodes*. Um *node* pode publicar informações em múltiplos *topics* simultaneamente e assinar diversos *topics* para receber dados.

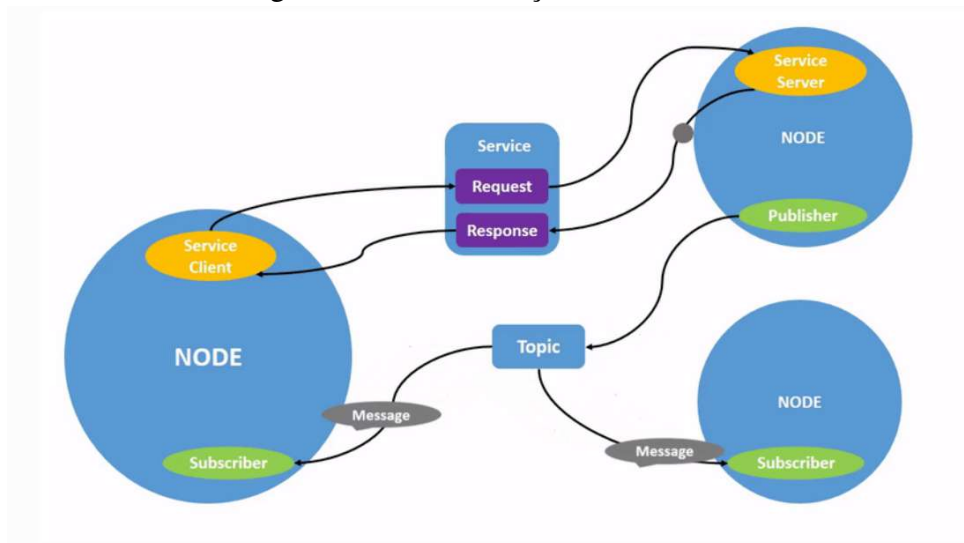
O *service* representa uma comunicação síncrona e bidirecional, na qual um cliente solicita a execução de uma tarefa específica e o servidor é responsável por fornecer a resposta correspondente.

As *actions*, exclusivas do ROS 2, destinam-se a tarefas de longa duração e são compostas por três elementos: meta, retorno e resultado. Construídas sobre *topics* e *services*, as *actions* são semelhantes aos *services*, porém possuem característica preemptiva, permitindo o cancelamento durante a execução. Além disso, elas fornecem atualizações contínuas, enquanto os *services* retornam apenas uma resposta única.

O *namespace* é um mecanismo para encapsular grupos de execução no ROS. Ele é especialmente útil quando há necessidade de executar múltiplas instâncias de um mesmo *node*, garantindo a separação de informações e evitando conflitos de nomenclatura.

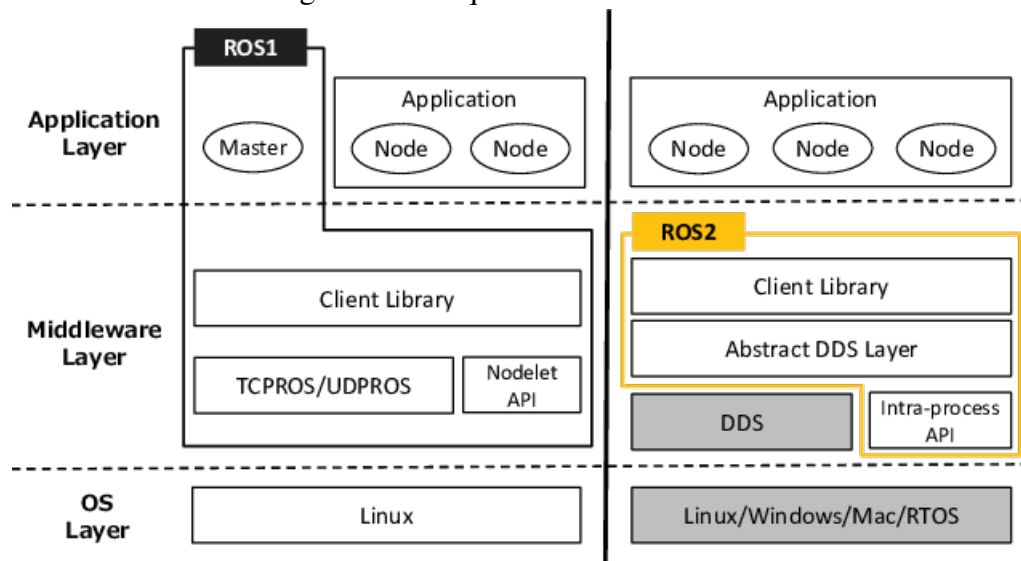
Sob esse viés, a Figura 12 ilustra o diagrama de comunicação entre *nodes* por meio de *topics* e *services* dentro do ambiente ROS.

Atualmente, o ROS encontra-se em sua segunda versão, denominada ROS 2. De maneira geral, o termo ROS é utilizado para se referir à comunidade e ao projeto como um conceito abrangente. Para diferenciar as versões, serão utilizados os termos ROS 1 e ROS 2 sempre que for necessário destacar características específicas de cada uma. A principal distinção entre as versões reside na arquitetura de comunicação entre os *nodes*. No ROS 1, a comunicação dependia de um *node* denominado *nodes*, responsável pelo gerenciamento das interações e utilizando os protocolos TCP/UDP. Já no ROS 2, a comunicação é realizada na camada de *middleware*, utilizando o protocolo *Data Distribution Service* (DDS) (MARUYAMA; KATO; AZUMI, 2016), conforme ilustrado na Figura 13.

Figura 12 – Comunicação entre *nodes*.

Fonte: Weon (2022).

Figura 13 – Arquitetura ROS 1/ROS 2.



Fonte: Maruyama, Kato e Azumi (2016).

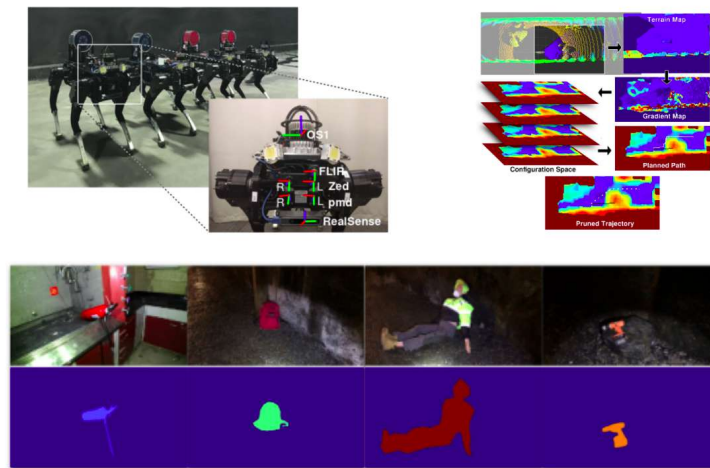
3.2.2 Casos de uso

Vilches (2022) divulgou uma lista pública com aproximadamente 400 empresas de renome que utilizam o ROS 1, ROS 2 ou ferramentas relacionadas para desenvolver produtos ou oferecer serviços. Entre essas organizações, encontram-se tanto empresas tradicionais, como Bosch, Boeing, BMW Group, John Deere e KUKA, quanto companhias mais recentes, como Toyota Woven Planet, TechnoYantra, Tangram Vision e Stogl Robotics). Algumas dessas empresas atuam especificamente na prestação de consultoria voltada ao uso do ROS.

No contexto de aplicações terrestres, Miller *et al.* (2020) relatam a utilização de múltiplos

robôs quadrúpedes da Ghost Robotics para exploração de túneis em minas subterrâneas (Figura 14). O ROS foi integrado ao sistema, oferecendo suporte ao controle da missão, ao mapeamento do ambiente e ao retorno. Durante a operação, o sistema processa e cruza informações com um banco de dados, permitindo a execução de ações específicas por meio de *Application Programming Interfaces* (APIs) que utilizam a interface *publisher-subscriber*.

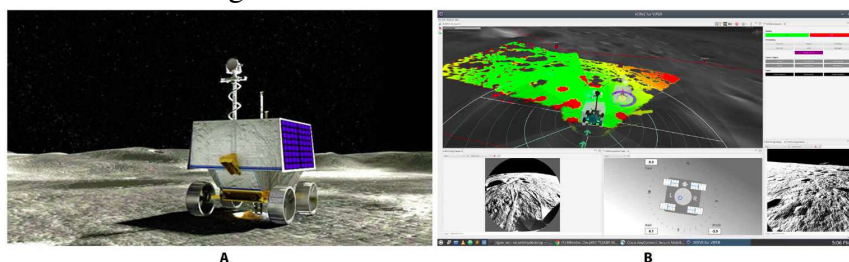
Figura 14 – Robôs quadrúpedes para exploração e mapeamento em minas subterrâneas.



Fonte: Miller *et al.* (2020).

Como outro exemplo de relevância, destaca-se a *National Aeronautics and Space Administration* (NASA), que conduz a missão *Volatiles Investigating Polar Exploration Rover* (VIPER), com o objetivo de explorar a região polar da Lua em busca de gelo e outros recursos. Diversas ferramentas de operação terrestre, módulos de cálculo e simulações de alta fidelidade utilizadas na missão são baseadas em ROS 2 e no simulador Gazebo. Conforme ilustrado na Figura 15, em “A” observa-se o robô VIPER sobre a superfície lunar (renderizado), enquanto em “B” é apresentado o *software* de operação e comando do sistema, conforme Macenski *et al.* (2022).

Figura 15 – Robô VIPER da NASA.



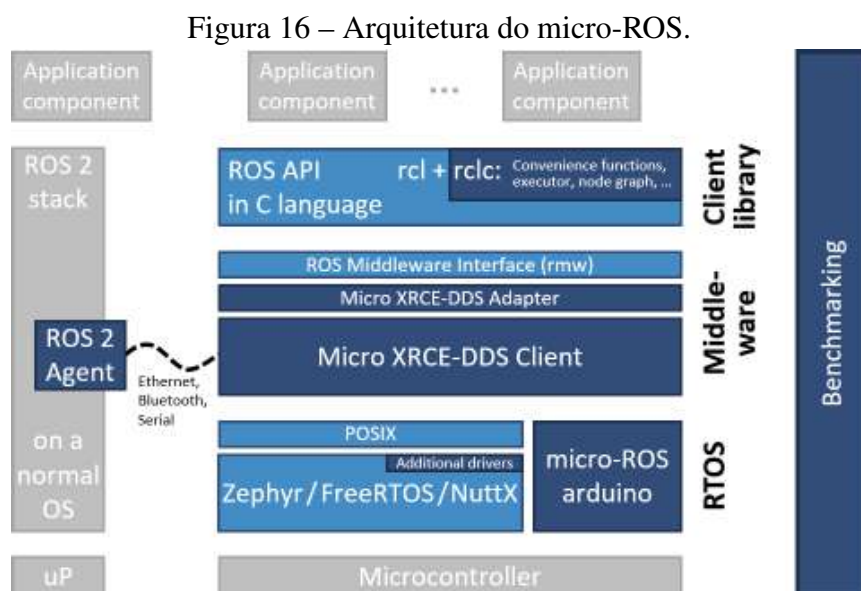
Fonte: Macenski *et al.* (2022).

3.2.3 Micro-ROS

Segundo ROS (2022), a missão da equipe responsável pelo desenvolvimento do micro-ROS é estabelecer a ponte entre microcontroladores com recursos limitados e processadores mais potentes em aplicações robóticas baseadas no ROS. O micro-ROS possibilita que microcontroladores, mais simples e econômicos que computadores ou notebooks, utilizem a infraestrutura do ROS. Esses dispositivos são amplamente empregados em robôs devido a vantagens como acesso direto ao *hardware*, baixa latência e eficiência energética.

O micro-ROS adota a arquitetura do ROS 2, aproveitando a conectividade fornecida pelo *middleware* por meio do *Data Distribution Service - Extremely Resource Constrained Environments* (DDS-XRCE), otimizado para microcontroladores. Diferentemente do ROS 2 tradicional, o micro-ROS utiliza o *Real Time Operating System* (RTOS) baseado na *Portable Operating System Interface* (POSIX), ao invés do Linux.

A Figura 16 ilustra o funcionamento da arquitetura do micro-ROS. Os componentes desenvolvidos especificamente para o micro-ROS estão destacados em azul escuro, enquanto os elementos padrão do ROS 2 utilizados aparecem em azul claro. O ROS 2 *Agent* é o nó responsável por estabelecer a ponte de comunicação entre o microcontrolador e o computador de maior capacidade de processamento.



Fonte: ROS (2022).

3.3 BITDOGLAB

Apesar do uso de protoboards ser um recurso didático habitual, ele pode apresentar desafios técnicos para docentes e discentes.

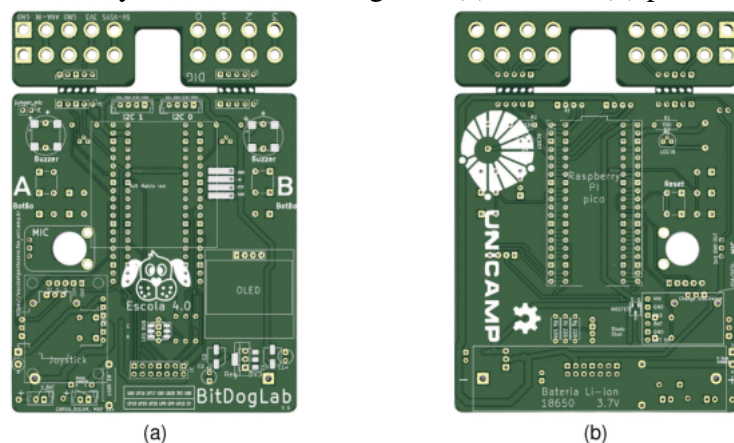
Isto posto, a BitDogLab, de acordo com Vieira *et al.* (2024), é uma ferramenta de hardware aberto desenvolvida com o apoio do projeto IEEE-EDS de 2023, a qual utiliza plataformas de hardware e software de código aberto para democratizar o acesso à tecnologia e à educação em ciência da computação e engenharia.

Esta placa de circuito impresso (PCB), conforme Fruett *et al.* (2024), usa como componente central um Raspberry Pi Pico, um módulo com um microcontrolador ARM de baixo custo e alto desempenho, que oferece flexibilidade de programação e recursos robustos para o ensino de conceitos de Ciências, Tecnologia, Engenharia, Artes e Matemática (STEAM).

Vieira *et al.* (2024) ainda aponta que a flexibilidade do design do BitDogLab possibilita uma expansão constante em software e hardware. Adicionalmente, estimula as contribuições da comunidade, fomentando o aprendizado em grupo e a inovação. Em contraste com as placas de ensaio, que podem representar obstáculos técnicos para professores e estudantes, o BitDogLab proporciona uma progressão de aprendizado mais suave.

Seu layout, de acordo com Fruett *et al.* (2024) e apresentado na Figura 17, possui o núcleo principal da placa, local onde os componentes são soldados, medindo 98 mm de comprimento, 88 mm de largura e 1,6 mm de espessura. A barra de terminais, elaborada para facilitar a conexão com pinças do tipo jacaré, tem 23 mm de comprimento, mantendo a mesma largura e espessura do núcleo principal.

Figura 17 – Layout do PCB BitDogLab: (a) frente e (b) parte traseira.



Fonte: Fruett *et al.* (2024).

Para mais, Fruett *et al.* (2024) ainda realiza uma análise comparativa do BitDogLab, Micro:bit e Arduino, onde destaca características distintas de cada plataforma em um contexto educacional, apresentada na tabela da Figura 18.

Figura 18 – Análise comparativa dos recursos do BitDogLab, Micro:bit e Arduino para uso educacional.

Feature	Micro:bit	Arduino	BitDogLab
Master Chip	nRF51822	ATmega328P	RP2040
Processor	32-bit ARM Cortex M0	8-bit AVR	Dual-core ARM Cortex M0+
Flash ROM	256KB	32KB	2MB
RAM	16KB	2KB	264KB
Oscillator Speed	16M	16M	Up to 133M
Display	5x5 LED - only red-dot matrix	1 LED	OLED 128x64 and a 5 x 5 RGB LED matrix
Button	3 buttons (A, B, Reset)	None	3 buttons (A, B, Reset)
Bluetooth	Yes	None	Optional (Pico W version)
wifi	None	None	Optional (Pico W version)
Accelerometer on board	Yes	None	None
Digital Compass on board	Yes	None	None
Powered By	USB/Battery Case	USB/AC power	USB, rechargeable battery or solar panel
Programming Language	MakeCode, Python	C/C++	MicroPython, C/C++
Additional Features	Edge Connector with support for alligator clamps	Analog Inputs	2 Independent Buzzers, Joystick, Neopixel, Analog Microphone, 2 I2C connectors, 1 14pin IDC connector, secure connectors for alligator clamp
Open Source	Partial	Yes	Yes

Fonte: Fruett *et al.* (2024).

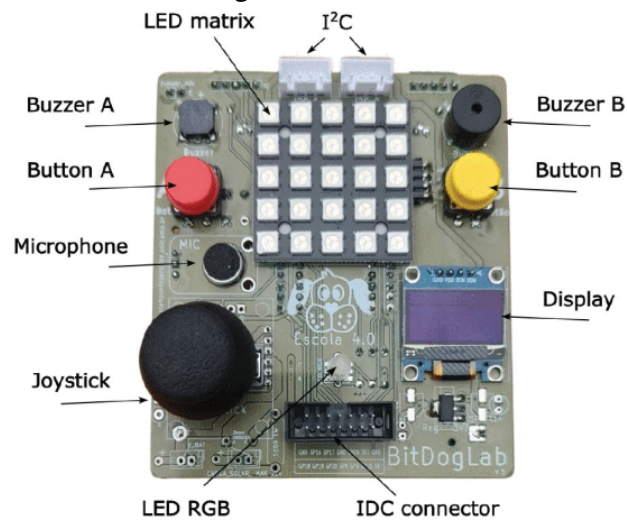
A PCB da placa BitDogLab com todos os componentes soldados é mostrada na Figura 19, com a vista frontal, e na Figura 20, com a visão ampliada dos terminais de conexão.

3.4 ESP-32

O ESP32, exposto na Figura 21, é uma série de microcontroladores SOC (*System On Chip* – Sistema em um Chip), desenvolvidos pela empresa chinesa Espressif Systems. É composto por um módulo Wi-Fi com conectividade 802.11b/g/n e Bluetooth de modo duplo integrados em sua placa de desenvolvimento, imprescindível para aplicações em IoT (BABIUCH; FOLTÝNEK; SMUTNÝ, 2019).

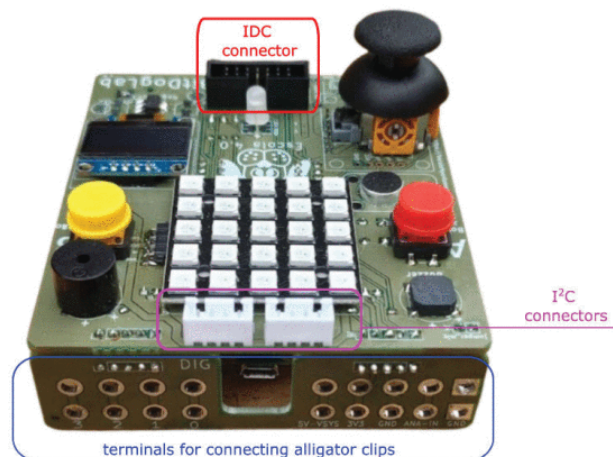
Este microcontrolador, da segunda geração de microcontroladores com soluções IoT,

Figura 19 – Vista frontal do BitDogLab montado com todos os seus componentes.



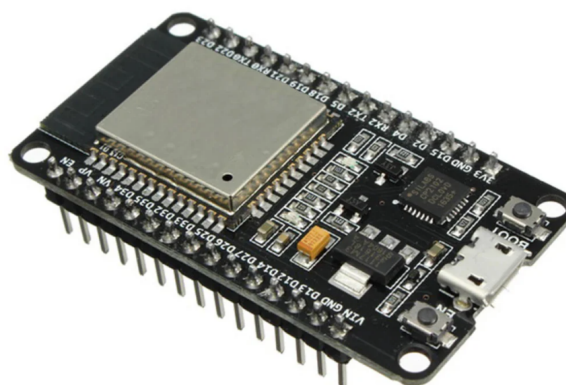
Fonte: Fruett *et al.* (2024).

Figura 20 – Visão ampliada dos terminais de conexão do BitDogLab que podem ser conectados com cliques jacaré.



Fonte: Fruett *et al.* (2024).

Figura 21 – Microcontrolador ESP-32.



Fonte: Retirada da Internet: <https://sl.bing.net/hTFIvGex6Gq> (2025).

fabricados pela Espressif, estabelece-se como sucessor direto do microcontrolador ESP8266. Esta série de microcontroladores compõe um microprocessador Xtensa LX6, fabricado pela empresa Tensilica, com arquitetura de 32 bits, em duas versões: de 1 ou 2 núcleos (single-core ou dualcore). A velocidade de clock é de até 240 MHz, com 520 KB de SRAM (*Static Random Access Memory* – Memória de Acesso Aleatório Estático)(DOKIC, 2020).

Além de contar com um módulo Wi-Fi mais rápido e com maior alcance, em comparação ao ESP-8266, em consoante com Rodrigues (2024), este microcontrolador possui também suporte para duas versões Bluetooth, versão 4.2 e BLE (*Bluetooth Low Energy*). Para mais, o ESP32 ainda conta com mais canais de entrada e saída (GPIO), conforme a Tabela 1 que apresenta uma comparação entre o ESP8266 e o ESP32.

Tabela 1 – Especificações do ESP8266 e ESP32.

Especificação	ESP8266	ESP32
Wifi 802.11b/g/n	2.4GHz, 72.2 Mbps	2.4GHz, 150 Mbps
Bluetooth	Não suporta	Bluetooth v4.2 e Bluetooth LE
CPU	Xtensa L106 de 1 núcleo	Xtensa LX6 de 1 ou 2 núcleos
Arquitetura	32 bits	32 bits
Frequência do Clock	80 MHz	160 a 240 MHz
Memória RAM	50 KB RAM	520 KB SRAM
Memória ROM	Não suporta	448 KB ROM
Memória Flash	512 KB	4 MB
GPIO	17 canais	36 canais
Conversor AD	1 canal de 10 bits	18 canais de 12 bits
Conversor DA	Não suporta	2 canais de 8 bits
SPI/I2C/I2S/UART (Serial)	2/1/2/2 canais	4/2/2/2 canais
PWM	8 canais	16 canais
Sensor de toque capacitivo	Não suporta	10 canais

Fonte: Rodrigues (2024).

Sob esse panorama, esse módulo é amplamente utilizada em projetos de Internet das coisas (IoT), conforme apresentado por Rodrigues e Elieser (2023), bem como outros projetos.

4 TRABALHOS RELACIONADOS

O desenvolvimento de plataformas robóticas móveis para fins educacionais, de pesquisa e industriais tem ganhado destaque nas últimas décadas, em virtude da crescente popularização de microcontroladores de baixo custo e do amadurecimento de frameworks de software abertos como o *Robot Operating System* (ROS). Essa combinação tem permitido a criação de sistemas robóticos cada vez mais acessíveis, modulares e escaláveis, abrindo espaço para experimentação e inovação em diferentes contextos.

No âmbito acadêmico, diversos trabalhos vêm propondo a utilização de robôs móveis como ferramentas didáticas para o ensino de conceitos de automação, eletrônica e programação embarcada. Dantas (2022) desenvolveu a adaptação de um robô aspirador Roomba® 614 iRobot em uma plataforma robótica móvel integrada ao *Robot Operating System 2* (ROS 2), apresentada na Figura 22, com o objetivo de fornecer um recurso didático para o ensino de robótica no curso de Engenharia Mecânica do Instituto Federal do Piauí (IFPI). Em sua proposta, o autor implementou uma arquitetura de comunicação baseada no micro-ROS, permitindo que o microcontrolador embarcado no robô atuasse como um nó ROS, capaz de enviar dados de sensores, receber comandos e operar dentro de um ambiente distribuído de controle. A escolha do Roomba se fundamentou na compatibilidade com o modelo educacional Create 2, amplamente utilizado em atividades acadêmicas internacionais, e na possibilidade de acoplar sensores e módulos adicionais por meio de uma base personalizada impressa em 3D. O trabalho demonstrou a viabilidade de integrar hardware comercial de baixo custo ao ecossistema ROS 2, evidenciando o potencial dessa abordagem para reforçar o aprendizado prático de cinemática, navegação e sistemas embarcados.

Figura 22 – Roomba montado com ESP 32, bateria e a base para adição de sensores variados.



Fonte: Dantas (2022).

Para mais, Vilanova (2022) propôs o desenvolvimento de uma plataforma robótica aérea baseada na aeronave DJI® Ryze Tello (Figura 23) no Robot Operating System (ROS 2) voltada para práticas educacionais de robótica móvel em laboratório, conforme previsto no Projeto Pedagógico do Curso (PPC) de Engenharia Mecânica do Instituto Federal do Piauí (IFPI). O autor defendeu a utilização de uma estrutura robótica experimental como recurso didático para aplicação dos conteúdos de controle, automação e sistemas embarcados, integrando sensores, atuadores e comunicação em rede em um ambiente real de aprendizagem. A pesquisa evidenciou que o uso do ROS 2 em contextos educacionais favorece a compreensão dos princípios de arquitetura modular e comunicação entre nós, permitindo que o estudante vivencie o ciclo completo de desenvolvimento robótico — da programação embarcada à análise de desempenho do sistema.

Figura 23 – DJI ® Ryze Tello Combo Boost.

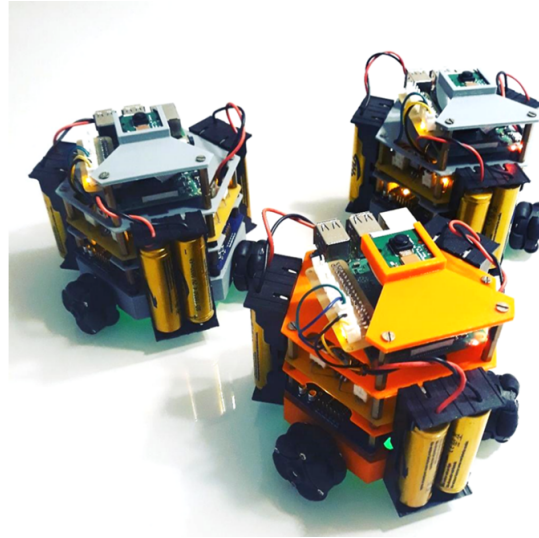


Fonte: Vilanova (2022).

Outro trabalho relevante para o contexto deste projeto é o Omnino (Figura 24), uma plataforma aberta de robô omnidirecional desenvolvida para pesquisas em robótica de enxame e ensino de sistemas distribuídos. De acordo com Ramalho *et al.* (2020), o Omnino foi projetado com arquitetura modular e de baixo custo, utilizando uma arquitetura de 3 rodas omnidirecionais e microcontroladores acessíveis, permitindo sua replicação em ambientes acadêmicos. A proposta do projeto destaca-se pela compatibilidade com o ROS e pela capacidade de operar de forma cooperativa em grupos de robôs, característica que amplia seu potencial para experimentação em algoritmos de controle coletivo e coordenação de agentes móveis.

Outrossim, vale ressaltar que no âmbito institucional, observa-se um avanço contínuo nas iniciativas de robótica do Instituto Federal do Piauí (IFPI). No ano de 2024, conforme discutido em Instituto Federal do Piauí (IFPI) (2024), quatro discentes dos cursos de Engenharia Mecânica

Figura 24 – Plataforma de Pesquisa e Ensino OMNINO.



Fonte: Ramalho *et al.* (2020).

e Análise e Desenvolvimento de Sistemas conquistaram a classificação da etapa regional para a fase nacional da competição AWS DeepRacer, uma competição de carros autônomos com inteligência artificial promovida pela Amazon Web Services que ocorreu presencialmente no evento satélite do Congresso da Sociedade Brasileira de Computação (CSBC) 2024, que ocorreu em Brasília, no período de 22 a 25 de julho do ano citado, embora nenhum tenha se posicionado entre os dez melhores do ranking nacional (Figura 25).

Figura 25 – Finalistas do AWS DeepRacer 2024.



Fonte: Instituto Federal do Piauí (IFPI) (2024).

Já em 2025, em consoante com Folha Expressa (2025), o desempenho apresentou progresso significativo: dez estudantes chegaram à final nacional, realizado novamente durante o CSBC, entre os dias 20 e 24 de julho, em Maceió (AL), sendo que cinco deles figuraram entre os dez melhores do país, ocupando o 2º, 3º, 4º, 7º e 9º lugares, resultado que garantiu ao IFPI a premiação de um robô físico AWS DeepRacer (Figura 26).

Figura 26 – Finalistas do AWS DeepRacer 2025.



Fonte: Folha Expressa (2025).

O AWS DeepRacer, de acordo com Petryshyn *et al.* (2024), constitui uma plataforma educacional da Amazon Web Services voltada ao ensino de aprendizado por reforço (*Reinforcement Learning* – RL) e robótica autônoma. A iniciativa integra simulação virtual e robótica física, permitindo que estudantes e profissionais construam, treinem e avaliem modelos de condução autônoma em um ambiente seguro e controlado. O robô físico DeepRacer (Figura 27), versão atualmente disponível no IFPI, possui arquitetura baseada em um pequeno veículo elétrico, equipado com câmera frontal para captura de imagens da pista, motores eletronicamente controlados e conectividade sem fio, possibilitando a execução de algoritmos de navegação autônoma.

Figura 27 – Robô AWS DeepRacer do IFPI.



Fonte: Elaborado pelo autor (2025).

5 METODOLOGIA

A metodologia adotada neste trabalho foi estruturada em etapas que abrangem desde o estudo teórico até a implementação prática do sistema robótico proposto. As fases foram divididas conforme segue.

5.1 PESQUISA BIBLIOGRÁFICA

Foi realizada uma pesquisa bibliográfica com o intuito de observar aplicações similares ou possíveis de serem feitas com robôs omnidirecionais e ROS. Os resultados dessa pesquisa se condensam no capítulo 4, o qual é denominado "TRABALHOS RELACIONADOS".

5.2 FERRAMENTAS E MATERIAIS

O desenvolvimento e a validação do trabalho ocorreram no Laboratório de Inovação e Pesquisa Aplicada (LABIRAS), um laboratório do Instituto Federal de Educação, Ciência e Tecnologia do Piauí (IFPI) Campus Teresina Central, utilizando equipamentos e ferramentas de eletrônica, informática e prototipagem disponíveis, incluindo uma impressora 3D PRO-GTMAX Core H5 (Figura 28) para confecção da estrutura física.

Figura 28 – Impressora 3D PRO-GTMAX Core H5.



Fonte: Elaborado pelo autor (2025).

Na etapa de desenvolvimento estrutural do projeto, optou-se pela utilização do filamento ABS (Acrilonitrila Butadieno Estireno) como material base para a impressão 3D das peças estruturais. Essa escolha foi motivada por suas propriedades mecânicas superiores em relação

a outros termoplásticos de uso comum, como o PLA, destacando-se pela maior resistência a impactos, durabilidade e capacidade de suportar temperaturas mais elevadas.

O uso do ABS permitiu a fabricação de componentes com boa rigidez estrutural e precisão dimensional, características essenciais para garantir o correto acoplamento dos motores, sensores e módulos eletrônicos da plataforma. As peças foram modeladas no software Autodesk Fusion 360, respeitando os parâmetros de montagem do sistema mecânico e dos módulos de controle. Isto posto, durante o processo de fabricação, adotaram-se parâmetros de impressão específicos para o ABS, como mesa aquecida a cerca de 110 °C e temperatura de extrusão em torno de 230 °C.

Outrossim, o sistema de tração do projeto foi composto por quatro motores JGA25-370 com *encoders* acoplados (Figura 29), responsáveis pelo acionamento individual de cada roda mecanum. Esses motores de corrente contínua apresentam um conjunto integrado de redutor metálico e sensor de realimentação, o que permite combinar torque elevado e controle de velocidade e posição. O *encoder* de dois canais, instalado no eixo traseiro do motor, gera pulsos proporcionais à rotação do eixo, possibilitando a estimativa da velocidade angular e da distância percorrida. Além disso, o baixo consumo de corrente e a tensão nominal de operação entre 6 e 12 V tornam o modelo JGA25-370 adequado para aplicações educacionais e experimentais, proporcionando um equilíbrio entre desempenho e eficiência energética no contexto de robótica móvel.

Figura 29 – Motores JGA25-370 com *encoders* acoplados.



Fonte: Elaborado pelo autor (2025).

Para mais, a presente proposta tinha por objetivo conseguir controlar remotamente outros dispositivos robóticos móveis terrestres já integrados ao sistema operacional robótico, seja físico ou simulado. Dito isso, foram utilizadas as plataformas robóticas móveis terrestres diferenciais (Figura 30), confeccionadas com base no Roomba® 614 iRobot e também presente no LABIRAS,

como ferramentas de testes para validação da possibilidade de controle de diferentes dispositivos através da solução proposta.

Figura 30 – Plataformas robóticas móveis terrestres do LABIRAS.



Fonte: Elaborado pelo autor (2025).

Diante disso, o desenvolvimento do robô omnidirecional proposto neste trabalho buscava contribuir para o aprimoramento das práticas educacionais e de pesquisa no que diz respeito à utilização de robôs com essa configuração, que vem ganhando cada vez mais espaço nos setores industriais e acadêmicos devido à sua alta manobrabilidade e eficiência em ambientes restritos.

Além disso, a integração de uma solução baseada em Internet das Coisas (IoT) visava fornecer uma abordagem prática e escalável para o gerenciamento remoto de múltiplos dispositivos robóticos. Essa solução permitirá que diferentes plataformas móveis possam ser monitoradas e controladas de maneira centralizada, otimizando os processos de operação e testes, tanto em ambientes reais quanto simulados. Dessa forma, esperava-se que os resultados obtidos pudessem servir de base para futuras aplicações em ambientes educacionais, industriais e de pesquisa, promovendo maior autonomia, controle e flexibilidade no uso de sistemas robóticos distribuídos.

5.3 SOFTWARE

O software de controle foi desenvolvido com base no ROS 2 Humble, através da linguagem de programação C++, executado em uma estação Linux. A arquitetura do sistema foi projetada para permitir a comunicação entre dois módulos distintos: o robô OmniBot e o controle remoto BitDogLab.

O OmniBot, equipado com um ESP32 DevKit V1, foi programado em C++ utilizando o Integrated Development Environment (IDE) Visual Studio Code, que possui várias extensões

que auxiliam no desenvolvimento de software. Dentre essas extensões, vale ressaltar o uso da extensão PlatformIO, que auxilia o envio do algoritmos para dentro do microcontrolador, além de auxiliar no gerenciamento de bibliotecas.

Isto posto, o microcontrolador atua como nó cliente do micro-ROS Agent, conectando-se ao servidor ROS por meio de rede Wi-Fi. Após a inicialização, o robô publica mensagens de telemetria e subscreve comandos de velocidade angular e linear (`cmd_vel`), enviados a partir de um nó de controle.

Por sua vez, o controle remoto, implementado na BitDogLab (Raspberry Pi Pico W), foi desenvolvido em C utilizando o Visual Studio Code e a extensão Raspberry Pi Pico, a extensão oficial para desenvolvimento no Raspberry Pi Pico, a qual equipa um conjunto de ferramentas projetadas para otimizar projetos Pico usando o Visual Studio Code e o SDK oficial.

O controle remoto opera como interface humana de controle (IHM), capturando os estados dos botões, do joystick e exibindo informações no display OLED. A comunicação é feita via requisições HTTP, nas quais o Pico W atua como cliente e o servidor ROS 2 processa as requisições, convertendo-as em mensagens publicadas nos tópicos apropriados.

Para garantir modularidade, os algoritmos foram estruturados em camadas: controle de rede, controle de movimento e camada de abstração dos sensores. A troca de dados é organizada em formato JSON, o que simplifica o tratamento das mensagens e favorece interoperabilidade com aplicações web e outros dispositivos IoT.

5.4 HARDWARE

A estrutura física do OmniBot foi confeccionada por meio de impressão 3D, utilizando a impressora PRO-GTMAX Core H5. O chassi foi composto por um conjunto modular, projetado para facilitar substituições e ampliações. O sistema de locomoção é composto por quatro rodas mecanum, cada uma acionada por motores DC independentes com *encoders*, o que permite movimentos translacionais e rotações em qualquer direção sem necessidade de alteração de orientação.

A arquitetura eletrônica foi montada em camadas:

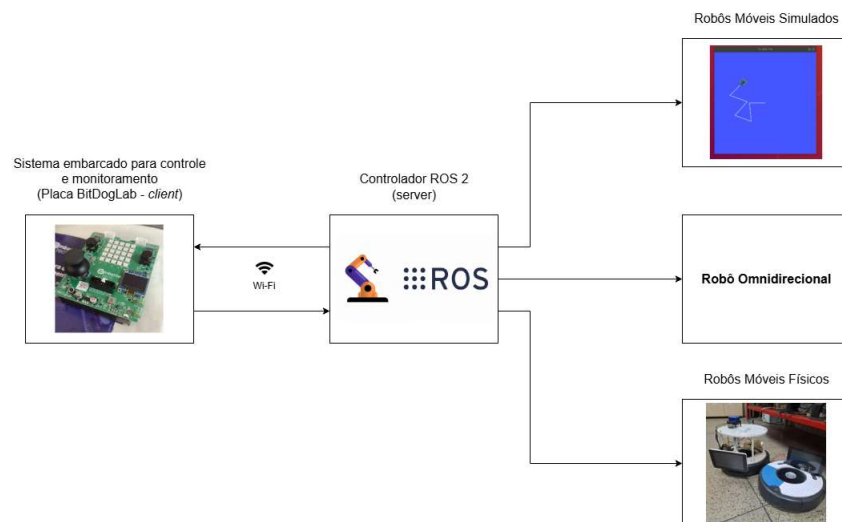
- Camada de controle: ESP32 DevKit V1 responsável pelo processamento principal;
- Camada de potência: ponte H L298N para acionamento dos motores;

- Camada de sensoriamento: sensores *encoders* fixados nos motores DC para contagem de pulsos do movimento dos respectivos motores;
- Camada de comunicação: módulo Wi-Fi integrado do ESP32, com antena interna;
- Alimentação: bateria 12V.

Por sua vez, o controle remoto BitDogLab opera com uma interface de joystick analógico, dois botões digitais (A e B) e um display OLED 128x64. Esse módulo permite o envio manual de comandos, oferecendo um controle ergonômico e de fácil integração com o ROS 2.

A Figura 31 ilustra a arquitetura geral de comunicação entre os dispositivos, destacando o fluxo bidirecional entre o OmniBot, o BitDogLab e o ROS 2 Agent.

Figura 31 – Diagrama com a proposta de arquitetura de conexão entre os dispositivos.



Fonte: Elaborado pelo autor (2025).

5.5 PADRONIZAÇÃO

A padronização dos códigos e mensagens, bem como a estrutura de diretórios segue o modelo de packages ROS 2, conforme apresentado na Figura 32 com base no modelo descrito pela documentação oficial ROS (Open Robotics, 2025), e os tópicos utilizam nomes normalizados (/omnibot/cmd_vel, /omnibot/wheels), de modo a facilitar a compatibilidade com outros projetos desenvolvidos no LABIRAS. Essa abordagem permite a expansão futura do sistema para múltiplos robôs e integração com simuladores, como o Gazebo.

Figura 32 – Estrutura de projeto ROS 2.

```

workspace_folder/
  src/
    cpp_package_1/
      CMakeLists.txt
      include/cpp_package_1/
      package.xml
      src/

    py_package_1/
      package.xml
      resource/py_package_1
      setup.cfg
      setup.py
      py_package_1/
    ...
    cpp_package_n/
      CMakeLists.txt
      include/cpp_package_n/
      package.xml
      src/

```

Fonte: Open Robotics (2025).

Para mais, a estrutura da mensagem `/omnibot_interface_msgs/wheels`, utilizada para intercomunicação através do tópico `/omnibot/wheels`, seguiu como referência o padrão das mensagens do tipo `/create_msgs/msg/wheels`, utilizado nas plataformas robóticas móveis terrestres presentes no LABIRAS, onde são publicados as informações referentes às rodas do Roomba. Essa estrutura possui seis campos para as rodas, três para roda esquerda e três para a direita, mais o *header*. Os campos das rodas são:

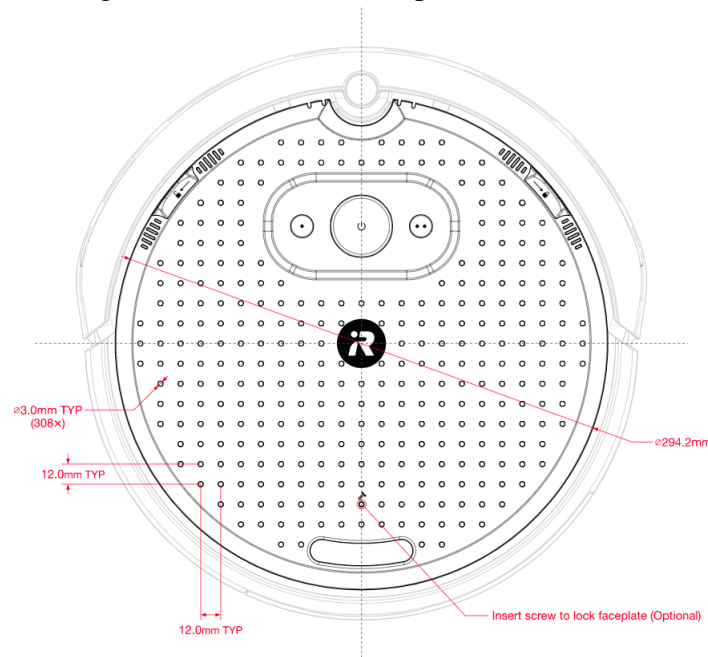
- ***drop_left***: Informa em booleano se a roda esquerda está fora do chão ou não;
- ***encoder_counts_left***: Número de contagens do codificador da roda esquerda;
- ***velocity_left***: Velocidade da roda esquerda em mm/s;
- ***drop_right***: Informa em booleano se a roda direita está fora do chão ou não;
- ***encoder_counts_right***: Número de contagens do codificador da roda direita;
- ***velocity_right***: Velocidade da roda direita em mm/s.

Diante disso, foram inseridos a estrutura da mensagem `/omnibot_interface_msgs/wheels` 4 campos referentes as rodas, sendo 1 para cada roda, mais o header. Os campos das rodas são:

- ***encoder_counts_front_left***: Número de contagens do codificador da roda frontal esquerda;
- ***encoder_counts_front_right***: Número de contagens do codificador da roda frontal direita;
- ***encoder_counts_back_left***: Número de contagens do codificador da roda traseira esquerda;
- ***encoder_counts_back_right***: Número de contagens do codificador da roda traseira direita;

No mais, planejou-se que a tampa da estrutura física do robô omnidirecional apresentasse um padrão de furação análogo ao do modelo Create 3 da iRobot, um padrão já adotado comercialmente e pelos dispositivos móveis terrestres presentes no LABIRAS. O padrão se dá por furos de 3mm com 12mm de espaçamento entre eles, como mostra a Figura 33.

Figura 33 – Dimensões da placa do Create 3.



Fonte: Shamlan (2022).

5.6 DOCUMENTAÇÃO

Uma documentação técnica completa foi produzida, incluindo esquemas elétricos, diagramas de blocos, fluxogramas de comunicação e instruções de compilação e operação. O objetivo é permitir que futuros pesquisadores e estudantes possam reproduzir e expandir o projeto. O material foi estruturado em formato digital, com orientações sobre montagem, manutenção e boas práticas de uso em laboratório.

6 RESULTADOS

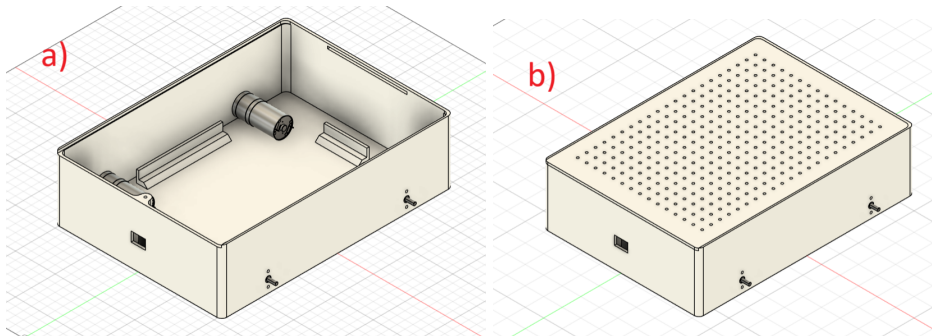
Os resultados obtidos com o desenvolvimento do OmniBot demonstram o êxito da proposta em integrar conceitos de robótica móvel, Internet das Coisas e controle distribuído.

Diante disso, este capítulo descreve como o robô a estrutura física do robô, como está funcionando atualmente, como testar os possíveis comandos e visualizar os dados que o robô disponibiliza. Também será apresentado os testes de controle de outras plataformas robóticas móveis terrestres físicas e simuladas com a solução IoT proposta com a BitDogLab e o ROS 2.

6.1 ESTRUTURA FÍSICA E MONTAGEM

A concepção do projeto iniciou-se pela modelagem tridimensional da estrutura do robô, desenvolvida no software Autodesk Fusion 360, onde foram definidos os parâmetros geométricos, pontos de fixação e o arranjo interno dos componentes, conforme apresentado na Figura 34. Essa etapa foi essencial para prever o posicionamento dos motores, placas e sensores, além de otimizar a distribuição de massa e o espaço interno para manutenção.

Figura 34 – Modelagem 3D do OmniBot no Autodesk Fusion 360 a) Vista interna b) Vista externa.



Fonte: Elaborado pelo autor (2025).

Durante os testes práticos realizados no LABIRAS, a plataforma mostrou-se plenamente funcional, com os deslocamentos esperados em todas as direções possíveis. A matriz cinemática das rodas mecanum foi devidamente implementada, permitindo que o robô se movesse lateralmente, diagonalmente e rotacionasse sobre o próprio eixo sem perda de estabilidade. O controle de velocidade foi ajustado por meio de modulação PWM, garantindo suavidade nas transições.

A Figura 35 apresenta o OmniBot completamente montado, após a impressão das peças estruturais e a instalação dos sistemas eletrônicos e mecânicos.

A comunicação entre o ESP32 e o ROS 2 Agent mostrou-se estável, onde o nó micro-ROS

Figura 35 – Montagem física do robô OmniBot.

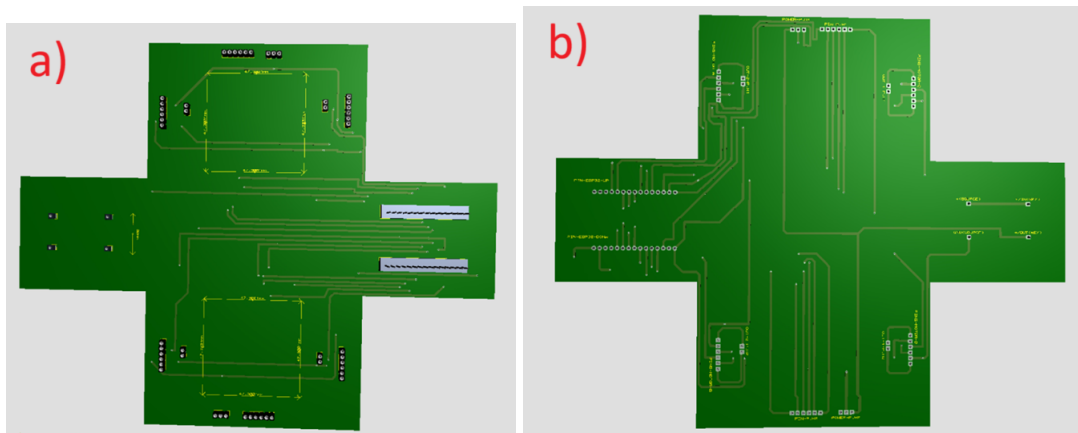


Fonte: Elaborado pelo autor (2025).

foi capaz de publicar informações de telemetria das rodas do robô (`omnibot_interfaces/Wheels`) e receber comandos (`geometry_msgs/Twist`) de forma contínua, validando a bidirecionalidade da arquitetura.

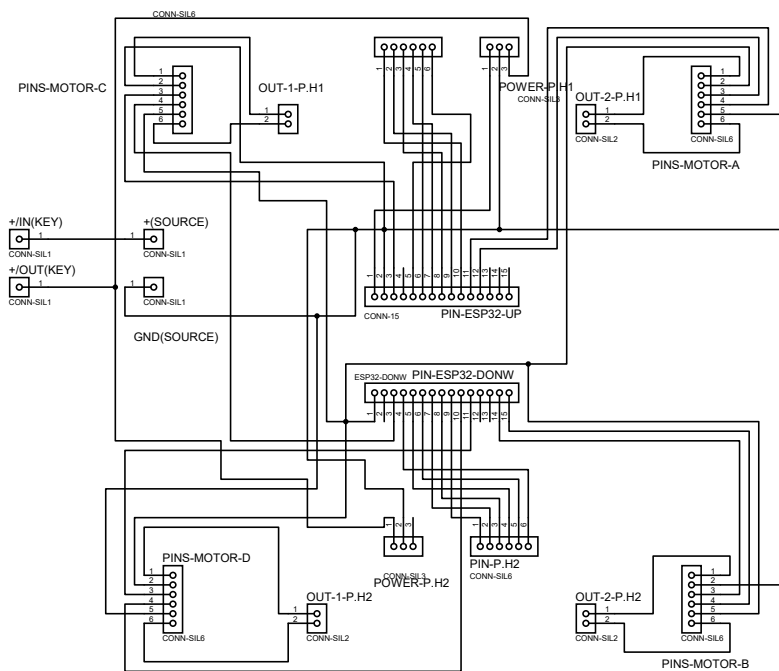
A placa de circuito impresso (PCB) desenvolvida para o projeto foi desenhada com o objetivo de organizar a ligação dos motores, *encoders*, pontes H, microcontrolador e módulos de alimentação. A Figura 36 apresenta o layout final da PCB, enquanto a Figura 37 exibe o diagrama esquemático do sistema eletrônico completo, mostrando a interligação entre o microcontrolador, drivers de motor e sensores.

Figura 36 – Layout da placa PCB de controle do OmniBot a) Vista superior b) Vista inferior.



Fonte: Elaborado pelo autor (2025).

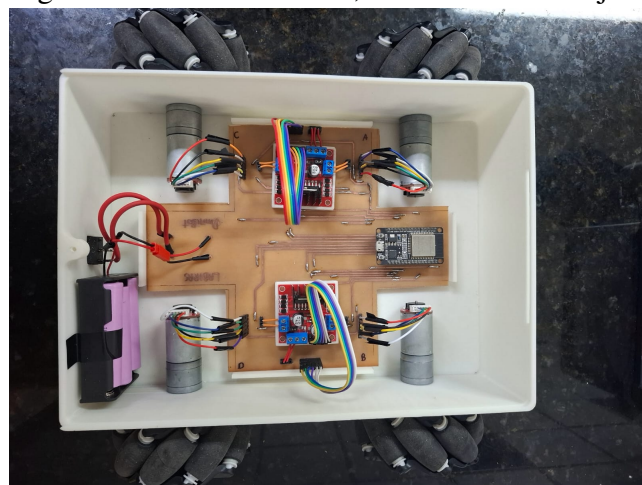
Figura 37 – Diagrama de esquemático conexões eletrônicas do OmniBot.



Fonte: Elaborado pelo autor (2025).

Isto posto, a Figura 38 mostra a montagem interna do robô, evidenciando a fixação do microcontrolador ESP32 DevKit V1, da ponte H L298N, bem como das conexões de alimentação e sinal. A disposição foi planejada para facilitar a montagem e manutenção de componentes.

Figura 38 – Montagem interna do OmniBot, mostrando o arranjo dos componentes.

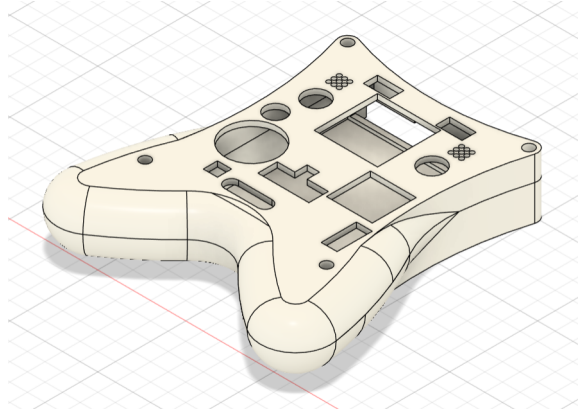


Fonte: Elaborado pelo autor (2025).

Para mais, a fim de acomodar a BitDogLab e permitir seu uso como controle portátil,

foi projetado um case dedicado, também modelado no Fusion 360, que garante proteção física, ergonomia e praticidade no manuseio. A Figura 39 apresenta o modelo virtual do case, e a Figura 40, sua versão física impressa em 3D e acoplada a placa BitDogLab.

Figura 39 – Modelagem do case desenvolvido para a BitDogLab.



Fonte: Elaborado pelo autor (2025).

Figura 40 – Case da BitDogLab impresso em 3D a) Vista frontal b) Vista posterior.



Fonte: Elaborado pelo autor (2025).

Em termos de construção, o chassi impresso em 3D, bem como o case da placa BitDogLab, apresentaram resistência adequada às solicitações mecânicas e facilidade de manutenção. A modularidade do projeto possibilitou a substituição de componentes sem necessidade de reconstrução integral da estrutura, reforçando o caráter educacional da plataforma.

6.2 OPERAÇÃO DO ROBÔ

Para a inicialização do robô, é necessário que o *micro-ROS Agent* esteja executando na máquina, a qual o endereço IP foi configurado no algoritmo do robô, e ambos estejam configurados na mesma rede, conforme exposto na Figura 41. O *micro-ROS Agent* pode ser

instalado criando um *ROS workspace* e clonando o *package micro_ros_setup* e compilando manualmente, ou utilizando uma imagem *Docker* como mostra Finocchiario (2020), onde adotou-se a primeira opção.

Figura 41 – Configuração dos parâmetros do robô.

```

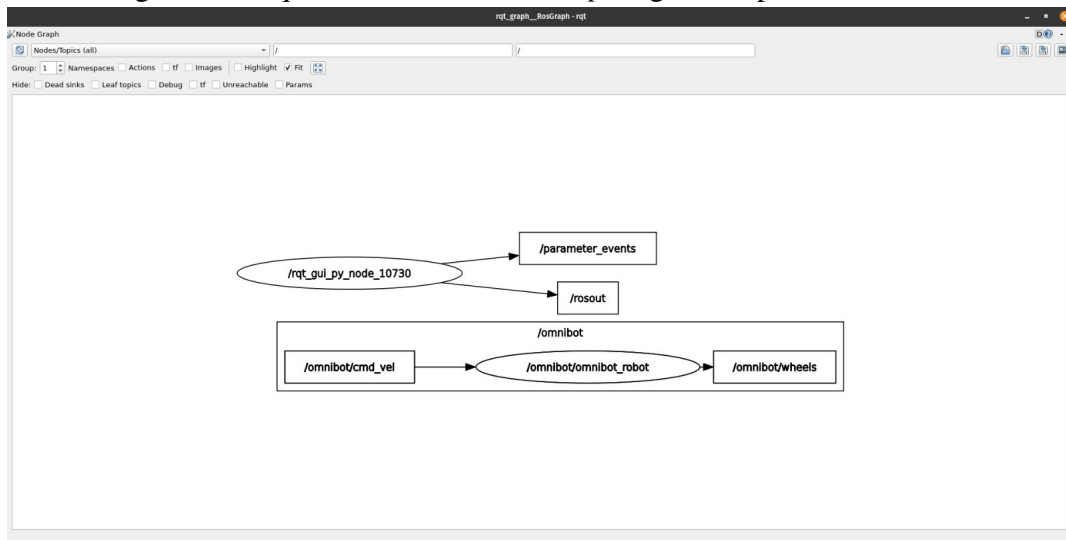
78
79 // Create node object
80 rcl_node_t node;
81 const char *node_name_ = "omnibot_robot";
82
83 // Node namespace (Can remain empty "")
84 const char *namespace_ = "omnibot";
85
86 char *ssid = "your_wifi_ssid";
87 char *password = "your_wifi_password";
88 IPAddress ip(192, 168, 137, 121); // IPV4
89

```

Fonte: Elaborado pelo autor (2025).

Após a conexão, o *Agent* cria o *node* e os *topics*, podendo ser visualizados usando a ferramenta *rqt_graph* do ROS (Figura 42). São dois *topics* e um *node*, todos possuindo o prefixo do *namespace* que foi definido nos parâmetros do robô. Nesse caso o *namespace* é */omnibot*.

Figura 42 – *rqt* mostrando o *node* e *topics* gerados pelo micro-ROS.



Fonte: Elaborado pelo autor (2025).

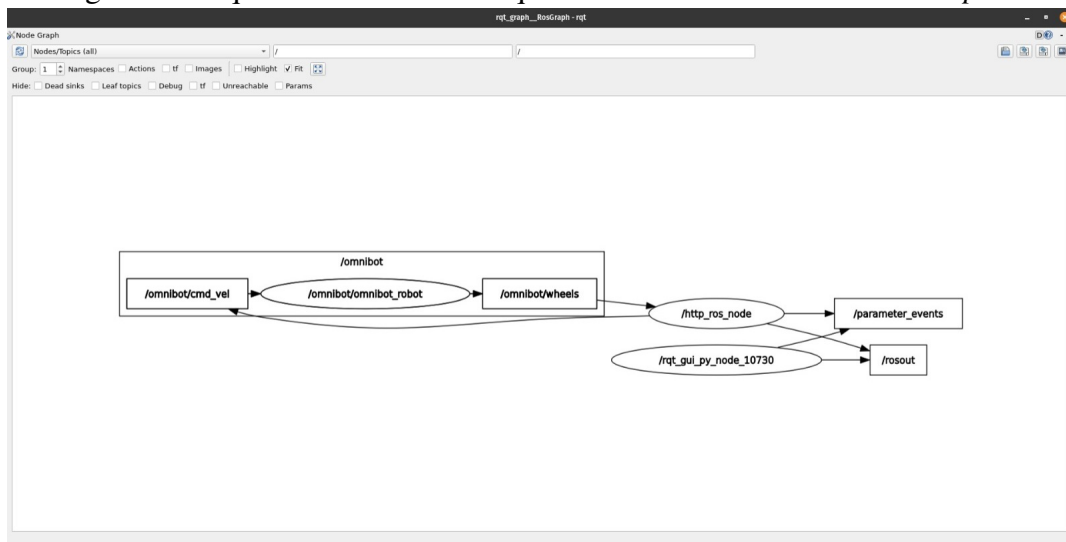
Para movimentação o ESP32 espera mensagens do tipo **geometry_msgs/msg/Twist** no *topic /omnibot/cmd_vel*. A mensagem é definida por dois vetores de velocidade, um linear com x, y e z, e um angular também com x, y e z. O robô só consegue se mover no plano, andando nas direções x e y, e rotacionando em z. Sendo assim, só as componentes lineares x e y, e angular z

serão necessárias. O ESP32 espera esses valores e comanda os motores de acordo.

Essa estrutura demonstra que o OmniBot é capaz de ser operado tanto em modo manual, quanto remoto, a partir de uma interface ROS, o que viabiliza futuras aplicações em ambientes distribuídos.

Ainda assim, para publicar em `/omnibot/cmd_vel` foi utilizado o *node* desenvolvido em C++ que funciona como *middleware* ROS 2 de comunicação entre o robô e a placa BitDogLab. Esse *node* cuida de receber os comandos via requisições HTTP da placa de controle e transformar em mensagens pdo tipo `geometry_msgs/msg/Twist` para serem enviadas ao Omnibot através do tópico especificado, bem como retornar a BitDogLab os dados dos *encoders* do robô provenientes do tópico `/omnibot/wheels`. Após a inicialização desse *node*, a estrutura de nós e tópicos ROS ficam conforme exposto pela ferramenta rqt da Figura 43.

Figura 43 – rqt mostrando toda a arquitetura de conexão dos *nodes* e *topics*.

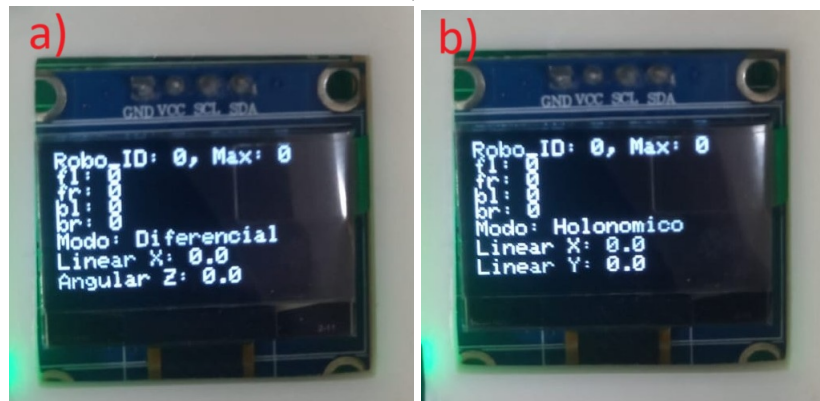


Fonte: Elaborado pelo autor (2025).

O *node* desenvolvido gerencia de forma dinâmica a descoberta e exclusão de robôs móveis controláveis. Isso é possível através de uma busca por todos os tópicos com terminação `cmd_vel` e `wheels`, e por uma análise se há *subscribers* nesses tópicos `cmd_vel` e se há *publishers* nos tópicos com `wheels`. Através dessa busca, é possível identificar novos robôs conectados ao ROS e robôs que foram desconectados, podendo assim gerenciar dinamicamente os dispositivos.

Dessa forma, a quantidade e o dispositivo robótico que será controlado podem ser visualizados através da IHM exibida pela tela OLED da BitDogLab, bem como as informações das leituras dos *encoders* das rodas do robô e dos dados da mensagem `cmd_vel` (linear x e y, e angular z), conforme elucidado na Figura 44.

Figura 44 – Informações do display OLED da BitDogLab do funcionamento em a) Modo diferencial b) Modo holonômico.



Fonte: Elaborado pelo autor (2025).

Diante disso, assim como Vilanova (2022), verifica-se a possibilidade de uso da plataforma robótica elaborada como recurso didático para aplicação dos conteúdos de controle, automação e sistemas embarcados, integrando sensores, atuadores e comunicação em rede em um ambiente real de aprendizagem.

No mais, análogo a Ramalho *et al.* (2020), obteve-se os mesmos princípios de modularidade, acessibilidade e interoperabilidade, mas com foco em aplicações educacionais e integração IoT para controle remoto e monitoramento via rede sem fio. Outro ponto, diferente de Ramalho *et al.* (2020), montou-se uma plataforma robótica omnidirecional com arquitetura de 4 rodas *mecanum*, mas que também apresenta compatibilidade com o ROS 2.

6.3 OPERAÇÃO DE OUTRAS PLATAFORMAS ROBÓTICAS

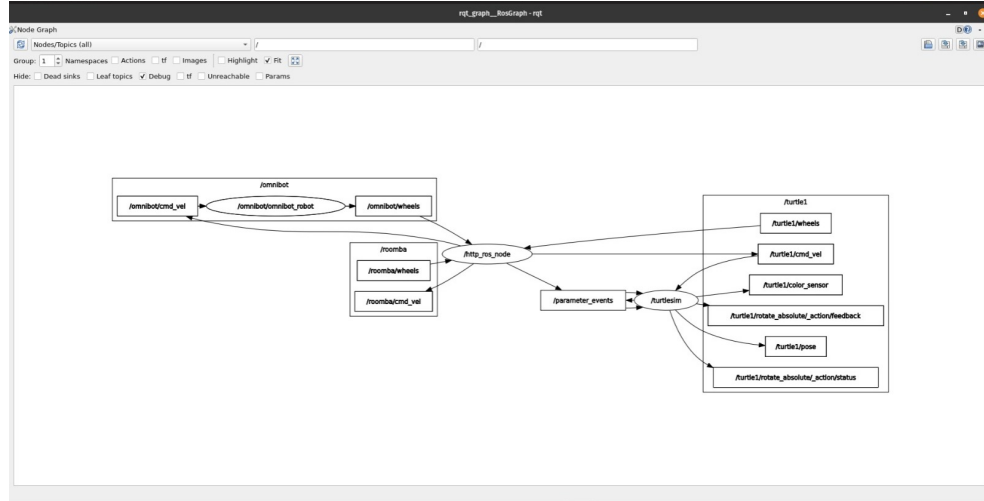
Com a arquitetura desenvolvida do projeto, conforme destacado na Figura 31, foi possível estabelecer conexão com outras plataformas robóticas móveis, físicas e simuladas.

Através do *node* ROS 2 desenvolvido, obteve-se validação de controle da plataforma robótica móvel terrestre diferencial do laboratório LABIRAS, desenvolvida com base no Roomba® 614 iRobot. Durante os testes, foi possível controlar a movimentação da mesma através da solução Iot elaborada, bem como coletar os dados dos *encoders* de suas rodas e disponibilizar na IHM elaborada.

Outrossim, o teste estendeu-se em validar, ao mesmo tempo, o controle de plataformas robóticas móveis terrestres simuladas. Para tal, utilizou-se o Turtlesim, um simulador de aprendizado ROS 2, também obtendo resposta positiva. Nas Figuras 45 e 46 são expostas a estrutura de nós e tópicos ROS através da ferramenta rqt e a IHM exibida pela tela OLED da

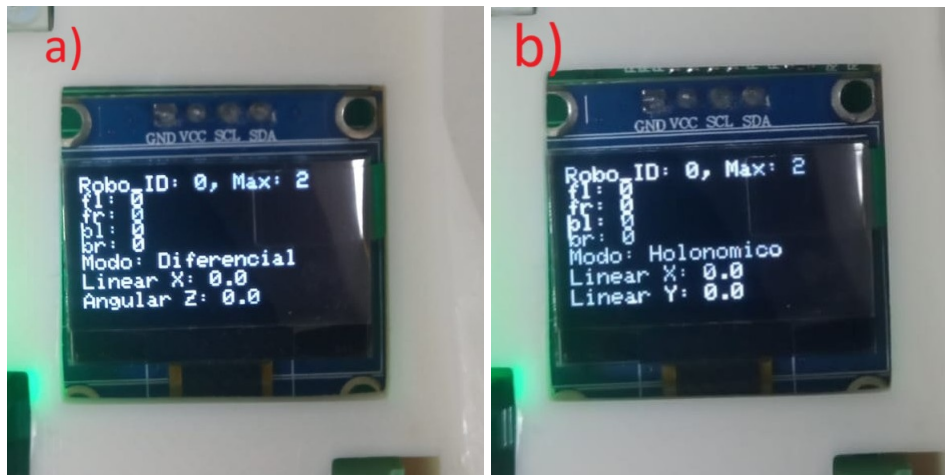
BitDogLab, respectivamente, durante a operação desses 3 dispositivos de forma simultânea.

Figura 45 – rqt mostrando toda a arquitetura de conexão dos *nodes* e *topics* para operação simultânea de 3 dispositivos robóticos.



Fonte: Elaborado pelo autor (2025).

Figura 46 – Informações do display OLED da BitDogLab do funcionamento durante a operação simultânea de 3 dispositivos robóticos em a) Modo diferencial b) Modo holonômico.



Fonte: Elaborado pelo autor (2025).

Destarte, assim, o controle remoto via BitDogLab apresentou desempenho satisfatório em termos de responsividade e confiabilidade da transmissão. Além disso, a tela OLED exibiu corretamente os parâmetros de conexão e as informações acerca dos *encoders* do robô em tempo real, oferecendo ao operador um feedback constante durante a operação.

Esses resultados indicam que o sistema é uma ferramenta eficiente para o ensino de conceitos como cinemática, controle e integração de sistemas embarcados com IoT, atendendo plenamente aos objetivos propostos.

7 CONCLUSÃO

O desenvolvimento do OmniBot representou a consolidação de uma plataforma robótica móvel omnidirecional voltada ao ensino e à pesquisa em Engenharia Mecânica, integrando conceitos de robótica móvel, Internet das Coisas (IoT) e controle distribuído via ROS 2. A proposta atendeu plenamente aos objetivos definidos, ao demonstrar a viabilidade de construir um robô modular e funcional, capaz de operar em conjunto com sistemas de software livres e escaláveis.

A adoção do ESP32 DevKit V1 como unidade de processamento e da BitDogLab como interface de controle remoto revelou-se eficaz, comprovando que tecnologias acessíveis podem ser aplicadas de forma robusta em ambientes acadêmicos. A arquitetura baseada em micro-ROS e HTTP garantiu flexibilidade de integração com outros dispositivos e sistemas em rede, tornando o OmniBot uma base para futuras aplicações educacionais e laboratoriais.

No âmbito educacional, o OmniBot se destacou como ferramenta didática de grande potencial, capaz de auxiliar na aplicação prática de conteúdos previstos no Projeto Pedagógico do Curso (PPC) de Engenharia Mecânica do IFPI, especialmente nas disciplinas de Robótica e Automação. A plataforma permite que estudantes explorem, de maneira prática, os princípios de cinemática, controle, sensoriamento e comunicação em rede, promovendo um aprendizado experimental alinhado às demandas da indústria.

Os testes realizados no LABIRAS confirmaram o pleno funcionamento do sistema em todos os modos de operação. O robô apresentou movimentação adequada, graças à matriz cinemática das rodas *mecanum*, e respondeu de forma estável aos comandos remotos via BitDogLab e ROS 2. O chassi impresso em 3D, projetado no Autodesk Fusion 360, demonstrou boa resistência mecânica, modularidade e facilidade de manutenção, validando o uso da manufatura aditiva como processo eficiente para prototipagem de plataformas robóticas educacionais.

Além do sucesso técnico e funcional, o OmniBot reforça a importância de projetos integradores no contexto da formação em Engenharia Mecânica, unindo aspectos de projeto mecânico, eletrônica embarcada, programação e automação industrial em um único sistema. O caráter aberto e documentado do projeto estimula sua replicação e evolução por outros estudantes e pesquisadores, ampliando seu impacto acadêmico e científico.

8 TRABALHOS FUTUROS

Este trabalho representa apenas o início da exploração do vasto campo da robótica. Ainda é possível adicionar muito mais funcionalidades, criar soluções e desenvolver pesquisas com esse robô. A seguir é exposto uma lista com algumas das possibilidades que podem ser elaboradas utilizando a plataforma como base:

- Desvio de obstáculos: Integração de sensores LIDAR e ultrassônicos para navegação autônoma com detecção de obstáculos. Com esses dispositivos, é possível fazer o robô seguir uma rota e desviar de obstáculos autonomamente.
- Seguidor de linha: adicionar sensores de cor na frente do robô e aprender sobre algoritmos de seguidores de linha.
- SLAM: vários tipos de SLAM existem, usando diferentes sensores, sendo que o robô já possui valores de codificadores nas rodas, e no LABIRAS existem *lidar*, câmera de profundidade, *webcam*, que podem ser adicionadas ao robô para realizar navegação.
- Base móvel com manipulador: é comum aplicações onde se usa em conjunto um robô móvel com um manipulador acoplado, como exemplo o robô da KUKA youBot. Graças a base impressa em 3D é factível analisar o acoplamento de um braço robótico.
- Múltiplos robôs: É possível realizar a montagem de outros robôs e fazer gerenciamento de frota desses dispositivos ou trabalhos em conjunto com eles.

REFERÊNCIAS

- ALBONICO, M.; ĐORĐEVIĆ, M.; HAMER, E.; MALAVOLTA, I. Software engineering research on the robot operating system: A systematic mapping study. **Journal of Systems and Software**, Elsevier, v. 197, p. 111574, 2023. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0164121222002503>. Acesso em: 10 abr. 2025.
- BABIUCH, M.; FOLTÝNEK, P.; SMUTNÝ, P. Using the esp32 microcontroller for data processing. In: IEEE. **2019 20th International Carpathian Control Conference (ICCC)**. 2019. p. 1–6. Disponível em: <https://ieeexplore.ieee.org/document/8765944>. Acesso em: 22 out. 2025.
- CONSIDINE, D. M.; CONSIDINE, G. D. **Standard handbook of industrial automation**. Springer Science & Business Media, 2012. Disponível em: <https://books.google.com/books?hl=pt-BR&lr=&id=vhXIBwAAQBAJ&oi=fnd&pg=PR11&dq=Standard+handbook+of+industrial+automation&ots=7ZOt1XMfmu&sig=6x1uODTTGEqzPJA0BaK-KpbGWIA>. Acesso em: 15 abr. 2025.
- DANTAS, M. A. **Aplicação de um robô aspirador como plataforma robótica do Instituto Federal do Piauí**. [S.l.], 2022. Disponível em: <http://bia.ifpi.edu.br:8080/jspui/handle/123456789/4975>. Acesso em: 14 nov. 2025.
- DICKERSON, S. L.; LAPIN, B. D. Control of an omni-directional robotic vehicle with mecanum wheels. In: IEEE. **NTC'91-National Telesystems Conference Proceedings**. 1991. p. 323–328. Disponível em: <https://doi.org/10.1109/NTC.1991.148039>. Acesso em: 09 mai. 2025.
- DOKIC, K. Microcontrollers on the edge—is esp32 with camera ready for machine learning? In: SPRINGER. **Image and Signal Processing: 9th International Conference, ICISP 2020, Marrakesh, Morocco, June 4–6, 2020, Proceedings 9**. 2020. p. 213–220. Disponível em: https://link.springer.com/chapter/10.1007/978-3-030-51935-3_23. Acesso em: 10 out. 2025.
- DUBEY, S.; PRATEEK, M.; SAXENA, M. *et al.* Robot locomotion—a review. **Int. J. Appl. Eng. Res**, v. 10, n. 3, p. 7357–7369, 2015. Disponível em: <https://eprints.whiterose.ac.uk/id/eprint/210874/>. Acesso em: 01 mai. 2025.
- FINOCCHIARO, F. **micro-ROS porting to ESP32**. 2020. Disponível em: <https://micro.ros.org/blog/2020/08/27/esp32/>. Acesso em: 16 nov. 2025.
- Folha Expressa. **Estudantes do Piauí se destacam em competição de inteligência artificial**. 2025. Disponível em: <https://folhaexpressa.com/educacao/estudantes-do-piaui-se-destacam-em-competicao-de-inteligencia-artificial/>. Acesso em: 16 nov. 2025.
- FOOTE, T. **Why ROS?** 2022. Disponível em: <https://www.ros.org/blog/why-ros/>. Acesso em: 03 jun. 2025.
- FRAGAPANE, G.; IVANOV, D.; PERON, M.; SGARBOSSA, F.; STRANDHAGEN, J. O. Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics. **Annals of operations research**, Springer, v. 308, n. 1, p. 125–143, 2022. Disponível em: <https://link.springer.com/article/10.1007/s10479-020-03526-7>. Acesso em: 12 abr. 2025.
- FRANCO, A. d. C. *et al.* Geração e controle de trajetória de robôs móveis omnidirecionais. Escola Politécnica/Instituto de Matemática, 2017. Disponível em: <https://repositorio.ufba.br/handle/ri/21580>. Acesso em: 19 abr. 2025.

FRUETT, F.; BARBOSA, F. P.; FRAGA, S. C. Z.; GUIMARÃES, P. I. A. Empowering steam activities with artificial intelligence and open hardware: The bitdoglab. **IEEE Transactions on Education**, IEEE, 2024. Disponível em: <https://ieeexplore.ieee.org/abstract/document/10500382>. Acesso em: 09 de nov. 2025.

Instituto Federal do Piauí (IFPI). **Estudantes do IFPI participam de evento AWS DeepRacer em Brasília**. 2024. Disponível em: <https://www.ifpi.edu.br/teresinacentral/noticias/estudantes-do-ifpi-participam-de-evento-aws-deepracer-em-brasilia>. Acesso em: 16 nov. 2025.

JOSEPH, L. **ROS Robotics projects**. Packt Publishing Ltd, 2017. Disponível em: https://books.google.com/books?hl=pt-BR&lr=&id=rLkrDwAAQBAJ&oi=fnd&pg=PP1&dq=ROS+Robotics+projects&ots=KxKZlgC0rJ&sig=yobA9PdRdryk92uJloCnx_O5lLg. Acesso em: 18 abr. 2025.

KOUBAA, A. *et al.* **Robot Operating System (ROS)**. Springer, 2017. v. 1. Disponível em: <https://link.springer.com/content/pdf/10.1007/978-3-319-26054-9.pdf>. Acesso em: 11 abr. 2025.

LALANCETTE, C.; POUBEL, L. **ROS Index**. 2022. Disponível em: <https://index.ros.org/stats/>. Acesso em: 23 jul. 2025.

LEAO, L.; AMARAL, I.; AZPÚRUA, H.; REZENDE, A.; PESSIN, G.; FREITAS, G. Exploração autônoma de ambientes planares utilizando um dispositivo robótico móvel. In: **Simpósio Brasileiro de Automação Inteligente-SBAI**. [s.n.], 2021. v. 1, n. 1. Disponível em: https://sba.org.br/open_journal_systems/index.php/sbai/article/view/2702. Acesso em: 08 abr. 2025.

LOPES, R. de A.; FORTES, A. F. C.; PEDROSA, A. Í. R.; ARAÚJO, A. G. F. de; ALVES, R. M. **PROJETO PEDAGÓGICO DO CURSO DE ENGENHARIA MECÂNICA NA MODALIDADE PRESENCIAL CAMPUS TERESINA CENTRAL**. 2018. Disponível em: <https://www.ifpi.edu.br/cursos/documentos-dos-cursos/ppc/PPCEMETSa.pdf>. Acesso em: 03 ago. 2025.

MACENSKI, S.; FOOTE, T.; GERKEY, B.; LALANCETTE, C.; WOODALL, W. Robot operating system 2: Design, architecture, and uses in the wild. **Science robotics**, American Association for the Advancement of Science, v. 7, n. 66, p. eabm6074, 2022. Disponível em: <https://www.science.org/doi/abs/10.1126/scirobotics.abm6074>. Acesso em: 23 abr. 2025.

MARSH, A. The proto-roomba - [past forward]. **IEEE Spectrum**, v. 57, n. 3, p. 50–50, 2020. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9014460/>. Acesso em: 22 mai. 2025.

MARTINS, D. A.; NETO, I. C. M.; FARIAS, R. F. Navegação, exploração de ambientes e manipulação de objetos através de slam visual utilizando uma plataforma robótica móvel integrada com um manipulador. **VI Seminário de Avaliação de Pesquisa Científica e Tecnológica (SAPCT) e V Workshop de Integração e Capacitação em Processamento de Alto Desempenho (ICPAD) - SENAI CIMATEC**, 2021. Disponível em: <https://doity.com.br/anais/sapcticpad2021/trabalho/186413>. Acesso em: 20 abr. 2025.

MARUYAMA, Y.; KATO, S.; AZUMI, T. Exploring the performance of ros2. In: . [s.n.], 2016. p. 1–10. Disponível em: <https://dl.acm.org/doi/abs/10.1145/2968478.2968502>. Acesso em: 08 mai. 2025.

MILLER, I. D.; CLADERA, F.; COWLEY, A.; SHIVAKUMAR, S. S.; LEE, E. S.; JARIN-LIPSCHITZ, L.; BHAT, A.; RODRIGUES, N.; ZHOU, A.; COHEN, A. *et al.* Mine tunnel exploration using multiple quadrupedal robots. **IEEE Robotics and Automation Letters**, IEEE, v. 5, n. 2, p. 2840–2847, 2020. Disponível em: <https://doi.org/10.1109/LRA.2020.2972872>. Acesso em: 09 mai. 2025.

NILOY, M. A.; SHAMA, A.; CHAKRABORTTY, R. K.; RYAN, M. J.; BADAL, F. R.; TASNEEM, Z.; AHAMED, M. H.; MOYEEN, S. I.; DAS, S. K.; ALI, M. F. *et al.* Critical design and control issues of indoor autonomous mobile robots: A review. **IEEE Access**, IEEE, v. 9, p. 35338–35370, 2021. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9363869/>. Acesso em: 22 mai. 2025.

OLIVEIRA, P. V. S. de; SANTOS, L. de F.; FERREIRA, M. P. Inteligência artificial na automação de processos industriais e seus impactos. **Revista de Economia Mackenzie**, v. 21, n. 1, p. 162–182, 2024. Disponível em: <https://editorarevistas.mackenzie.br/index.php/rem/article/view/16714>. Acesso em: 11 set. 2025.

Open Robotics. **Creating Your First ROS 2 Package**. 2025. <https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Creating-Your-First-ROS2-Package.html>. Disponível em: <https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Creating-Your-First-ROS2-Package.html>. Acesso em: 16 nov. 2025.

PARADA-SALADO, J. G.; ORTEGA-GARCÍA, L. E.; AYALA-RAMÍREZ, L. F.; PEREZ-PINAL, F. J.; HERRERA-RAMIREZ, C. A.; PADILLA-MEDINA, J. A. A low-cost land wheeled autonomous mini-robot for in-door surveillance. **IEEE Latin America Transactions**, IEEE, v. 16, n. 5, p. 1298–1305, 2018. Disponível em: <https://doi.org/10.1109/TLA.2018.8407100>. Acesso em: 19 abr. 2025.

PETRYSHYN, B.; POSTUPAIEV, S.; BARI, S. B.; OSTREIKA, A. Deep reinforcement learning for autonomous driving in amazon web services deepracer. **Information**, MDPI, v. 15, n. 2, p. 113, 2024. Disponível em: <https://www.mdpi.com/2078-2489/15/2/113>. Acesso em: 03 ago. 2025.

PYO, Y.; CHO, H.; JUNG, R.; LIM, T. **ROS Robot Programming**. 1st. ed. Seoul,: ROBOTIS Co.,Ltd., 2017.

QUIGLEY, M.; CONLEY, K.; GERKEY, B.; FAUST, J.; FOOTE, T.; LEIBS, J.; WHEELER, R.; NG, A. Y. *et al.* Ros: an open-source robot operating system. In: KOBE. **ICRA workshop on open source software**. 2009. v. 3, n. 3.2, p. 5. Disponível em: http://lars.mec.ua.pt/public/LAR%20Projects/BinPicking/2016_RodrigoSalgueiro/LIB/ROS/icraoss09-ROS.pdf. Acesso em: 03 abr. 2025.

QUIGLEY, M.; GERKEY, B.; SMART, W. D. **Programming Robots with ROS: a practical introduction to the Robot Operating System**. "O'Reilly Media, Inc.", 2015. Disponível em: <https://books.google.com/books?hl=pt-BR&lr=&id=Hnz5CgAAQBAJ&oi=fnd&pg=PR11&dq=Programming+Robots+with+ROS:+a+practical+introduction+to+the+Robot+Operating+System&ots=-7vgKWVCM3&sig=-231N9InMwQM2u-ScZBJThLehYM>. Acesso em: 13 abr. 2025.

RAMALHO, J. H. V. *et al.* Omnino: plataforma aberta de robô omnidirecional para enxame. Universidade Federal de Alagoas, 2020. Disponível em: <http://www.repositorio.ufal.br/jspui/handle/123456789/15089>. Acesso em: 06 out. 2025.

ROBOTICS, I. F. of. **World Robotics 2020 Report**. 2020. Disponível em: <http://reparti.free.fr/robotics2000.pdf>. Acesso em: 01 dez. 2025.

ROBOTICS, O. **ROS - Robot Operating System**. 2022. Disponível em: <https://www.ros.org/>. Acesso em: 19 jul. 2025.

RODRIGUES, F.; ELIESER, T. Aplicações da placa peltier para desenvolvimento de projetos de feiras de ciências. Universidade Federal do Pampa, 2023. Disponível em: <https://repositorio.unipampa.edu.br/items/a07e5376-e369-44ce-abd2-620300575d11>. Acesso em: 01 dez. 2025.

RODRIGUES, G. S. Internet das coisas (iot) aplicado no controle online de variação de velocidade de um motor de indução. Fundação Universidade Federal de Mato Grosso do Sul, 2024. Disponível em: <https://repositorio.ufms.br/retrieve/ca60cb77-e252-40d9-96f5-2efde4b24597/21331.pdf>. Acesso em: 12 nov. 2025.

ROMANO, V.; DUTRA, M. Introdução a robótica industrial. **Robótica Industrial: Aplicação na Indústria de Manufatura e de Processo**, São Paulo: Edgard Blücher, p. 1–19, 2002.

ROS micro. **micro-ROS puts ROS 2 onto microcontrollers**. 2022. Disponível em: <https://micro.ros.org/>. Acesso em: 14 nov. 2025.

RUBIO, F.; VALERO, F.; LLOPIS-ALBERT, C. A review of mobile robots: Concepts, methods, theoretical framework, and applications. **International Journal of Advanced Robotic Systems**, SAGE Publications Sage UK: London, England, v. 16, n. 2, p. 1729881419839596, 2019. Disponível em: <https://journals.sagepub.com/doi/abs/10.1177/1729881419839596>. Acesso em: 20 mai. 2025.

SA, L. N. D. **MODELAGEM E CONTROLE DE UM VEÍCULO OMNIDIRECIONAL**. Tese (ENGENHEIRO MECÂNICO) — PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO, Rio de Janeiro, Brazil, 2016. Disponível em: http://www.maxwell.vrac.puc-rio.br/Busca_etds.php?strSecao=resultado&nrSeq=30772@1. Acesso em: 16 de abr. 2025.

SCIAVICCO, L.; SICILIANO, B. **Robotica Industriale: modellistica e controllo di manipolatori**. [S.l.]: McGraw-Hill Libri Italia, 1995.

SHAMLIAN, S. **iRobot® Create® 3 Mechanical System**. 2022. Disponível em: https://iroboteducation.github.io/create3_docs/hw/mechanical/. Acesso em: 16 nov. 2025.

SIEGWART, R.; NOURBAKHSI, I. R.; SCARAMUZZA, D. **Introduction to autonomous mobile robots**. MIT press, 2011. Disponível em: https://books.google.com/books?hl=pt-BR&lr=&id=zys3AgAAQBAJ&oi=fnd&pg=PP1&dq=Introduction+to+autonomous+mobile+robots&ots=nfkRtZPE_B&sig=GmdJiKKTW7zwxMK0a_zji1vpf_c. Acesso em: 21 abr. 2025.

STANDARTIZATION, I. O. for. **ISO 8373:2021(en)**. 2012. Disponível em: <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-3:v1:en>. Acesso em: 01 nov. 2025.

TĂȚAR, M. O.; POPOVICI, C.; MÂNDRU, D.; ARDELEAN, I.; PLEȘA, A. Design and development of an autonomous omni-directional mobile robot with mecanum wheels. In: IEEE. **2014 IEEE International Conference on Automation, Quality and Testing, Robotics**. 2014. p. 1–6. Disponível em: <https://doi.org/10.1109/AQTR.2014.6857869>. Acesso em: 11 mai. 2025.

VIEIRA, D. F.; BARBOSA, F. P.; FRUETT, F.; SWART, J. W. School 4.0-hands on project based learning in stem and the inct namitec network. In: IEEE. **2024 IEEE Latin American Electron Devices Conference (LAEDC)**. 2024. p. 1–4. Disponível em: <https://ieeexplore.ieee.org/abstract/document/10638152>. Acesso em: 01 dez. 2025.

VILANOVA, C. d. S. F. **Aplicação de uma aeronave remotamente pilotada como plataforma robótica para práticas em laboratório da graduação em Engenharia Mecânica do Instituto Federal do Piauí.** [S.l.], 2022. Disponível em: <http://bia.ifpi.edu.br:8080/jspui/handle/123456789/4349>. Acesso em: 12 out. 2025.

VILCHES, V. M. **ROS Robotics Companies.** 2022. Disponível em: <https://github.com/vmayoral/ros-robotics-companies>. Acesso em: 14 nov. 2025.

WEON, E. S. **Understanding nodes.** 2022. Disponível em: <https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Nodes/Understanding-ROS2-Nodes.html>. Acesso em: 16 nov. 2025.

ZEIDIS, I.; ZIMMERMANN, K. **Dynamics of a four-wheeled mobile robot with Mecanum wheels.** Wiley Online Library, 2019. e201900173 p. Disponível em: <https://doi.org/10.1002/zamm.201900173>. Acesso em: 15 de mai. 2025.