

# Integração do ROS em uma plataforma robótica móvel de baixo custo para o ensino superior

Anderson Abner de S. Souza, Raul Benites Paradedá

<sup>1</sup>Departamento de Ciência da Computação  
Universidade do Estado do Rio Grande do Norte (UERN)  
Caixa Postal 59122-365 – Natal – RN – Brasil  
orcid{0000-0001-6353-8674, 0000-0002-4031-6275}

{andersonabner, raulparadedá}@uern.br

**Abstract.** *Mobile robotics has been widely used in education, including in higher education. However, often the robots used are expensive or the simulation environments do not fully reflect reality. We propose developing a low-cost hardware framework for mobile robotics education in higher education. We use the NodeMCU - ESP8266 microcontroller board and integrate it with the Robot Operating System (ROS). This integration allows for more efficient development and learning, with hardware abstraction and leveraging existing tools to facilitate sensor integration, actuator control, and data communication. To validate the framework, we conducted tests using sensor data from the constructed robot and implemented commonly explored algorithms in mobile robotics education.*

**Resumo.** *A robótica móvel tem sido amplamente utilizada no ensino superior. Em geral, os robôs utilizados são caros ou os ambientes simulados não refletem completamente a realidade. Neste artigo, propomos o desenvolvimento de um framework de hardware de baixo custo para o ensino de robótica móvel no ensino superior. Utilizamos a placa NodeMCU - ESP8266 e integramos com o ROS (Robot Operating System). Isso permite um desenvolvimento e aprendizado mais eficientes, com abstração do hardware e uso de ferramentas existentes para facilitar a integração com sensores, atuadores e comunicação de dados. Realizamos testes utilizando dados sensoriais do robô construído e executamos diferentes algoritmos comumente explorados no ensino de robótica móvel.*

## 1. Introdução

A robótica móvel tem ganhado destaque nas últimas décadas, impulsionada pelo rápido avanço tecnológico e pela ampla gama de aplicações dos robôs móveis na sociedade. Esses robôs são utilizados em diversas áreas, como limpeza (robôs aspiradores de pó) [Gross 2020], exploração espacial [Bayer and Faigl 2019], exploração de ambientes hostis [Feng et al. 2021], guias em exposições [Gasteiger et al. 2021], suporte a serviços médicos [Wang and Wang 2021], transporte de materiais [Bačík et al. 2020], etc.

Dada a versatilidade da robótica móvel, é natural considerar seu uso no ambiente acadêmico. Estudos anteriores, como o de [Anwar et al. 2019], mencionam cerca de 147 pesquisas que utilizaram agentes robóticos na educação básica entre 2000 e 2018. Outro trabalho, realizado por [Papadopoulos et al. 2020], identificou 21 estudos no período de janeiro a abril de 2018 que empregaram robôs assistentes para auxiliar alunos em fase

pré-universitária no aprendizado de ciências e matemática. No entanto, poucos trabalhos exploram o uso dessa tecnologia no ensino superior. Independentemente do nível de ensino, a introdução da robótica pode trazer benefícios no processo de aprendizagem, como estímulo ao pensamento crítico [Merlo-Espino et al. 2018], resolução de problemas [Eteokleous 2018] e outras habilidades. Além disso, esses estudos normalmente envolvem agentes robóticos sociais, que interagem e se comunicam com humanos seguindo comportamentos e regras sociais. Esse tipo de agente requer recursos como captura de imagem, reconhecimento facial, habilidade de fala e compreensão, simulação de emoções, entre outros. Requisitos que podem ser de difícil obtenção para muitas instituições de ensino e pesquisadores, além de exigir programação complexa.

Portanto, é importante explorar a robótica móvel no ensino superior, dentro de cursos como Ciência da Computação, Engenharia da Computação, Engenharia Mecatrônica, Engenharia de Automação, entre outros. Além disso, é fundamental pesquisar estratégias de robótica que sejam de fácil aquisição e desenvolvimento.

Normalmente, no ensino superior, a robótica móvel é abordada em disciplinas teórico-práticas, nas quais os alunos consolidam seus conhecimentos teóricos programando robôs reais e/ou simulando em ambientes virtuais. A primeira opção depende de recursos financeiros provenientes de projetos de pesquisa ou parcerias externas, já que as universidades geralmente não têm recursos suficientes para adquirir vários robôs, que podem ser caros. A segunda opção depende da qualidade, disponibilidade e portabilidade dos simuladores, que não reproduzem completamente as características de um experimento no ambiente real.

Portanto, plataformas robóticas de baixo custo são adequadas para o ensino de robótica [Araújo et al. 2015]. Essas plataformas podem ser construídas em grande quantidade, utilizando placas microcontroladas populares, simples, de código aberto, que permitem fácil programação, integração com sensores, modularidade e possuem vasta documentação, como é o caso das placas Arduino [Araújo et al. 2015]. Essas soluções são atraentes para a construção de robôs móveis de baixo custo para fins educacionais [Warren et al. 2011]. No entanto, as placas microcontroladas possuem limitações em termos de poder computacional e capacidade de processamento para algoritmos mais complexos. Algoritmos de planejamento de trajetórias, odometria visual e SLAM (*Simultaneous Localization and Mapping*) são exemplos de processos que podem exigir processadores robustos para sua execução, tornando inviável o uso de microcontroladores mais simples, como o ATmega2560 de alguns Arduinos ou o ESP32.

Uma abordagem promissora para o desenvolvimento de programas de controle e algoritmos de navegação inteligente é o uso do ROS (*Robot Operating System*). O ROS fornece um *framework* unificado e de código aberto para o desenvolvimento de programas de controle de robôs, tanto em ambientes reais quanto simulados [R GOEBEL 2012]. Isso possibilita o reuso de código, abstração de hardware e troca de mensagens entre processos.

Este trabalho tem como objetivo apresentar a construção de um robô móvel de baixo custo com rodas e sua integração com o ROS. A plataforma robótica é baseada na placa microcontrolada NodeMCU - ESP8266, que permite o controle de atuadores, leitura de sensores e comunicação via Wi-Fi com um computador servidor executando o ROS. A integração dessa plataforma com o ROS por meio da comunicação Wi-Fi oferece

eficiência no tempo de desenvolvimento e aprendizado, aproveitando ferramentas já desenvolvidas e facilitando a integração com sensores, atuadores e comunicação de dados [Araújo et al. 2015]. Isso promove a aprendizagem ao permitir atividades práticas com um robô real, considerando suas capacidades, restrições e limitações, por meio do desenvolvimento de software utilizando um *framework* popular na comunidade de robótica, tanto no meio acadêmico quanto no mercado tecnológico. Dentre as características deste projeto, destacam-se a acessibilidade na aquisição dos componentes, modularidade, facilidade de montagem, integração com o ROS, *open-hardware* e *open-software*. Validamos esse *framework* por meio de testes utilizando dados sensoriais do robô construído e a execução de diferentes algoritmos comumente explorados no ensino de robótica móvel.

## 2. Trabalhos relacionados

Nesta seção, apresentamos estudos relacionados ao ensino da robótica, com foco na integração do ROS com robôs reais. Também são mencionados trabalhos educacionais que exploram aspectos da robótica, mesmo sem utilizar o ROS.

Em [Raudmäe et al. 2023] uma plataforma robótica de código aberto para fins educacionais e de pesquisa, que inclui suporte ao ROS é proposta. O robô desenvolvido pelos autores possui recursos de movimentação omnidirecional com 3 motores, um computador embarcado, uma placa microcontrolada STM32 NUCLEO-L476RG, uma câmera RGB\_D e sensores de rotação (*encoders*) em cada roda. O robô é capaz de executar o ambiente ROS em seu próprio computador embarcado, permitindo a programação e a comunicação com sensores e atuadores. Embora ofereça diversas possibilidades de aplicação, desde funcionalidades básicas como localização por odometria até tarefas mais complexas como SLAM visual, a plataforma tem um custo consideravelmente alto para a realidade brasileira, com um preço em torno de 2000 euros<sup>1</sup>.

Em outro trabalho [Miranda 2022], é apresentada a concepção de um robô de baixo custo, que também é integrado ao ROS. O robô possui um sistema de acionamento diferencial, motores com *encoders* e é baseado em Arduino e Raspberry Pi. Conta com um sensor LiDAR, câmera e outros equipamentos. O custo final do robô foi de 428,59 euros<sup>2</sup>. Embora seja mais acessível em termos de preço, quando multiplicado por várias unidades para atender a uma turma, pode não ser tão vantajoso. Além disso, a complexidade da construção e manipulação dessa plataforma, devido à quantidade de dispositivos eletrônicos envolvidos, pode representar um desafio, especialmente para alunos menos experientes com hardware.

Em outra proposta [Karalekas et al. 2020], é apresentado um robô integrado ao ROS com o objetivo de ser utilizado no ensino médio em disciplinas do eixo STEM (Ciência, Tecnologia, Engenharia e Matemática), podendo ser adaptado para o ensino superior. O robô é baseado na placa Raspberry 3B+, motores diferenciais com *encoders*, sonar, LiDAR e um braço manipulador. Os autores realizaram uma análise de outras plataformas, concluindo que o Duckiebot, entre as opções com integração ao ROS, é o de menor custo, com preço em torno de 299 euros<sup>3</sup>. Embora essa proposta se assemelhe à ideia apresentada neste trabalho, o presente trabalho busca uma abordagem de custo mais

---

<sup>1</sup>Aproximadamente R\$ 10.540,00, com a taxa de câmbio na data da redação deste trabalho, de R\$ 5,27.

<sup>2</sup>Aproximadamente R\$ 2.258,67.

<sup>3</sup>Cerca de R\$ 1.575,73. <https://get.duckietown.com>, acessado em: 21/06/23.

baixo e montagem mais fácil.

Um dos kits robóticos educacionais mais populares é o Lego Mindstorms NXT<sup>4</sup>. Ele é adequado para iniciantes em robótica e pode ser utilizado desde os níveis mais básicos da educação até disciplinas iniciais de robótica no ensino superior. O robô é equipado com motores diferenciais com *encoders* e uma variedade de sensores acessíveis, como acelerômetro, de luz, de som, ultrassom e sensores de toque. Além disso, possui uma interface de controle baseada em ROS, utilizando o pacote `nxt_ros` (nota 4). O preço aproximado dessa plataforma é de US\$ 858,50<sup>5</sup>.

Pode-se citar outras plataformas robóticas que podem ser usadas com propósitos educacionais e aplicadas ao ensino superior, porém, tais propostas não possuem a integração com o ROS e, em sua maioria, são mais custosas que a plataforma proposta. Esse é o caso do E-puc2 [Mondada et al. 2009], OpenBot [Müller and Koltun 2021], Khepera [Mondada et al. 1999], Robobo [Bellás et al. 2018], entre outros.

### 3. Fundamentação Teórica

Nesta seção, apresentamos pontos fundamentais para a compreensão deste trabalho. Abordamos as limitações das placas microcontroladas, a função do ROS e a utilização da placa NodeMCU - ESP8266 na robótica de baixo custo e sua integração com o ROS.

#### 3.1. Limitações das placas microcontroladas

As placas microcontroladas, como os Arduinos ou ESP32, são amplamente utilizadas na robótica de baixo custo devido à sua popularidade, simplicidade e facilidade de programação [Araújo et al. 2015]. No entanto, elas possuem limitações importantes a serem consideradas.

Em termos de poder computacional e capacidade de processamento de dados, as placas microcontroladas podem apresentar restrições. Algoritmos mais complexos, como planejamento de caminhos, odometria visual e SLAM, podem exigir um processamento mais robusto do que essas placas oferecem. Além disso, a capacidade de expansão e integração de sensores e atuadores pode ser limitada, devido ao número de pinos I/O disponíveis nas placas microcontroladas. A falta de recursos avançados de comunicação também pode restringir a interação do robô com outros dispositivos ou sistemas.

Ao utilizar placas microcontroladas na construção de robôs móveis, especialmente em aplicações complexas, é importante considerar essas limitações. Em alguns casos, pode ser necessário buscar alternativas mais poderosas, como placas de desenvolvimento baseadas em microprocessadores ou sistemas embarcados mais robustos.

#### 3.2. Robot Operating System (ROS)

O ROS é um *framework* de programação amplamente utilizado na robótica, que oferece uma abordagem modular e flexível para o desenvolvimento de software de controle de robôs. O ROS fornece uma arquitetura baseada em troca de mensagens entre nós independentes, permitindo a comunicação e colaboração entre diferentes componentes do sistema robótico [Quigley et al. 2009]. Ele também oferece um conjunto abrangente

---

<sup>4</sup><https://education.lego.com/pt-br/>, acessado em: 21/06/23.

<sup>5</sup>Aproximadamente R\$ 4.223,82 com a taxa cambial de R\$ 4,92.

de bibliotecas, ferramentas e pacotes que facilitam o desenvolvimento de software para robôs [Vaughan and Gerkey 2007]. O ROS é altamente integrável com sensores, atuadores e plataformas robóticas, permitindo o acesso e controle fácil desses componentes [Quigley et al. 2009]. No contexto educacional, o ROS desempenha um papel importante na formação de estudantes e pesquisadores em robótica, fornecendo uma plataforma padronizada de desenvolvimento e recursos de aprendizado. Embora o uso do ROS apresente desafios em termos de requisitos computacionais e configuração, sua flexibilidade, modularidade e comunidade ativa de usuários o tornam uma escolha popular e poderosa para o desenvolvimento de software robótico.

### **3.3. Placa microcontrolada NodeMCU - ESP8266**

A placa microcontrolada NodeMCU - ESP8266 é uma opção viável e econômica para a construção de robôs móveis no contexto educacional e de pesquisa [Araújo et al. 2015]. Ela oferece recursos como controle de atuadores, leitura de sensores e comunicação via Wi-Fi, tornando-se amplamente utilizada nesse campo. A integração da placa com o ROS proporciona uma abordagem eficiente para o desenvolvimento e aprendizado, facilitando a integração com sensores, atuadores e comunicação de dados. Com vantagens como baixo custo, arquitetura de hardware *open-source* e programação intuitiva, a NodeMCU - ESP8266 permite aos estudantes adquirirem habilidades práticas em programação de robôs de forma acessível. Além disso, a comunicação direta com o ROS e o aproveitamento de bibliotecas e ferramentas disponíveis ampliam as possibilidades de projetos robóticos, permitindo a exploração de conceitos avançados de forma eficiente. Assim, essa placa se destaca como uma opção versátil e econômica para a construção de robôs móveis no ambiente educacional e de pesquisa.

## **4. Materiais e Métodos**

### **4.1. Visão geral da plataforma robótica**

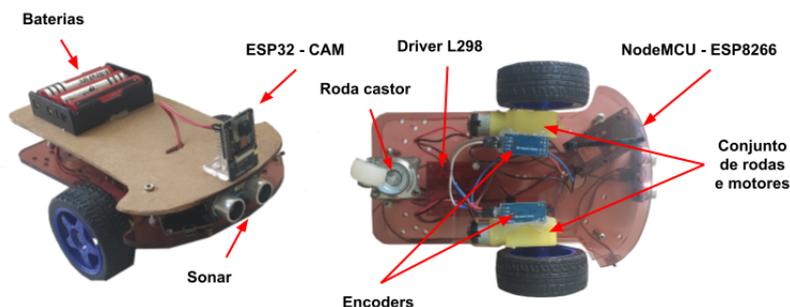
Como ilustrado na Figura 1, a plataforma robótica proposta neste trabalho consiste em um robô de duas rodas com *drive* diferencial. O robô é equipado com um *motor driver* para acionamento dos motores, sensor de ultrassom, câmera (ESP32-CAM) e a placa microcontroladora NodeMCU - ESP8266. Além disso, são utilizadas baterias para alimentar os componentes eletrônicos e mecânicos.

Essa proposta segue a filosofia de ser *open-hardware* e *open-software*, permitindo que alunos e professores possam replicá-la e aprimorá-la. A construção do robô é simples e modular, possibilitando a adição de outros dispositivos sensoriais. A integração com o ROS proporciona interoperabilidade e extensibilidade. Os materiais utilizados são de baixo custo e facilmente encontrados no mercado nacional ou internacional. Além disso, alguns componentes podem ser impressos em 3D ou construídos com madeira MDF.

Com tais características, é possível empregar o robô no ensino da robótica móvel, desde a apresentação de conceitos mais iniciais de movimentação, cinemática e percepção à conceitos mais avançados como odometria visual, navegação e SLAM.

### **4.2. Hardware**

O robô é construído sobre uma base de acrílico e outra de madeira MDF, ambas com o mesmo formato. Essas bases fornecem suporte para os componentes mecânicos e



**Figura 1. Estrutura básica do robô proposto.**

eletrônicos do robô. Dois motores diferenciais com rodas e *encoders* ópticos são responsáveis pela locomoção do robô, e uma roda boba (*castor wheels*) usada para equilibrar a estrutura mecânica. Esse conjunto permite que o robô atinja velocidades de até 1,5 m/s. Para acionar e controlar a velocidade dos motores, é utilizada uma ponte H L298.

Na parte intermediária da plataforma, estão o sensor de ultrassom e a placa microcontrolada NodeMCU - ESP8266. O sonar (HC-SR04) mede a distância entre o robô e obstáculos no intervalo de 2 cm a 400 cm, podendo ser utilizado para tarefas de mapeamento e desvio de obstáculos. A placa microcontrolada permite a leitura dos *encoders* e do sonar, realiza o controle de velocidade dos motores por PWM e permite a comunicação da plataforma com um computador externo por meio de troca de mensagens ROS.

No topo do chassi, está instalada uma câmera RGB, ESP32-CAM, que adquire imagens para posterior processamento no computador externo. Além disso, são utilizadas baterias para alimentar o conjunto eletromecânico do robô. São utilizadas três baterias recarregáveis do tipo 18650 para alimentar os motores, e um *power bank* para alimentar a placa microcontrolada NodeMCU - ESP8266 e os sensores.

A Tabela 1 apresenta os custos aproximados dos equipamentos adquiridos para a montagem da plataforma, tanto no mercado nacional quanto no internacional.

### 4.3. Software

A plataforma proposta utiliza o ROS para comunicação e controle. O ROS permite a comunicação entre placas microcontroladas, como Arduino e ESP, por meio da biblioteca *rosserial*. Essa biblioteca adiciona funcionalidades ao *firmware* da placa NodeMCU - ESP8266, permitindo emular a linguagem ROS diretamente [Araújo et al. 2015]. Essa abordagem permite a comunicação entre um computador externo e a placa NodeMCU - ESP8266 por meio de uma conexão serial ou Wi-Fi.

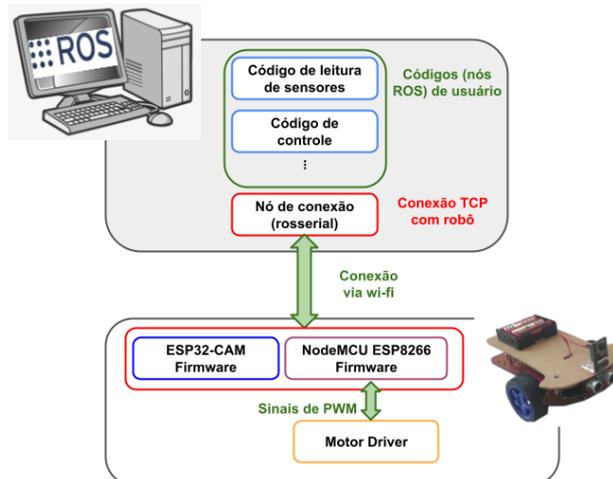
No computador externo, é necessário ter os seguintes componentes em execução: 1) o ROS Master, responsável pela execução e comunicação dos nós do ROS; 2) o nó de comunicação entre o computador e a placa NodeMCU - ESP8266; e 3) o nó desenvolvido pelo usuário, responsável pela leitura de mensagens dos sensores enviadas pelo robô, envio de comandos de controle ao robô e visualização dos dados. A Figura 2 ilustra essa arquitetura.

No robô, a placa NodeMCU - ESP8266 executa o *firmware*, que inclui o código

**Tabela 1. Tabela de custo em reais e dólares (taxa cambial de R\$ 4,92) e a quantidade utilizada de cada componente.**

Item	Preço unitário Brasil (R\$)	Preço unitário exterior (US\$)	Qtd	Preço total Brasil (R\$)	Preço total exterior (U\$)
Fios	0,38	0,08	40	15,20	1,22
NodeMCU	45,00	4,55	1	45,00	4,55
ESP32-CAM	51,99	6,50	1	51,99	6,50
Motor + roda	22,00	2,36	2	44,00	4,72
Encoder	10,50	0,92	2	21,00	1,84
Módulo driver	28,90	2,36	1	28,90	4,92
Sonar	12,50	0,68	1	12,50	0,68
Parafusos, porcas e arruelas	0,70	0,14	10	7,00	1,40
Baterias	21,00	3,86	3	63,00	13,58
Case de bateria	21,00	0,15	1	21,00	0,15
Total				309,59	39,56

para a leitura dos sensores (*encoders* e sonar), envio de comandos para os atuadores (motores DC), criação da comunicação com o ROS Master e criação dos tópicos para o envio dos dados dos sensores e a leitura dos comandos de movimentação provenientes do usuário no computador externo. A câmera ESP32-CAM, também presente no robô, deve ser configurada e inicializada para se conectar à rede Wi-Fi, a fim de fornecer imagens que podem ser lidas pelos nós em execução no computador externo, por meio do endereço de servidor HTTP fornecido pela própria ESP32-CAM.



**Figura 2. Arquitetura de módulos de software.**

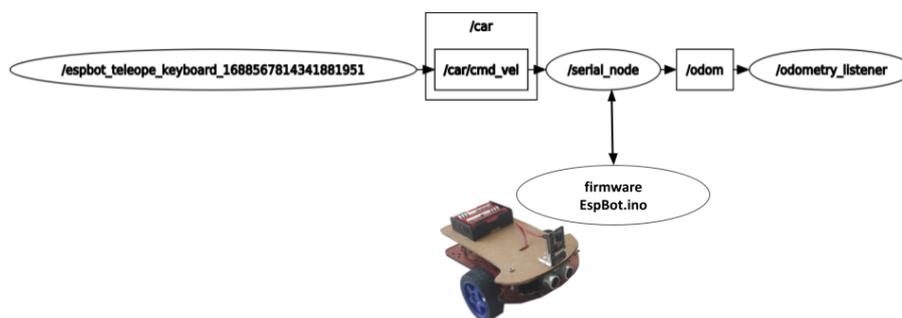
## 5. Experimentos

A fim de validar o *framework* proposto e sua integração com o ROS, foram realizados experimentos com base em conceitos de robótica móvel, como cinemática com odometria e mapeamento de ambientes com sensores.

## 5.1. Cinemática com odometria

No experimento de cinemática, foi estabelecido um trajeto quadrado de  $0,62 \times 0,62\text{m}$  que o robô deveria percorrer. O objetivo era analisar os dados gerados pelo modelo de odometria implementado no *firmware* do robô, que calcula sua pose (coordenadas  $x$  e  $y$  em  $m$  e orientação em  $rad$ ) com base nas informações dos *encoders* das rodas.

O robô foi inicializado juntamente com o ambiente ROS, e os nós necessários para o experimento foram executados: `serial_node`, `espbote_teleop_keyboard` e `odometry_listener`. Também foram criados os tópicos `/car/cmd_vel` e `/odom`, conforme ilustrado na Figura 3, gerada pela ferramenta `rqt_graph` do ROS e adaptada.



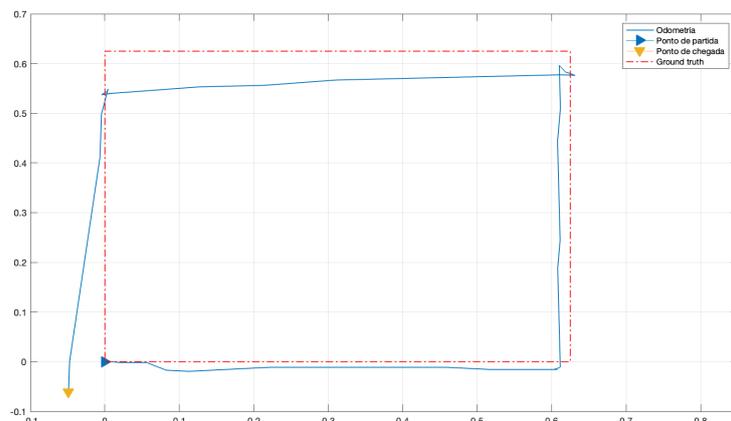
**Figura 3. Nós e tópicos ROS criados para habilitar a execução do cálculo de odometria do trajeto especificado.**

Em suma, o robô se conecta à mesma rede Wi-Fi do computador *off-board* que executa o ROS. No computador, o ROS master é inicializado (`roscore`) e em seguida o nó `/serial_node` é inicializado para permitir a comunicação Wi-Fi entre o NodeMCU-ESP8266 e o computador. Este nó também lê os dados do robô e cria tópicos relacionados a cada informação, como odometria (`/odom`), bem como tópicos para comandos a serem enviados e executados pelo robô, como velocidades (`/car/cmd_vel`).

Em seguida, o nó de teleoperação do robô (`/espbote_teleop_keyboard`) e o nó de leitura e exibição das informações de odometria (`/odometry_listener`) são inicializados. Os dados da odometria são armazenados e plotados. A Figura 4 mostra os resultados obtidos pela odometria do robô durante a execução do trajeto quadrado, com o eixo horizontal representando o eixo  $x$  e o eixo vertical representando o eixo  $y$ , ambos em metros. Os resultados demonstraram que, para este trajeto, a odometria do robô calculou uma posição final aproximada de  $x = -0,05\text{m}$  e  $y = -0,06\text{m}$ , com um erro absoluto final de aproximadamente  $0,078\text{m}$ . Essa medida é considerada aceitável para um robô de baixo custo, com *encoders* de baixa resolução, conforme observado em outras propostas semelhantes [Karalekas et al. 2020].

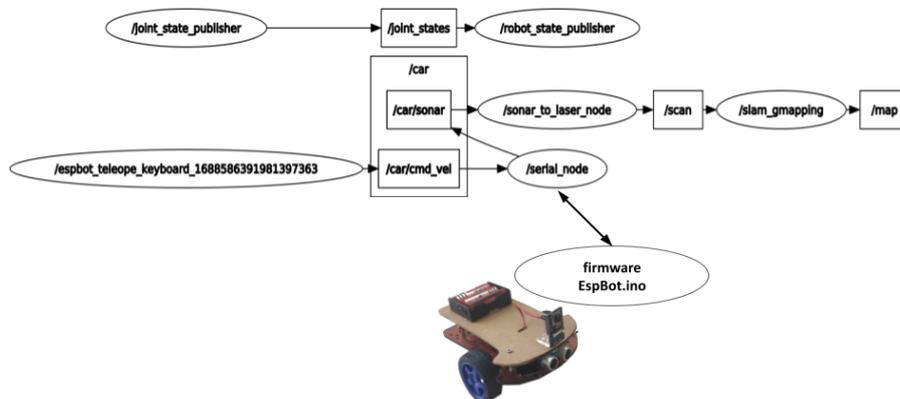
## 5.2. Mapeamento

Para o experimento de mapeamento, foi utilizado o pacote *gmapping* do ROS, que permite a construção de mapas em grades de ocupação a partir de leituras sensoriais de um sensor de distância, como o LiDAR, combinadas com odometria [Grisetti et al. 2007]. O *gmapping* utiliza um filtro de partículas para estimar a pose otimizada do robô e construir mapas consistentes.



**Figura 4. Trajeto a ser realizado (*ground truth*) e trajeto calculado pelo modelo cinemático de odometria.**

A Figura 5 mostra os nós e tópicos criados para o experimento, utilizando a ferramenta `rqt_graph`. Para adaptar o *gmapping* ao uso de sonar como sensor de distância, foi implementado o nó `/sonar_to_laser_node`, responsável por converter as informações do sonar (tópico `/car/sonar`) para o formato de leituras de laser (tópico `/scan`). Esse nó respeitou as características intrínsecas do sonar utilizado no robô, como a distância máxima medida e o campo de visão. Com os dados disponíveis, o algoritmo de mapeamento pôde construir o mapa em grade de ocupação, que foi publicado no tópico `/map` e visualizado pelo `rviz`, a ferramenta de visualização do ROS.



**Figura 5. Nós e tópicos ROS criados para habilitar a execução do algoritmo de mapeamento *gmapping*.**

Para visualizar o mapeamento no `rviz`, foi criado um modelo virtual 3D do robô compatível com a ferramenta. Isso permitiu a visualização *online* do processo de mapeamento. As Figuras 6 e 7 mostram o ambiente mapeado e o mapa resultante, respectivamente. Os principais obstáculos são destacados nas imagens, tanto no ambiente mapeado quanto no mapa, utilizando cores específicas. O mapa foi construído com células quadradas de 0,05m de lado. As células em preto representam obstáculos mapeados, as brancas representam áreas livres de obstáculos e as cinza indicam áreas ainda não mapeadas.



Figura 6. Ambiente mapeado.

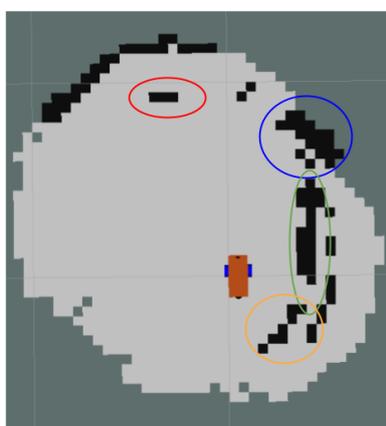


Figura 7. Mapa construído com o *gmapping* e exibido pelo *rviz*.

## 6. Conclusão

Neste trabalho, foi desenvolvido um *framework* de hardware de um robô de baixo custo e integrado ao ROS, visando sua utilização em disciplinas de robótica móvel no ensino superior. A integração com o ROS permitiu a abstração do hardware e o uso de ferramentas já desenvolvidas, como o pacote *gmapping*, facilitando a integração de sensores, atuadores e comunicação de dados. Essa plataforma robótica pode ser uma ferramenta útil no ensino da robótica, especialmente em instituições com restrições financeiras.

Foram realizados testes com algoritmos comumente utilizados na robótica móvel, como cinemática com odometria e mapeamento de ambientes. Apesar das limitações dos dispositivos, os resultados foram satisfatórios, demonstrando a funcionalidade da teleoperação, localização por odometria e mapeamento com sonar. Além disso, estão em andamento testes com algoritmos de câmera e integração sensorial, e há a possibilidade de montagem de um sistema em que o sonar possa ser direcionado para diferentes orientações, economizando energia ao evitar o movimento constante do robô.

É importante destacar que, embora tenha sido desenvolvido para o ensino superior, essa plataforma robótica pode ser adaptada para outros níveis de ensino.

## Referências

- Anwar, S., Bascou, N. A., Menekse, M., and Kardgar, A. (2019). A systematic review of studies on educational robotics. *Journal of Pre-College Engineering Education Research (J-PEER)*, 9(2):2.
- Araújo, A., Portugal, D., Couceiro, M. S., and Rocha, R. P. (2015). Integrating arduino-based educational mobile robots in ros. *Journal of Intelligent & Robotic Systems*, 77:281–298.
- Bačák, J., Tkáč, P., Hric, L., Alexovič, S., Kyslan, K., Olexa, R., and Perduková, D. (2020). Phollower—the universal autonomous mobile robot for industry and civil environments with covid-19 germicide add-on meeting safety requirements. *Applied Sciences*, 10(21):7682.
- Bayer, J. and Faigl, J. (2019). On autonomous spatial exploration with small hexapod walking robot using tracking camera intel realsense t265. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE.
- Bellas, F., Naya, M., Varela, G., Llamas, L., Bautista, M., Prieto, A., and Duro, R. J. (2018). Robobo: the next generation of educational robot. In *ROBOT 2017: Third Iberian Robotics Conference: Volume 2*, pages 359–369. Springer.
- Eteokleous, N. (2018). Employing educational robotics for the development of problem-based learning skills. In *Encyclopedia of Information Science and Technology, Fourth Edition*, pages 2492–2502. IGI Global.
- Feng, S., Shi, H., Huang, L., Shen, S., Yu, S., Peng, H., and Wu, C. (2021). Unknown hostile environment-oriented autonomous wsn deployment using a mobile robot. *Journal of Network and Computer Applications*, 182:103053.
- Gasteiger, N., Hellou, M., and Ahn, H. S. (2021). Deploying social robots in museum settings: A quasi-systematic review exploring purpose and acceptability. *International Journal of Advanced Robotic Systems*, 18(6):17298814211066740.
- Grisetti, G., Stachniss, C., and Burgard, W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23:34–46.
- Gross, J. (2020). Interviewing roomba: A posthuman study of humans and robot vacuum cleaners. *Explorations in Media Ecology*, 19(3):285–297.
- Karalekas, G., Vologiannidis, S., and Kalomiros, J. (2020). Europa: A case study for teaching sensors, data acquisition and robotics via a ros-based educational robot. *Sensors*, 20(9):2469.
- Merlo-Espino, R., Villareal-Rodríguez, M., Morita-Aleander, A., Rodríguez-Reséndiz, J., Pérez-Soto, G., and Camarillo-Gómez, K. (2018). Educational robotics and its impact in the development of critical thinking in higher education students. In *2018 XX Congreso Mexicano de Robótica (COMRob)*, pages 1–4. IEEE.
- Miranda, A. C. (2022). Teaching ros2 with a minimalistic mobile robot.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous*

- robot systems and competitions*, volume 1, pages 59–65. IPCB: Instituto Politécnico de Castelo Branco.
- Mondada, F., Franzi, E., and Guignard, A. (1999). The development of khepera. In *Experiments with the Mini-Robot Khepera, Proceedings of the First International Khepera Workshop*, number CONF, pages 7–14.
- Müller, M. and Koltun, V. (2021). Openbot: Turning smartphones into robots. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*.
- Papadopoulos, I., Lazzarino, R., Miah, S., Weaver, T., Thomas, B., and Koulouglioti, C. (2020). A systematic review of the literature regarding socially assistive robots in pre-tertiary education. *Computers & Education*, 155:103924.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., et al. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan.
- R GOEBEL, P. (2012). Ros by example a do-it-yourself guide to the robot operating system volume 1.
- Raudmäe, R., Schumann, S., Vunder, V., Oidekivi, M., Nigol, M. K., Valner, R., Masnavi, H., Singh, A. K., Aabloo, A., and Kruusamäe, K. (2023). Robotont–open-source and ros-supported omnidirectional mobile robot for education and research. *HardwareX*, page e00436.
- Vaughan, R. T. and Gerkey, B. P. (2007). Reusable robot software and the player/stage project. *Software Engineering for Experimental Robotics*, pages 267–289.
- Wang, X. V. and Wang, L. (2021). A literature survey of the robotic technologies during the covid-19 pandemic. *Journal of Manufacturing Systems*, 60:823–836.
- Warren, J.-D., Adams, J., Molle, H., Warren, J.-D., Adams, J., and Molle, H. (2011). *Arduino for robotics*. Springer.