



CONCEPÇÃO DE UM ROBÔ *OPEN SOURCE* MÓVEL VOLTADO PARA PESQUISAS EM ROBÓTICA COLABORATIVA

Natanael Magalhães Lima de Sousa*¹, Otávio Augusto Rocha da Cruz², Daniel Costa Ramos¹

¹FEELT – Universidade Federal de Uberlândia – Campus Patos de Minas

²PPGEE - Programa de Pós-Graduação em Engenharia Elétrica - UFRGS

Resumo - A robótica móvel é uma área em constante desenvolvimento, com diversas pesquisas em curso. No entanto, realizar estudos práticos muitas vezes requer a montagem ou compra de um robô, o que pode ser complexo, demorado e dispendioso. Para abordar esses problemas, o projeto se propôs a preencher essas lacunas com a criação do SciCoBot (Science, Coffee e Robot), uma plataforma robótica móvel de baixo custo, aberta para colaboração, modular e expansível. Essa plataforma é baseada em tecnologias como ROS 2, micro-ROS, Arduino Due e Raspberry Pi. Para atingir esse objetivo, foi criada uma versão da plataforma equipada com sensores e implementada uma arquitetura de código modular para o Arduino e o Raspberry Pi, além de um guia ilustrado para iniciantes com o objetivo de facilitar a compreensão, e consequentemente a replicação. Todos os detalhes foram documentados e disponibilizados no GitHub para validar a versão construída. Ao comparar a proposta com outras estruturas disponíveis, destacou-se sua adequação para uso específico na Universidade Federal de Uberlândia, campus Patos de Minas, pois foi concebido como uma ferramenta de pesquisa especialmente para essa instituição.

Palavras-Chave – Arduino, Raspberry Pi, Robótica móvel, ROS2, Scicobot.

CONCEPTION OF AN OPEN SOURCE MOBILE ROBOT FOCUSED ON RESEARCH IN COLLABORATIVE ROBOTICS

Abstract - Mobile robotics is an area in constant development, with several ongoing researches. However, carrying out practical studies often requires the assembly or purchase of a robot, which can be complex, time-consuming and expensive. To address these problems, the project set out to fill these gaps with the creation of SciCoBot (Science, Coffee and Robot), a low-cost, open-collaboration, modular, and expandable mobile robotics platform. This platform is based on technologies such as ROS 2, micro-ROS, Arduino Due and Raspberry Pi. To achieve this objective, a version of the platform equipped with sensors was created and a modular code architecture

was implemented for the Arduino and the Raspberry Pi, in addition to an illustrated guide for beginners in order to facilitate understanding, and consequently replication. All details were documented and made available on GitHub to validate the built version. When comparing the proposal with other available structures, its suitability for specific use at the Federal University of Uberlândia, campus Patos de Minas, was highlighted, as it was designed as a research tool especially for that institution.

Keywords - Arduino, Mobile Robotics, Raspberry Pi, ROS2, Scicobot.

I. INTRODUÇÃO

A robótica é o campo que se concentra na construção e no design de robôs. Seu avanço persiste em ritmo acelerado, trazendo de forma contínua novas descobertas, tecnologias e aplicações. Essa área desempenha uma função fundamental em nossa sociedade contemporânea, aprimorando a eficiência, segurança e bem-estar em diversas áreas e setores de estudo. Com suas constantes inovações, a robótica segue moldando o futuro de maneira revolucionária. O Brasil ainda tem uma participação limitada nesse desenvolvimento. De acordo com a Federação Internacional de Robótica, estima-se que o Brasil possua aproximadamente 10 robôs para cada 10.000 trabalhadores. Além disso, o número de robôs instalados no Brasil representa apenas 0,6% do total mundial, sendo que a indústria automotiva sozinha responde por mais de 50% dos robôs utilizados no país [1].

O constante avanço da robótica ao longo dos anos tem impulsionado a busca por dispositivos altamente autônomos, capazes de tomar decisões e não dependerem de orientação humana em suas ações. Para resolver esse desafio, surgiram os robôs móveis, que podem se movimentar no ar, na água ou na terra. Esses robôs móveis fornecem uma plataforma valiosa e flexível para conduzir pesquisas em robótica.

Por meio de experimentação e desenvolvimento de algoritmos e tecnologias, os pesquisadores avançam, buscando inovações que contribuam para a evolução da ciência e da tecnologia robótica, abrindo caminho para novas aplicações em diversos setores da sociedade. A robótica voltada para a pesquisa oferece a capacidade de validar vários

algoritmos e soluções, já que uma implementação desde o início é inviável devido a todas as complexidades envolvidas.

A Tabela 1 apresenta algumas das plataformas desenvolvidas para fins educacionais e/ou de pesquisa, destacando que os custos variam e o desenvolvimento de novas plataformas está em constante progresso. É notável que os valores na tabela apresentam uma ampla variação, abrangendo uma faixa que vai de aproximadamente R\$1300,00 a R\$17500,00. A partir desses dados, pode-se deduzir que o custo de um robô nesta categoria é significativamente elevado, especialmente quando se leva em consideração o contexto universitário, que muitas vezes depende de recursos públicos para investimentos.

Tabela I – Exemplos de robôs usados na área de educação e em pesquisas. Conversão para real com um dólar homólogo a R\$ 4,84, uma Libra R\$ 6,32 e um Euro a R\$ 5,43, realizada 17/07/2023.

Robô	Ano	Custo
Pioneer 3DX	1995	R\$ 17.571,02
Amigobot	2001	R\$ 17.119,92
E-puck	2009	R\$ 4.567,88
Turtlebot 2	2012	R\$ 9.092,58
Rice r-one	2013	R\$ 1.305,12
Colias	2014	R\$ 1.631,38
Edison	2014	R\$ 2.803,32
Khepera IV	2016	R\$ 16.299,25
MicroMVP	2017	R\$ 4.567,88
Turtle 3 (burger)	2017	R\$ 3.184,57
Turtle 3 (waffle)	2017	R\$ 7.603,42
Rosbot 2.0	2018	R\$ 9.027,36
Husky A200	-	R\$ 1.497,30
Summit-XL	-	R\$ 7.003,50
Robotino	-	R\$ 4.105,50
E-puck2	-	R\$ 4.830,00
Turtlebot 2i	-	R\$ 10.626,00

Fonte: Adaptado de (P. Slaets, 2019; B. Lennox, 2018; L. Marín, 2018).

Um elemento de extrema importância para qualquer sistema robótico é a arquitetura do *software*. Devido à complexidade das pesquisas nessa área, os projetistas frequentemente utilizam arquiteturas existentes como base para seus trabalhos. Dentre as opções disponíveis, o *Robot Operating System* (ROS) tem se destacado significativamente, graças à sua popularidade tanto no meio industrial quanto no acadêmico. O ROS oferece um conjunto de ferramentas de software, bibliotecas e convenções que simplificam o desenvolvimento de aplicações para robôs, além de contar com uma comunidade ativa que compartilha módulos reutilizáveis.

Considerando as necessidades da atualidade, foi desenvolvido o ROS 2, uma versão mais atual e melhorada da plataforma de desenvolvimento de software de código aberto voltada para a área da robótica. Seu objetivo é semelhante ao de sua versão anterior, fornecendo uma base de software padrão para desenvolvedores em diversos setores, abrangendo desde pesquisas e prototipagem até implantação e produção. Ambos utilizam um sistema de publicação e assinatura semelhante, mas a segunda versão apresenta o middleware DDS/RTPS, amplamente empregado em sistemas de tempo real, proporcionando uma comunicação eficiente via rede [19].

Algumas características que tornam o ROS 2 mais atrativo em comparação com a versão original são:

- Suporta as versões 11 e 14 do padrão C++;
- Cada nó é capaz de descobrir outros nós, eliminando a necessidade de um nó mestre;
- Oferece opções para escolher políticas de qualidade de serviço adequadas;
- É compatível com sistemas operacionais como Ubuntu, Mac OS e Windows 10, e não apenas com Linux;
- Conta com suporte para respostas em tempo real, permitindo aplicações que exigem alta performance e tempos de resposta rápidos.

Essas melhorias e recursos adicionais tornam o ROS 2 uma escolha mais avançada e abrangente para os desenvolvedores de robótica na contemporaneidade.

No âmbito da estrutura eletrônica do robô, a utilização de dispositivos mutáveis é comum. Dentre os diversos componentes eletrônicos disponíveis, destaca-se aqui o Raspberry Pi e o Arduino, que são amplamente empregados. O Arduino é uma plataforma programável de baixo custo, com hardware e software de fácil acesso e manipulação, compatível com Mac, Windows e Linux, além de contar com uma comunidade ativa e oferecer facilidade de adaptação e expansão. Por outro lado, o microcomputador Raspberry Pi é econômico e possui uma vasta gama de exemplos disponíveis na Internet. Ele suporta sensores de piloto automático e pode ser configurado com o sistema operacional Ubuntu, que é naturalmente compatível com o ROS 2.

Nesse contexto, a criação de uma plataforma padronizada é a solução para reduzir potenciais custos de manutenção e complexidade. É relevante ressaltar também a importância do desenvolvimento da robótica no Brasil, buscando acompanhar o progresso dessa tecnologia. Consequentemente, para desenvolver algoritmos e métodos de resolução de problemas nessa área, torna-se fundamental verificar o programa de alguma forma, seja por meio de simulações, teoria ou testes práticos.

Este trabalho propõe o desenvolvimento de um protótipo de um robô móvel destinado à pesquisa em robótica cooperativa, acompanhado de uma respectiva documentação didática, para que outros estudantes de uma graduação possam desenvolver soluções nesta plataforma.

O nome atribuído ao projeto é SciCoBot (Science, Coffee and Robots), e ele foi projetado para empregar o sistema ROS 2 em conjunto com os dispositivos Arduino e Raspberry Pi. Ao utilizar a segunda versão do sistema operacional projetado para robótica, não há necessidade de uma estação central, permitindo que a comunicação ocorra de forma descentralizada. Além do mais, a acessibilidade é um dos aspectos primordiais considerados no projeto, tornando o SciCoBot adequado para uso interno e proporcionando um software modular que permita sua adaptação em pesquisas futuras de robótica cooperativa, especialmente no campus Patos de Minas da Universidade Federal de Uberlândia (UFU).

II. MATERIAIS E MÉTODOS

Considerando um robô inteligente capaz de desempenhar tarefas complexas em diversos ambientes, com o objetivo de realizar experimentos futuros, é essencial que a plataforma

escolhida ofereça autonomia, custo acessível, poder computacional, versatilidade, acessibilidade e conectividade. Devido a essas exigências, foi decidido utilizar o Arduino em conjunto com a placa Raspberry para aproveitar ao máximo cada componente. O Raspberry Pi é escolhido por sua capacidade de processamento, enquanto o Arduino é utilizado para controlar os sensores em níveis mais baixos, permitindo a execução simultânea de tarefas e tornando possíveis aplicações mais complexas.

Além do Arduino e do Raspberry Pi, outros componentes, como um chassi de acrílico, conversor de tensão 12V/ 5V/ 3.3V, sensor ultrassônico, ponte H, conversor de nível lógico e enconders foram selecionados para complementar o robô e aprimorar sua funcionalidade e eficiência. A Figura 1 indica os elementos escolhidos, oferecendo uma representação visual do projeto:

Figura 1: Componentes SciCoBot.



Fonte: Autor.

Na Tabela 2 são apresentados os materiais necessários e suas respectivas estimativas de custos para a aquisição dos componentes utilizados na construção do SciCoBot. É importante mencionar que os itens de maior valor são o Raspberry 3B+ e o Arduino Due, cuja soma representa mais de 60% do custo total.

Tabela 2: Materiais utilizados e seus respectivos custos

Nome	Preço	Quantidade
Placa Arduino Due	R\$ 349,90	1
Raspberry 3B+	R\$ 899,97	1
Demais componentes eletrônicos	R\$ 426,39	-
Total:	R\$ 1.676,23	

Fonte: MercadoLivre e Aliexpress, 17/07/2023.

O desenvolvimento da estrutura do robô seguiu as seguintes etapas:

1. Realizou-se um estudo detalhado de plataformas com conceitos semelhantes, com o objetivo de identificar possíveis problemas, analisar o design de *software*, as ferramentas utilizadas e as funcionalidades implementadas.
2. Definiu-se a comunicação entre as placas Arduino e Raspberry, estabelecendo uma estratégia eficiente para a troca de informações entre os dois componentes.
3. Enumerou-se todos os componentes utilizados no projeto, identificando cada elemento necessário para o funcionamento adequado do robô.
4. Montaram-se diversas bibliotecas Arduino para atender às diferentes necessidades dos componentes

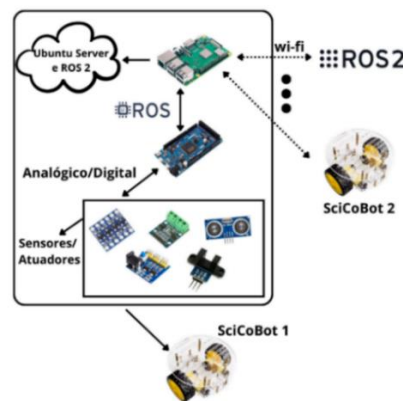
já estabelecidos, garantindo uma boa integração e um controle adequado das funcionalidades.

5. No Raspberry, foram criados pacotes ROS 2 em linguagem C++ para lidar com as informações provenientes do sistema ROS 2, assegurando a manipulação correta dos dados.
6. Foram realizados testes extensivos de comunicação entre os robôs, a fim de verificar a eficiência e confiabilidade da interação entre eles.
7. Com o intuito de facilitar a utilização da plataforma por futuros usuários, foi produzido um manual simples e didático, contendo instruções claras e detalhadas sobre o funcionamento e operação do robô. Utilizando o SciCoBot original e o conteúdo produzido, realizou-se uma pesquisa detalhada sobre seus elementos, conexões e metodologia empregada, visando possibilitar uma reprodução do mesmo. Durante essa fase, também se empenhou em compreender os comandos ROS 2 utilizados nos exemplos, bem como as informações obtidas a partir dos resultados correspondentes. Além disso, buscou-se aprimorar o conhecimento sobre máquinas virtuais, comandos do terminal Linux, bibliotecas Arduino e estabelecimento de conexões via SSH.

III. DESENVOLVIMENTO

O projeto adota uma estrutura similar à ilustrada na Figura 2. O robô é composto por um Raspberry Pi como elemento de sistema de alto nível, enquanto um Arduino Due é utilizado como controlador e leitor de sensores. A placa Raspberry Pi desempenha um papel crucial na coordenação entre os dois dispositivos na rede. Para isso, o Pi conta com um ambiente de trabalho ROS 2 Foxy e conectividade Wi-Fi, permitindo a interação fluída entre as camadas do *framework*.

Figura 2: Arquitetura de hardware e estrutura de comunicação.



Fonte: Autor.

Optou-se pelo uso específico do Arduino Due devido à sua ampla quantidade de portas digitais e à compatibilidade com o micro-ROS (biblioteca que habilita ROS 2 para microcontroladores). Quanto ao Raspberry 3B+, sua escolha se deu pela disponibilidade já existente no laboratório.

Com a definição dos periféricos necessários, realizou-se uma pesquisa de artigos que apresentassem descrições de sensores utilizados em plataformas robóticas similares. Ao

mesmo tempo, foram analisados os componentes empregados em arquiteturas com designs semelhantes ao SciCoBot, buscando partes que pudessem ser utilizadas. A Tabela 3 exibe essa relação, destacando a câmera e o encoder (adaptado para sensor ultrassônico) como elementos proeminentes.

Tabela 3 - Robôs com finalidades similares ao SciCoBot.

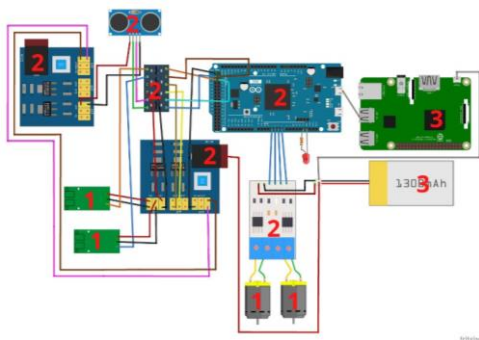
Robôs	Componentes
(Brezina, M. Thácik, S. Jadlovska, 2019) [5]	Raspberry, câmera, sonar e infravermelho.
Pheeno (S. Wilson, 2015) [6]	Encoder, IR e câmera.
Linorobot (J. Jimeno, 2021) [7]	Raspberry e motor com encoder.
Hadabot (Hadabot, 2021) [8]	ESP32, encoder e sensor de distância.
Nanosaur (R. Bonghi, 2022) [9]	NVIDIA Jetson Nano e câmera.
PR2 (Willow-Garage. Technical specifications., 2010) [10]	ROS, Linux e câmera

Fonte: Autor.

Para realizar a montagem do projeto, onde a conexão dos componentes pode ser visualizada esquematicamente na Figura 3, foram adquiridos os seguintes materiais: um chassi de acrílico, duas rodas e dois rodízios giratórios. Além disso, três discos adicionais foram utilizados para adicionar mais níveis à estrutura. Abaixo estão as descrições de todas as componentes alocadas em cada um dos respectivos níveis, bem como a Figura 4, uma ilustração criada para um fácil entendimento de futuros alunos em processos de recriações:

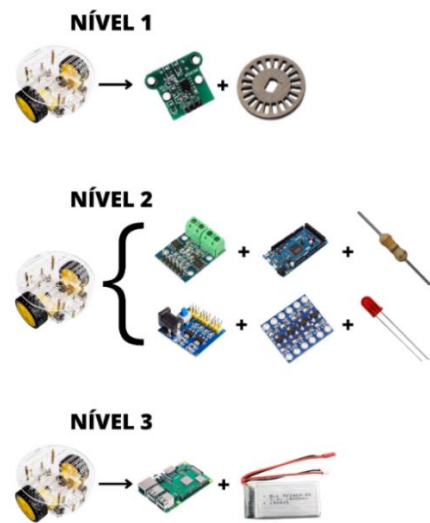
1. Chassi Inferior: foram colocados os *encoders* conectados aos eixos dos motores, enquanto na parte superior foram acomodados os sensores.
2. Chassi Intermediário: foram posicionados a ponte H, o Arduino Due, o conversor de nível lógico e os conversores de tensão 12V / 5V / 3.3V, responsáveis por converter a tensão de saída e entrada de alguns componentes. Também na parte intermediária, instalou-se um sensor ultrassônico. Adicionalmente, na porta D13 do Arduino, um LED e um resistor de 330 ohms foram acoplados.
3. Chassi Superior: foram posicionados a placa Raspberry Pi 3B+ e uma bateria, sendo conectadas por um fio que interliga as duas placas.

Figura 3: Arquitetura de hardware e estrutura de comunicação.



Fonte: Autor.

Figura 4: Ilustração da montagem de todos os níveis do SciCoBot.



Fonte: Autor.

Com base no conhecimento de cada componente do robô e nos conceitos de programação, foram criadas as primeiras bibliotecas em linguagem C++. Essas bibliotecas são as seguintes: *Arduino core*, *serial_debug*, *led_debug*, *different_drive*, *encoder*, *ultrassom* e *motor_control*. Além disso, optou-se por modularizar o *middleware*, encapsulando-o no *micro_ros_arduino*. Durante a implementação, foi notado um padrão recorrente, seguindo as etapas abaixo:

1. Declaração das entidades ROS sem suas atribuições;
2. Definição das funções *callbacks*;
3. Definição do transporte, nó, *allocator* e *support*;
4. Definição dos inscritesores, editores e *timers*, bem como a adição do(s) *timer(s)* e/ou inscriteor(es) a eles;
5. Definição de um *spin* para o executor mencionado.

Assim, foi criada uma arquitetura de *software* em que o *micro_ros_arduino* habilita a comunicação com os sensores e controla o movimento do robô dado pelo usuário. Nesse momento, foram desenvolvidas as bibliotecas *ros_init*, *ros_motor_control*, *ros_ultrasonic*, *ros_encoder* e *ros_move_on*, todas gerando funções *get* que retornam entidades para uso posterior.

Com as bibliotecas do Arduino prontas, foi criado um repositório no GitHub [11] com exemplos de uso dos sensores e atuadores. Esses exemplos incluem: movimentar o motor via teclado, movimentar o robô via teclado e publicar os dados dos *encoders*, publicar as medidas do sensor ultrassônico e movimentar o robô a partir do teclado enquanto publica dados dos sensores ultrassônicos e *encoders*.

Pacotes ROS 2 [12] foram desenvolvidos no Raspberry Pi para acessar, manipular e publicar os dados do sensor. Além disso, o software criado possui bibliotecas independentes, evitando o desperdício de recursos. Por essa razão, foram criados dois sistemas utilizando arquivos *.bash*: um para Arduino [13] e outro para Raspberry Pi [14], buscando reduzir a complexidade do gerenciamento de arquivos.

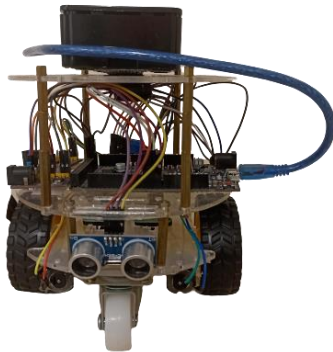
Outro aspecto importante a mencionar é a elaboração de material didático pensado para futuros projetos com a plataforma. Como o SciCoBot é um robô versátil e facilmente adaptável para diversas finalidades, foi criado um guia simples e direto, com instruções importantes para iniciantes acerca do ROS 2, permitindo que qualquer aluno com requisitos mínimos de conhecimento possa construir um robô do zero. O guia foi reproduzido passo a passo, enfatizando com imagens e descrições como replicar o robô cooperativo da UFU.

Vale ressaltar que todos os programas mencionados estão disponíveis no CRUZ (2022e) [15], o GitHub que contém todos os softwares desenvolvidos para o SciCoBot. Isso inclui o conteúdo do material explicativo e a apostila didática. Além disso, futuras atualizações do projeto serão postadas na página.

IV. RESULTADOS

Na Figura 5, pode-se visualizar o resultado do projeto descrito anteriormente. Nela está presente o SciCoBot, construído na Universidade Federal de Uberlândia, Campus Patos de Minas, juntamente com o produto obtido a partir da aplicação do guia destinado a principiantes, também desenvolvido na UFU Patos de Minas.

Figura 5: SciCoBot, o robô cooperativo UFU.



Fonte: Autor.

Após montar a plataforma física, realizaram-se testes executando projetos no Arduino e pacotes ROS no Raspberry para validar as implementações desenvolvidas para os componentes do SciCoBot. Os seguintes testes foram realizados:

- Controle do robô por teclado: No Arduino, um inscrito recebeu os comandos enviados por um editor no Raspberry. Isso permitiu mover os motores conforme os dados recebidos pelo Arduino.
- *Encoder*: Enquanto o robô se movimentava, a distância percorrida foi calculada. No Raspberry, um inscrito publicou os dados no terminal para visualização.
- Ultrassônico: Foi criado um editor ROS no Arduino e um inscrito no Raspberry. O editor publicou a distância medida pelo sensor ultrassônico, e as medições foram exibidas no terminal.

A fim de testar a comunicação entre dois robôs, uma simulação simples foi realizada entre o SciCoBot e um Raspberry. O SciCoBot enviou os dados lidos do *encoder* para o Raspberry, que, por sua vez, retornou a temperatura para o

SciCoBot. Para viabilizar esse teste, um computador foi utilizado exclusivamente como conexão SSH para executar e visualizar os comandos aplicados.

Através da utilização de um sistema de multicast, no qual todos os inscritos têm acesso às informações produzidas, outros robôs podem, dessa maneira, receber os dados. A Figura 6 indica os terminais de comando, onde é apresentada a temperatura de 49,388000 e a informação do encoder é mostrada como zero. Com base nessas informações, verificou-se que a temperatura enviada é recebida pela plataforma e as informações do encoder são acessadas no Raspberry. Dessa forma, constatou-se que é possível utilizar o Wi-Fi e o ROS 2 Foxy para transmitir dados entre dois dispositivos que utilizam o ROS 2.

Figura 6: Saída dos Terminais do SciCoBot e do Raspberry durante a simulação de cooperação.

```
[scicobot_cooperation]: Temperature: '49,388000'
[scicobot_cooperation]: Encoder counts: '0'
[scicobot_cooperation]: Encoder counts: '0'
[scicobot_cooperation]: Encoder counts: '0'
[scicobot_cooperation]: Temperature: '49,388000'
[rasp_cooperation]: I heard: '0'
[rasp_cooperation]: I heard: '0'
[rasp_cooperation]: Publishing: '49,388000'
[rasp_cooperation]: I heard: '0'
[rasp_cooperation]: I heard: '0'
```

Fonte: Autor.

Adicionalmente, um manual educativo foi elaborado considerando a comunidade e, principalmente, os estudantes que possuem conhecimentos básicos em circuitos elétricos. O guia abrange diversos exemplos de uso do ROS 2, desde os mais simples até comandos intermediários essenciais para compreender o funcionamento do Scicobot. Ele também contém explicações detalhadas sobre os componentes empregados e a construção de *hardware* e *software*.

Todo o material produzido com a plataforma original foi utilizado como base, e em seguida, foi feito um esforço para reproduzir cuidadosamente cada etapa, destacando e ilustrando todo o processo realizado. Um registro detalhado foi elaborado, abrangendo desde a função de cada componente até a criação completa do sistema, incluindo a instalação do ROS 2 Foxy, a configuração do Arduino e também, a implementação dos exemplos criados. Além disso, o manual contém uma breve explicação sobre Linux e máquinas virtuais, além de particularidades exclusivas relacionadas ao ambiente de utilização interno da UFU-Patos. Cada passo foi minuciosamente documentado, uma lista dos principais tópicos pode ser visualizada abaixo, para permitir uma replicação precisa e facilitar o entendimento de todo o procedimento executado.

1. Definições de componentes;
2. Montagem;
3. Criação de uma máquina virtual Ubuntu 20.04 no software Oracle VirtualBox;
4. Instalação do ROS 2 Foxy no Ubuntu 20.04;
5. Instalação do Ubuntu 20.04 no Raspberry 3B+;
6. Conexão SSH;
7. Resolução de erros comuns;
8. Instalação do Arduino e pacotes;
9. Simulação de exemplos.

V. CONCLUSÕES

O foco deste trabalho foi o desenvolvimento de uma plataforma robótica de baixo custo e modular, destinada principalmente para pesquisas na UFU, campus Patos de Minas. Além de apresentar os resultados obtidos, o trabalho descreveu os conceitos teóricos fundamentais, os materiais utilizados para montar o robô e detalhou a metodologia, procedimentos e processo de desenvolvimento empregados.

Como resultado, os objetivos de criar um robô acessível, adequado para uso interno, com software modular e aplicável em pesquisas foram alcançados. O projeto do SciCoBot se inspirou em conceitos de outras plataformas existentes para definir melhor o software e os componentes a serem utilizados. Isso resultou na criação de uma plataforma que inclui exemplos práticos de implementações baseadas na estrutura de software proposta.

A comunicação entre os dispositivos e o software foi testada e comprovada experimentalmente. A comparação entre o SciCoBot e outras estruturas estudadas demonstrou que o projeto UFU se destaca por permitir que os alunos utilizem conhecimentos prévios e componentes disponíveis nos laboratórios de ensino, enfatizando o baixo custo e facilitando o processo de aprendizado.

Além disso, foi desenvolvido um guia ilustrado e explicativo para conceber um robô similar ao projeto UFU. Esse guia aproveitou o material já elaborado e os exemplos criados para explorar de forma detalhada todas as interfaces e como as alterações afetam as simulações realizadas. Dessa forma, o guia ilustra passo a passo todas as etapas do processo e estuda possíveis adaptações ou mudanças para facilitar o aprendizado dos futuros usuários.

A análise comparativa entre o SciCoBot e outras referências revelou que a vantagem da plataforma UFU está na sua documentação abrangente em português, que aborda conceitos e ferramentas robóticas relevantes. Além disso, o sistema é adequado para implantação na UFU, permitindo a seleção de dispositivos comumente utilizados durante o curso e disponíveis no laboratório, o que facilita o uso da ferramenta em pesquisas pelos estudantes de Patos de Minas. Outro aspecto relevante é que os exemplos fornecidos são estruturas mais acessíveis, adequadas para projetos de média complexidade, o que faz do SciCoBot uma opção com um bom custo-benefício em comparação com outras alternativas no mercado.

AGRADECIMENTOS

Agradecemos à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) pelo financiamento do Projeto RoboPatos APQ-02616-22 e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela concessão de bolsa.

REFERÊNCIAS

[1] J. Fenerick, C. R. Volante (2020), *A Evolução Das Indústrias, os Benefícios da Automação e as Perspectivas do Mercado da Robótica no Brasil e no Mundo*. Interface Tecnológica, 2020, ISSN 2447-0864. Acedido em 11 de abril de 2021, em:

<https://revista.fatectq.edu.br/index.php/interfacetecnologica/article/view/805/510>.

- [2] P. Slaets, R. Amsters (2019), *Turtlebot 3 as a Robotics Education Platform*. *Robotics in Education. Advances in Intelligent Systems and Computing*, Springer, Cham, v. 1023.
- [3] B. Lennox, F. Arvin, S. Watson, *ROS Integration for Miniature Mobile Robots. 2018. Conference: 19th Towards Autonomous Robotic Systems (TAROS) Conference*. Acedido em 23 de janeiro de 2022, em: https://www.researchgate.net/publication/325499148_ROS_Integration_for_Miniature_Mobile_Robots.
- [4] L. Marín, *Modular Open Hardware Omnidirectional Platform for Mobile Robot Research*. IEEE, 2018.
- [5] A. Brezina, M. Thácik, S. Jadlovska, *Design of a Prototype for a Modular Mobile Robotic Platform*, IFAC Conference on Programmable Devices and Embedded Systems PDES, 16ª Edição, 2019.
- [6] S. Wilson (2015), *Pheeno, A Versatile Swarm Robotic Research and Education Platform*, IEEE Robotics and Automation Letters.
- [7] J. Jimeno (2021), *GitHub Linorobot*. Acedido em 20 de janeiro de 2022, em: <https://github.com/linorobot/linorobot>.
- [8] Hadabot (2021), *Learn ROS2 robotics by building a Hadabot robot*. Acedido em 22 de janeiro de 2022, em: <https://www.hadabot.com/>.
- [9] R. Bonghi, *Nanosaur. 2022a*. Acedido em 22 de janeiro de 2022, em: <https://github.com/rnanosaur>.
- [10] Willow-Garage (2010), *Technical specifications*. Acedido em 25 de junho de 2023, em: <http://www.willowgarage.com>.
- [11] O. Cruz, *GitHub scicobot_arduino. 2022d*. Acedido em 26 de janeiro de 2022, em: https://github.com/SciCoBot/scicobot_arduino.
- [12] O. Cruz, *GitHub scicobot_rasp. 2022a*. Acedido em 26 de janeiro de 2022, em: https://github.com/SciCoBot/scicobot_rasp.
- [13] O. Cruz, *GitHub build_scicobot_arduino. 2022b*. Acessado em 26 de janeiro de 2022, em: https://github.com/SciCoBot/build_scicobot_arduino.
- [14] O. Cruz, *GitHub build_scicobot_rasp. 2022c*. Acedido em 26 de janeiro de 2022, em: https://github.com/SciCoBot/build_scicobot_rasp.
- [15] O. Cruz, *GitHub SciCoBot. 2022e*. Acedido em 18 de fevereiro de 2022, em: <https://github.com/SciCoBot>.
- [16] Arduino, *Writing a Library for Arduino, 2022b*. Acedido em 26 de janeiro de 2022, em: <https://www.arduino.cc/en/Hacking/libraryTutorial>.
- [17] Micro_Ros_Arduino (2023), *Micro-ros library for arduino*. Acedido em 23 de janeiro de 2022, em: https://github.com/micro-ROS/micro_ros_arduino.
- [18] O. Cruz (2022), *Desenvolvimento de um Robô Móvel Open Source Baseado em ROS 2 para Pesquisa em Robótica Cooperativa*. Acedido em 01 de agosto de 2022.
- [19] ROS2. ROS 2. Documentation. 2022a. Acedido em 22 de janeiro de 2022, em: <https://docs.ros.org/en/foxy/index.html>.