

**CENTRO PAULA SOUZA  
FATEC SANTO ANDRÉ  
Tecnologia em Mecatrônica Industrial**

**Lucas Henrique Delgado Barbosa**

**Vinicius Fagundes de Medeiros**

**PLATAFORMA UNIVERSAL DE ROBÔ MÓVEL SOBRE RODAS  
CONTROLADA REMOTAMENTE**

**Santo André**

**2023**

**Lucas Henrique Delgado Barbosa**  
**Vinicius Fagundes de Medeiros**

**PLATAFORMA UNIVERSAL DE ROBÔ MÓVEL SOBRE RODAS  
CONTROLADA REMOTAMENTE**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Mecatrônica Industrial da FATEC Santo André como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial.

Orientador: Prof. Dr. Edson C. Kitani  
(FATEC)

Coorientador: Prof. Dr. Luiz A. Celiberto  
Junior (UFABC)

**Santo André**  
**2023**

## FICHA CATALOGRÁFICA

M488p

Medeiros, Vinicius Fagundes de  
Plataforma universal de robô móvel sobre rodas controlada remotamente / Vinicius Fagundes de Medeiros, Lucas Henrique Delgado Barbosa. - Santo André, 2023. – 73f: il.

Trabalho de Conclusão de Curso – FATEC Santo André.  
Curso de Tecnologia em Mecatrônica Industrial, 2023.

Orientador: Prof. Dr. Edson Caoru Kitani

1. Mecatrônica. 2. Robô móvel. 3. Construção. 4. Desenvolvimento. 5. Tecnologia. 6. Plataforma. 7. Pesquisa. 8. Programa ESP-32. 9. Linguagem Python. 10. Robótica. I. Barbosa, Lucas Henrique Delgado. II. Plataforma universal de robô móvel sobre rodas controlada remotamente.

**LISTA DE PRESENÇA**

Santo André, 19 DE JUNHO DE 2023.

LISTA DE PRESENÇA REFERENTE À APRESENTAÇÃO DO  
TRABALHO DE CONCLUSÃO DE CURSO COM O TEMA:  
“PLATAFORMA UNIVERSAL DE ROBÔ MÓVEL SOBRE RODAS  
CONTROLADA REMOTAMENTE” DOS ALUNOS DO 6º  
SEMESTRE DESTA U.E.

**BANCA**

PRESIDENTE:

PROF. EDSON CAORU KITANI \_\_\_\_\_

MEMBROS:

PROF. LUIZ CELIBERTO \_\_\_\_\_

PROF. FERNANDO GARUP DALBO \_\_\_\_\_

**ALUNOS:**

LUCAS HENRIQUE DELGADO BARBOSA \_\_\_\_\_

VINÍCIUS FAGUNDES DE MEDEIROS \_\_\_\_\_

## DEDICATÓRIA

Dedicamos este trabalho aos familiares e todas as pessoas que ajudaram e estiveram presente no desenvolvimento deste projeto.

## **AGRADECIMENTO**

Gostaríamos de agradecer a Deus, por ter-nos dado força, foco, fé e sabedoria para concluirmos esta etapa de nossas vidas.

Aos nossos familiares e amigos por terem nos apoiados, por terem dado força nos momentos difíceis e por não deixarem nós desistirmos.

Ao nosso orientador Prof. Dr. Edson Caoru Kitani e coorientador Prof. Dr. Luiz Antonio Celiberto Junior da UFABC por nos direcionar e permitir com que nós façamos parte desta ideia.

Lucas gostaria de agradecer aos seus familiares, principalmente sua mãe Helena Barbosa Delgado e seu pai Ernando José Barbosa, por estarem sempre presentes e por terem dado todo o suporte necessário.

Juntos agradecemos também ao nosso amigo Juliano Casseli Vieira, que começou o curso conosco, mas infelizmente parou no meio do caminho, honraremos sua passagem pelo curso!

## EPÍGRAFE

“Não tente se tornar uma pessoa de sucesso, tente se tornar uma pessoa de valor!”

*Albert Einstein*

## RESUMO

Robôs móveis autônomos tem sido objetos de grande interesse e pesquisa nas universidades, principalmente pela grande aplicabilidade que ele tem em várias atividades humanas, seja transportando bens e objetos ou em entretenimento e apoio aos humanos. Contudo, uma das grandes dificuldades tem sido a disponibilidade de uma plataforma robótica de baixo custo, mas com capacidade de carga para atender diferentes pesquisas e aplicações. E para auxiliar na solução do problema de disponibilidade de recursos para pesquisa, este projeto tem como proposta o desenvolvimento e construção de um robô capaz de transportar cargas e equipamentos. O objetivo é desenvolver uma estrutura robótica e um sistema de controle remoto que seja capaz de se locomover através dos ambientes sendo direcionado pelo programa desenvolvido. Esta plataforma universal controlada remotamente servirá para implementação de projetos futuros. Este projeto foi criado utilizando novas tecnologias a fim de desenvolver conhecimentos na área da mecatrônica. Entre os objetivos está a possibilidade de expandir a quantidade de robôs na FATEC Santo André para novos projetos. O protótipo foi construído utilizando ferramentas de baixo custo, mas que puderam proporcionar um robô funcional, tudo com base no controle através de um ESP-32. Um programa escrito em Python será o responsável pela tomada de decisões. Foram realizados vários testes com diversas cargas a fim de demonstrar o desempenho do robô em diferentes situações. O protótipo passou nos testes mostrando ser eficiente com uma quantidade considerável de carga, apresentou alguns problemas mecânicos que deverão ser resolvidos e melhorados em propostas futuras.

Palavras-chave: Robôs Móveis. Autônomos. Robótica. ESP-32. Plataforma.

## ABSTRACT

Autonomous mobile robots have been objects of great interest and research at universities, mainly due to their great applicability in various human activities, whether transporting goods and objects or in entertainment and support for humans. However, one of the significant difficulties has been the availability of a low-cost robotic platform with load capacity to meet different research and applications. And to help solve the problem of resource availability for research, this project proposes developing and constructing a robot capable of transporting loads and equipment. The objective is to create a mechanical structure and a remote-control system that can move through environments being directed by the developed program. This remotely controlled universal platform will serve for the implementation of future projects. This project was created using new technologies to develop knowledge in the field of mechatronics. Among the objectives is the possibility of expanding the number of robots at FATEC Santo André for new projects. The prototype was built using low-cost tools that could provide a functional robot, all based on control through an ESP-32. A program written in Python will be responsible for making decisions. Several tests were carried out with different loads to demonstrate the robot's performance in different situations. The prototype passed the tests showing efficiency with a considerable load; it presented some mechanical problems that will have to be solved and improved in future proposals.

Keywords: Mobile Robots. Autonomous. Robotics. ESP-32. Platform.

## Lista de Quadros

Quadro 1 - Tabela comparando os recursos disponíveis no ESP8266 e no ESP-32. .....	26
Quadro 2 - Tabela dos resultados dos testes de carga e consumo de corrente dos motores na menor velocidade. ....	48
Quadro 3 - Tabela dos resultados dos testes de carga e consumo de corrente dos motores na maior velocidade. ....	48
Quadro 4 - Tabela dos testes de curvatura do robô. ....	51
Quadro 5 - Tabela dos principais custos do projeto. ....	55

## Lista de Imagens

Figura 1- Imagem do robô RHINO, desenvolvido para guiar pessoas em ambiente público. ....	16
Figura 2. Na figura (a) uma vista geral do Kihon juntamente com a computação remota (Notebook), na figura (b) o robô realizando a operação de seguir uma linha sobre a mesa. ....	17
Figura 3. Em (a) temos a arquitetura de hardware do controle (Arduíno) juntamente com os sensores e atuadores e em (b) a arquitetura de software.....	18
Figura 4. Robô mapeador de ambientes usando uma câmera Kinetic e um Arduíno para controlar dois motores. ....	19
Figura 5. Na figura (a) uma vista interna do robô proposto por Buzzi, 2019, e em (b) o mesmo robô com a carenagem.. ....	19
Figura 6 – Robô sobre rodas Husky desenvolvido pela Clearpath para aplicações gerais. ....	20
Figura 7 – Grupo de engenheiros da Clearpath montando os sensores no corpo do robô Husky. ....	20
Figura 8 – Robô Contadino em seu ambiente de trabalho mostrando o uso de um de seus sensores. ....	21
Figura 9 - Robô SCOUT em frente à casa de um cliente para mostrar que consegue chegar até o endereço desejado. ....	22
Figura 10 - Robô TROMBIA FREE realizando limpeza urbana no meio do dia. ....	23
Figura 11 – Arquitetura simplificada da montagem dos componentes dentro do robô plataforma. ....	25
Figura 12 - Imagem de um ESP-32. ....	27
Figura 13 - Imagem que representa as funções de cada pino do ESP-32. ....	27
Figura 14 - Desenho eletrônico de uma Ponte H e seus contatos. ....	28
Figura 15 - Representação do giro de um motor através da Ponte H. ....	29
Figura 16 – Demonstração dos possíveis sinais PWM. ....	29
Figura 17 – Imagem do corpo do robô plataforma montado. ....	31
Figura 18 - Imagem para ver a parte interna do robô plataforma com os componentes internos montados. ....	33
Figura 19 - Imagem para mostrar a face do robô. ....	34
Figura 20 – Modelo da Furadeira utilizada no projeto. Em vermelho os furos de fixação do corpo da furadeira. ....	35
Figura 21 – Desenho para demonstrar as ligações eletrônicas feitas no projeto. ....	36
Figura 22 – Imagem para mostrar os pontos de conexão da Ponte H BTS7960 com o motor e a fonte. ....	36
Figura 23 – Imagem para mostrar os pinos da Ponte H. ....	37
Figura 24 - Demonstração das bibliotecas utilizadas no desenvolvimento do programa. ....	38
Figura 25 - Ilustração do ponto de acesso criado no ESP-32. ....	38
Figura 26 - Definição dos pinos dos motores. ....	39

Figura 27 - Função criada para movimentar o Robô Plataforma. ....	40
Figura 28 - Parte do programa que representa a página html. ....	41
Figura 29 - Demonstração da criação de um novo ciclo para representar a conexão entre o servidor e o cliente. ....	42
Figura 30 – Forma com que os dados em bytes são convertidos em <i>string</i> . ...	42
Figura 31 - Criação dos botões que correspondem as variáveis. ....	42
Figura 32 - Parte do programa que atribui valores aos botões. ....	43
Figura 33 - Função que é chamada para definir a resposta do cliente. ....	43
Figura 34 - Print da tela do celular com a página HTML aberta com o robô ao lado para demonstrar o sentido das setas em relação ao robô. ....	44
Figura 35 - Local do Teste em Linha Reta com Diferentes Pesos do Robô Plataforma. ....	45
Figura 36 - Robô Plataforma com os pesos de 10 kg e 50 kg em cima. ....	46
Figura 37 - Robô Plataforma com o peso de 100 kg em cima. ....	46
Figura 38 - Gráfico expressando a relação entre os níveis de PWM necessários para movimentação de acordo com a carga que o robô está carregando em sua menor velocidade. ....	48
Figura 39 - Gráfico que mostra a relação da velocidade e da carga, quanto maior a carga menor a velocidade. ....	49
Figura 40 - Local do Teste de Curvatura do Robô Plataforma. ....	50
Figura 41 - Representação do sentido de giro do robô no primeiro teste. ....	51
Figura 42 - Representação do sentido de giro do robô no segundo teste. ....	51

## **Lista de Abreviaturas e Siglas**

IA	Inteligência Artificial
PIB	Produto Interno Bruto
RTK	Real Time Cinematic
GPS	Global Positioning System
IoT	Internet Of Things
RTOS	Real Time Operating System
PWM	Pulse Width Modulation

## Sumário

1.	Introdução .....	13
1.1.	Motivação .....	13
1.2.	Objetivo .....	14
1.3.	Metas Principais .....	14
1.4.	Organização Textual.....	15
2.	Referencial Teórico .....	16
2.1.	Trabalhos Semelhantes .....	16
2.2.	Considerações Finais .....	23
3.	Metodologia e Desenvolvimento do Projeto .....	24
3.1	Arquitetura do Projeto.....	24
3.2.	ESP-32.....	25
3.3.	Aplicações .....	26
3.4.	Estrutura de Pinagem do ESP-32.....	27
3.5.	Ponte H .....	28
3.7.	Controle de Rotação e Direção.....	28
3.8.	Controle de Velocidade .....	29
3.9.	Desenvolvimento do robô proposto .....	30
3.10.	Chassi de Madeira.....	31
3.11.	Motor e Redutor.....	32
3.12.	Instalação das Furadeiras.....	34
3.13.	Desenvolvimento do Circuito Eletrônico.....	35
3.14.	Desenvolvimento do Software de Controle .....	37
4.	Testes e Análise dos Resultados .....	45
4.1.	Teste de Capacidade de Realização de Curvas .....	50
4.2.	Análise dos Resultados .....	52
5.	Conclusão .....	53
6.	Referências.....	56
	Apêndice 1 – Dimensões da Peça Traseira .....	58
	Apêndice 2 – Dimensões da peça Dianteira.....	59
	Apêndice 3 – Lateral Esquerda do Robô.....	60
	Apêndice 4 – Lateral Direita do Robô.....	61

Apêndice 5 – Dimensões da peça do Topo .....	62
Apêndice 6 – Dimensões da peça da Base .....	63
Apêndice 7 – Dimensões do Eixo Traseiro.....	64
Apêndice 8 – Dimensões do Eixo Dianteiro .....	65
Apêndice 9 – Montagem Simplificada do Robô .....	66
Apêndice 10 – Programa Desenvolvido em Python .....	67

## 1. Introdução

No contexto atual do século vinte e um, a indústria está mergulhando na Revolução 4.0, caracterizada pela integração estratégica da *Internet das Coisas* e do armazenamento de dados na nuvem, objetivando a otimização da produção de recursos e a melhoria da qualidade de vida. Diante da contínua evolução social, muitos elementos que antes pertenciam exclusivamente ao reino da ficção agora se concretizam como realidade

E com o avanço da tecnologia na área da robótica, tem-se criado inúmeras possibilidades, e uma delas é a possibilidade de criar uma máquina autônoma, um robô que possa se locomover sem que um humano esteja no controle. Criar algo que faça suas ações por conta própria parece algo complexo e até difícil de acreditar, por isso um dos passos é programar algoritmos de Aprendizado de Máquina para que a interferência humana seja mínima. Além de toda computação é necessário também dotar o robô com um sistema de GPS conectado a central para localizar o aparelho dentro da fábrica ou até mesmo nas ruas, um sistema com sensores para detectar relevo e obstáculos, uma câmera conectada a um algoritmo de Visão Artificial e aos sensores de percepção do ambiente para que a locomoção do robô seja autônoma e interativa com o ambiente.

E esse cenário tem se tornado mais comum do que imaginamos, pois já existem empresas fornecendo robôs móveis para aplicações, não somente industriais, mas também em ambientes menos estruturados tais como plantações agrícolas e mineração.

### 1.1. Motivação

A robótica industrial já está bem estabelecida, pois desde a década de 60, a indústria automotiva vem investindo na instalação de manipuladores robóticos para realizar operação pesadas ou em ambientes insalubres, tais como: soldagem e pinturas. Contudo, há uma nova tendência que vem tomando espaço no ambiente industrial, que são os robôs móveis, aplicados e operações de logística para movimentação de cargas.

Portanto, há interesse muito grande nas pesquisas com robótica móvel pois, diferentemente dos seus irmãos manipuladores que são fixos, os robôs móveis

precisam se locomover e interagir nos mesmos espaços físicos que os humanos. E há um grande interesse e motivação nas universidades para o desenvolvimento da estrutura física de um robô móvel que possa circular livremente pelo ambiente em que ele se encontra, e ao mesmo tempo que se locomove, mande informações para a central, que monitorará o percurso, cujas informações conterão dados de terrenos e até mesmo imagens.

Um robô com todos esses recursos estaria apto para realizar o transporte de materiais e/ou cargas dentro de uma indústria. Pode ainda ser utilizado em outras áreas profissionais tais como: transportar equipamentos e ferramentas de uma área para outra. E para atender os interesses da comunidade acadêmica que estuda a robótica móvel, observou-se que um dos grandes desafios é construir uma plataforma mecânica e elétrica que atenda aos requisitos mínimos de robustez, baixo custo e flexibilidade de aplicação. Apesar de existirem soluções de mercado, a grande maioria delas não atende o requisito de baixo custo.

## **1.2. Objetivo**

O objetivo deste projeto de graduação projeto é criar uma plataforma robótica sobre rodas que possa transportar cargas, sem exceder uma certa quantidade de peso, de um ponto para outro dentro de uma indústria ou de outro ambiente. Após o desenvolvimento de um mecanismo que possa transportar carga de um ponto ao outro sem sofrer danos, teremos como objetivo desenvolver e aplicar no robô um sistema que consideremos independente, que dependendo da situação, possa tomar suas próprias decisões mediante a uma ordem de fatores.

A principal meta deste projeto é desenvolver a estrutura mecânica e elétrica de um robô sobre rodas, e que possa se mover sozinho no ambiente em que ele for colocado transportando carga de um ponto ao outro. O robô será dotado de 4 rodas e dois motores e uma eletrônica de controle dos motores. Será instalado também um microcontrolador para atuar nos motores usando o recurso do WiFi, para permitir o controle remoto robô via um aplicativo de celular.

## **1.3. Metas Principais**

O trabalho desenvolvido tem as seguintes metas:

- Pesquisar sobre os robôs móveis sobre rodas de aplicações gerais

- Projetar e desenvolver a estrutura mecânica e eletrônica de um robô móvel capaz de transportar até 100 kg, capacidade de andar em diversos tipos de terreno;
- Ser capaz de receber atualizações e melhorias e novos recursos;
- Ter boa autonomia;
- Baixo custo de fabricação e com materiais encontrados no mercado local.

#### **1.4. Organização Textual**

O texto está organizado da seguinte maneira: O Capítulo 2 apresenta alguns projetos semelhantes ao que será desenvolvido neste projeto, e ilustrará alguns dos principais aspectos teóricos para iniciar o projeto de um robô móvel sobre rodas.

O Capítulo 3 apresenta a metodologia de desenvolvimento e os trabalhos e ideias que foram aplicados como base para a execução do projeto. No Capítulo 4 serão apresentados os resultados dos testes com o robô e análise dos resultados e, finalmente, no Capítulo 5 as conclusões deste trabalho.

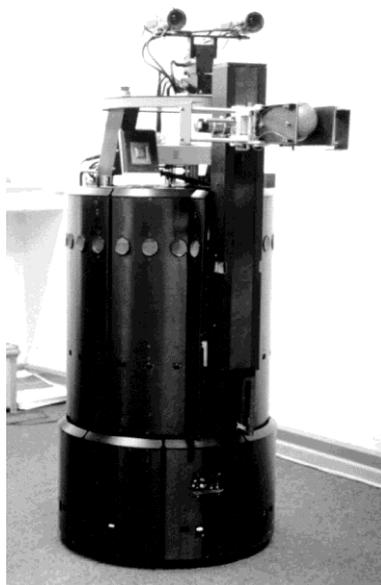
## 2. Referencial Teórico

Para melhor compreensão do tema proposto para este projeto, no Capítulo 2 serão abordados estudos sobre a robótica, além de trabalhos de outros autores sobre plataformas robóticas móveis e componentes que serão necessários para esta realização.

### 2.1. Trabalhos Semelhantes

Antes dos anos 2000 algumas pessoas já tinham a ideia de implementar Inteligência Artificial em alguns robôs, para que eles pudessem interagir tanto com o público quanto com o ambiente.

Figura 1- Imagem do robô RHINO, desenvolvido para guiar pessoas em ambiente público.



Fonte: [www.researchgate.net](http://www.researchgate.net)

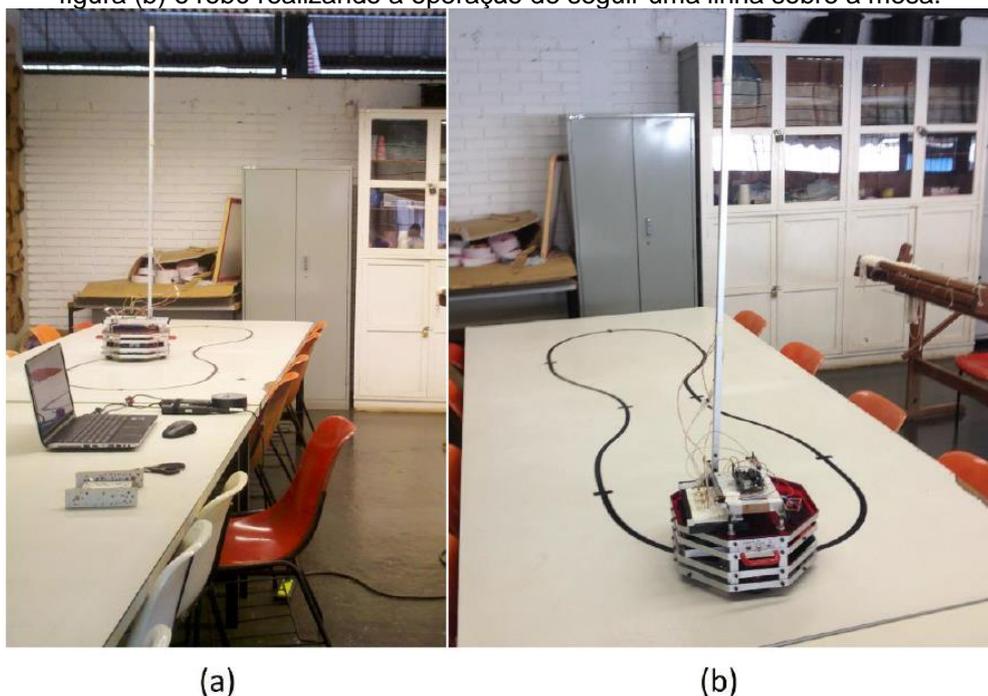
Um exemplo é o artigo de BUHMANN, et al. (2000) onde é apresentado a ideia de um robô chamado RHINO que foi desenvolvido na Universidade de Bonn, e que era responsável por interagir e guiar as pessoas em um ambiente público, tal como um museu. Essa interação seria feita de maneira verbal, gráfica ou até mesmo sonora. Para que tal ideia pudesse sair do papel, RHINO precisaria contar com uma inteligência artificial, sensores ultrassônicos, infravermelhos e laser. E já que seria um robô interativo precisaria também de uma tela *touch screen*, além de toda a programação necessária para que tudo ocorra de maneira harmoniosa e sem interferência humana. O principal sensor de navegação do Rhino eram os Ultrassons,

montados no perímetro do corpo do Rhino, sendo utilizados também para montar o mapa de navegação do ambiente. A câmera instalada no topo do Rhino tinha dois propósitos, um para identificar os objetos do ambiente e outra para auxiliar na navegação do robô, fornecendo informações sobre local.

No Brasil já foram realizados muitos trabalhos e pesquisas voltadas para os robôs móveis sobre rodas, e muitas delas foram resultados de trabalhos de conclusão de curso ou de dissertações e mestrado e até mesmo em teses de doutorado.

Em Matsumura (2014), o autor apresentou a proposta do desenvolvimento de uma plataforma aberta para propósitos gerais denominado Kihon, que foi construído com uma estrutura de alumínio, mas com dimensões reduzidas. O objetivo era criar uma estrutura para pesquisar o desenvolvimento da robótica móvel usando um Arduíno com conexão com um computador externo usando RS232C. Também foram instalados no robô alguns sensores, tais como: ultrassom, sensores infravermelhos e encoders nas rodas. O objetivo era criar um o robô para coletar informações climáticas do ambiente onde ele se encontrava.

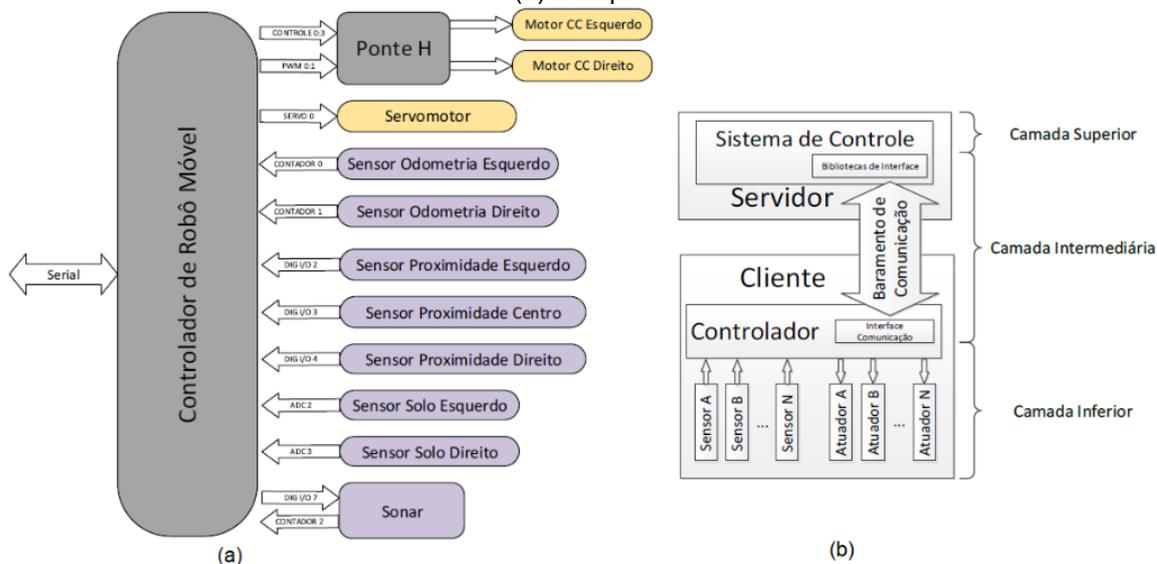
Figura 2. Na figura (a) uma vista geral do Kihon juntamente com a computação remota (Notebook), na figura (b) o robô realizando a operação de seguir uma linha sobre a mesa.



Fonte: Extraído de Matsumura, 2014.

O Arduíno foi utilizado para controlar localmente os motores do robô e realizava as operações da camada denominada intermediária da arquitetura proposta pelo autor.

Figura 3. Em (a) temos a arquitetura de hardware do controle (Arduíno) juntamente com os sensores e atuadores e em (b) a arquitetura de software.

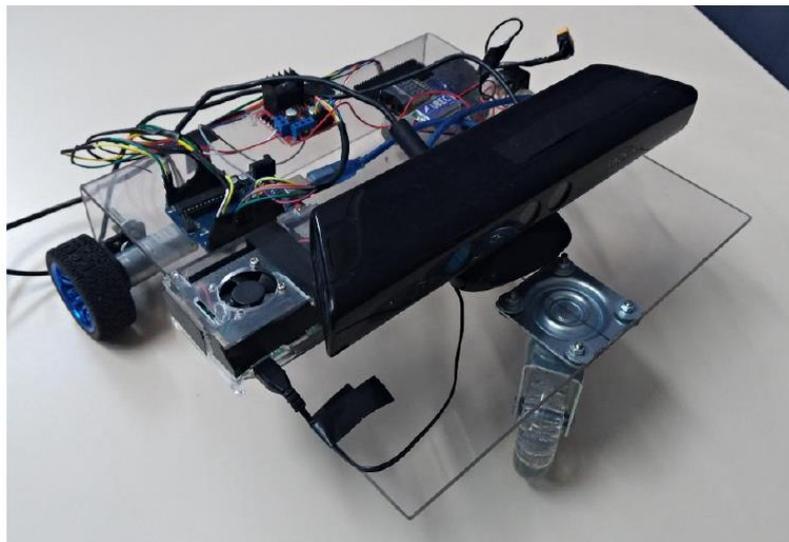


Fonte: Extraído e adaptado de Matsumura, 2014.

Em Barbosa (2017) foi desenvolvido um robô de exploração controlado remotamente, e que dispunha de uma câmera para transmissão de imagens em tempo real. O foco do projeto foi desenvolver a parte mecânica e estudar principalmente a parte de amortecimentos contra impactos. Contudo, o projeto usou também um Arduíno para controlar os motores. Como o autor estava se formando em Engenharia Mecânica o objetivo maior foi estudar os efeitos e potenciais soluções sobre os problemas de impactos nas partes mecânicas robô e na dinâmica de movimento das rodas sobre diferentes tipos de pisos.

Já em Daniluski & Rosa (2018), os autores, que foram alunos da 1ª turma do curso de Mecatrônica Industrial da Fatec Santo André, desenvolveram um robô mapeador de ambientes. Esse robô foi construído sobre uma plataforma em acrílico, com dois motores de tração e uma roda louca. O objetivo do trabalho foi implementar um sistema de visão 3D usando uma câmera Kinetic de um Xbox da Microsoft. Foram instalados um Arduíno para controlar os motores e um Raspberry Pi 3 para processar as imagens vindas das câmeras do Kinetic. Objetivo era mapear o ambiente e determinar os espaços sem obstáculos que permitiriam a movimentação do robô.

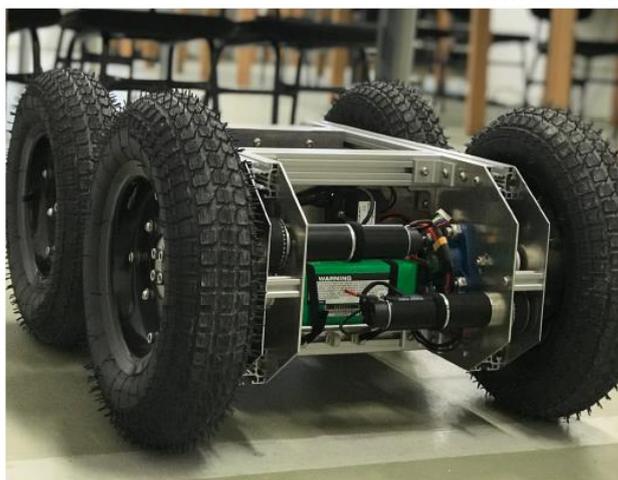
Figura 4. Robô mapeador de ambientes usando uma câmera Kinetic e um Arduino para controlar dois motores.



Fonte: Extraído de Daniluski & Rosa, 2018.

Finalmente, dentre as pesquisas realizadas para este projeto, o trabalho que mais se aproxima dos objetivos deste TCC foi apresentado por Buzzi (2019) na qual o autor desenvolveu uma plataforma robótica com integração via ROS-1. A estrutura mecânica foi desenvolvida com perfis de alumínio e contava com quatro motores, um em cada roda. O robô tinha capacidade de carga total de 50 kg e foi testado em várias condições de terreno e rampas. O dispositivo tinha um LiDAR 2D de 360 graus e uma câmera e o processamento ficou a cargo de um computador Intel NUC rodando Linux.

Figura 5. Na figura (a) uma vista interna do robô proposto por Buzzi, 2019, e em (b) o mesmo robô com a carenagem.



(a) Robô montado sem a carenagem.



(b) [Robô montado com a carenagem.

Fonte: Extraído de Buzzi, 2019.

O campo da robótica móvel sempre atraiu a atenção dos pesquisadores e, naturalmente, muitas pesquisas começaram nas universidades. A *Clearpath Robotics* é um exemplo de empresa que nasceu a partir de quatro amigos de uma universidade Canadense e que gostavam de robôs.

Figura 6 – Robô sobre rodas Husky desenvolvido pela Clearpath para aplicações gerais.



Fonte: [www.clearpathrobotics.com](http://www.clearpathrobotics.com) [09].

Eles fundaram a empresa justamente para desenvolver plataformas robóticas para permitir que outros pesquisadores avançassem nas pesquisas, mas sem consumir tempo no desenvolvimento das estruturas mecânicas e de controle.

Figura 7 – Grupo de engenheiros da Clearpath montando os sensores no corpo do robô Husky.



Fonte: [www.clearpathrobotics.com](http://www.clearpathrobotics.com)

Juntando a robótica com a agronomia, teve-se a ideia de criar o robô batizado de Contadino, um robô criado e apresentado pela Continental em 2019. É um robô totalmente autônomo que pode ser usado para tarefas leves, como: semear, capinar, pulverizar, fertilizar e monitorar. Em seu corpo tem um espaço para encaixe de diversos tipos de ferramentas que podem ser utilizadas em diferentes processos. Cada ferramenta leva consigo um tipo de programação, com isso o robô consegue realizar diferentes funções trocando apenas a ferramenta acoplada. Para que sua operação tenha menos interferência humana possível, o robô é equipado com sensores ultrassônicos, radar, RTK GPS e câmera, podendo identificar os diferentes tipos de terreno e irregularidades.

Figura 8 – Robô Contadino em seu ambiente de trabalho mostrando o uso de um de seus sensores.



Fonte: [www.continental.com](http://www.continental.com)

Nos tempos atuais, ano de 2023, onde a modernidade e o avanço da tecnologia têm tomado conta de tudo, as pessoas não querem mais ir atrás dos produtos, querem que os produtos cheguem até elas, com base nisso a Amazon desenvolveu o SCOUT.

O SCOUT é um robô autônomo desenvolvido pela Amazon para fazer entregas de pacotes em áreas residenciais. Ele é projetado para se mover de forma autônoma nas calçadas e ruas, sem a necessidade de um motorista ou entregador humano. Ele pode viajar em velocidades de até 6 km/h, e entrega os pacotes diretamente nas portas dos clientes. O robô tem a aparência de um pequeno carrinho de compras com seis rodas e é equipado com sensores de detecção de luz, sensores de distância por laser e câmeras para detectar e evitar obstáculos, como pedestres, cães e outros objetos em seu caminho.

Ainda tem um compartimento com tampa que pode armazenar vários pacotes e pode ser destravado por um código enviado ao cliente. Depois de entregar o pacote, ele retorna ao armazém ou centro de distribuição para recarregar suas baterias e estar pronto para outra entrega.

Atualmente, o SCOUT está em fase de testes em algumas áreas dos Estados Unidos, com a intenção de melhorar a eficiência das entregas e reduzir o tempo de espera dos clientes. A *Amazon* acredita que o uso de robôs autônomos pode ajudar a resolver problemas de logística e aumentar a velocidade e eficiência das entregas.

Figura 9 - Robô SCOUT em frente à casa de um cliente para mostrar que consegue chegar até o endereço desejado.



Fonte: [www.olhardigital.com.br](http://www.olhardigital.com.br)

No mundo moderno a tecnologia não fica presa somente em indústria ou em empresas conhecidas mundialmente, ela se espalha por todas as áreas para um benefício mútuo da sociedade, com isso foi criado o *Trombia Free*.

O Trombia Free é um veículo autônomo desenvolvido pela empresa finlandesa *Trombia Technologies* para a limpeza de ruas e calçadas. Ele é projetado para remover detritos e resíduos, como folhas, galhos e lixo, sem a necessidade de um motorista humano.

O veículo é equipado com um sistema de varredura que usa sensores para detectar e mapear o ambiente, permitindo que ele navegue de forma autônoma. Ele também tem um sistema de aspiração que pode sugar detritos e deixá-los em um compartimento de armazenamento interno. O Trombia Free é alimentado por baterias elétricas e pode operar por até 12 horas antes de precisar ser recarregado. Ele é projetado para ser mais eficiente e ecológico do que os veículos de limpeza

tradicionais, que geralmente são movidos a diesel e exigem um motorista para operar. Ele é capaz de limpar uma área maior em menos tempo e sem gerar emissões poluentes.

O veículo foi testado em várias cidades em todo o mundo, incluindo Dubai, onde foi usado para limpar as ruas após uma tempestade de areia, e Helsinque, onde foi usado para remover neve e gelo das ruas durante o inverno. O Trombia Free é um exemplo de como a tecnologia autônoma pode ser usada para melhorar a eficiência e sustentabilidade de serviços de limpeza urbana.

Figura 10 - Robô TROMBIA FREE realizando limpeza urbana no meio do dia.



Fonte: [www.pplware.sapo.pt](http://www.pplware.sapo.pt)

## 2.2. Considerações Finais

Este capítulo buscar apresentar de maneira sucinta o desenvolvimento de robôs móveis sobre rodas dentro das universidades e na indústria. O objetivo foi buscar referências de trabalhos anteriores para formar a base deste projeto. Notou-se que somente um trabalho se aproxima da proposta de ser uma base robótica de aplicações gerais e que possa ser utilizado por outros trabalhos e pesquisas futuras. A pesquisa com as empresas que produzem robôs comerciais mostrou também que é bastante limitado o oferecimento de plataformas universais e de baixo custo. A maioria está voltado para alguma aplicação específica.

### **3. Metodologia e Desenvolvimento do Projeto**

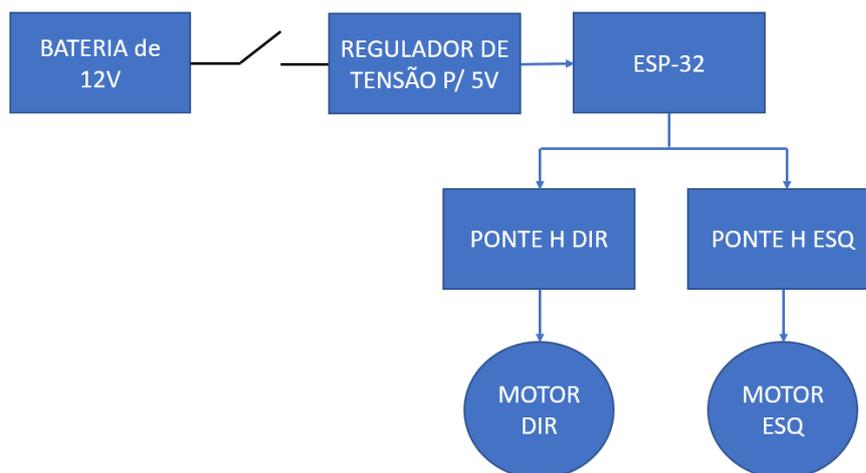
Neste capítulo será apresentado o desenvolvimento da plataforma robótica. Contudo, o projeto da Plataforma Universal de Robô Móvel sobre Rodas Controlada Remotamente não terá a mesma estrutura que RHINO (BUHMANN, et al.,2000), já que ele é relativamente grande e sua função principal é guiar e interagir as pessoas, mas contará com os mesmos componentes robóticos para que a movimentação possa ser feita de uma maneira livre de controles.

Um dos objetivos propostos para este projeto, é de que ele possa ser utilizado em diversas áreas, tanto da indústria quanto em outras partes do mercado em geral. Uma das áreas que mais cresceram nos últimos anos foi a área da agronomia, um dos setores que mais recebe investimentos de tecnologia em todo o mundo. No Brasil, o setor de agronomia é a segunda maior fonte de arrecadação, onde desde 2019 tem uma participação maior do que 20% no PIB do país.

#### **3.1 Arquitetura do Projeto**

A figura abaixo ilustra a arquitetura simplificada do controle do robô, onde a bateria alimenta as Pontes H, que por sua vez alimentam os motores esquerdo e direito. A bateria também alimenta um circuito que foi montado para reduzir a tensão para 5V, que é a tensão ideal para o ESP-32, o ESP-32 por sua vez se conecta ao outro lado das Pontes H pois assim consegue transmitir os sinais PWM para controlar os motores. As Pontes H também são conectadas ao circuito dedicado para redução da tensão.

Figura 11 – Arquitetura simplificada da montagem dos componentes dentro do robô plataforma.



Fonte: Do próprio autor, 2023.

### 3.2. ESP-32

O ESP-32 é considerado um microcontrolador inovador e um dispositivo essencial para o desenvolvimento de projetos automatizados. Foi desenvolvido pela empresa Espressif e colocado no mercado em 2016. Esse componente demonstra ter mais utilidades e ser mais versátil do que seu antecessor, o microcontrolador ESP8266, já que, além do modo clássico de comunicação através do Wi-Fi, conta com a presença de um sistema com processador Dual Core, múltiplos sensores embutidos e Bluetooth híbrido, e tudo isso acaba simplificando a construção de sistemas e deixando tudo mais compacto.

Com essa tecnologia é possível diversificar as operações realizadas pelo IoT (Internet das Coisas), que é definido como qualquer objeto que possua um sistema eletrônico embutido, capaz de receber e transmitir informações da rede. Por ser controlada pela internet e não precisar que tenham pessoas mexendo em seu programa, é considerada uma automação sem fio.

Quadro 1 - Tabela comparando os recursos disponíveis no ESP8266 e no ESP-32.

Especificações	ESP8266	ESP32
MCU	xtensa® single 32-bit l106	xtensa® dual-core 32-bit lx6 600 DMIPS
802.11 b/g/n Wi-Fi	HT20	HT40
Bluetooth	Não	Bluetooth 4.2 Le
Frequência	80 MHz	160 MHz
SRAM	160 Kbytes	512 Kbytes
Flash	SPI Flash, 16 Mbytes	SPI Flash, 16 Mbytes
GPIO	17	36
Hardware/ Software PWM	Não/ 8 canais	1 / 16 canais
SPI/ I2C/I2S/UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	Não	1
Interface Ethernet Mac	Não	1
Sensor Capacitivo	Não	Sim
Sensor de Temperatura	Não	Sim
Temperatura de Trabalho	-40° C a 125° C	-40° C a 125° C

Fonte: [www.curtocircuito.com.br](http://www.curtocircuito.com.br).

### 3.3. Aplicações

Os equipamentos da linha ESP-32 possuem várias aplicabilidades, baseados em RTOS, podem ser usados em praticamente qualquer projeto de prototipagem, IoT e automação, sendo IoT a principal. Todos os componentes desta família são capazes de se comunicar com outros dispositivos pelos meios de Internet, dependendo do modelo essa comunicação também pode ser feita por Bluetooth. Nas aplicações que necessitam de uma comunicação *wireless* são onde o ESP se destaca.

Estes microcontroladores são compactos, montado numa placa desenvolvimento que tem uma área de 52x28 mm e um processador dual core com até 240 MHz de velocidade, podendo variar de acordo com o modelo escolhido. Podem suportar uma grande variação de temperatura, desde -40°C até 125°C, o que permite que sejam usados em diversos ambientes.

Pode-se dizer que a família ESP é direcionada para projetos que necessitam de comunicação com a internet, ainda mais em ambientes que dificultam o acesso da tecnologia, como espaços pequenos e temperaturas extremas. Não se limitam apenas

a operações com essas características, dependendo da situação é aconselhável que seja usado um ESP no lugar de outros componentes, por exemplo quando um Arduino não é o suficiente para o andamento do projeto e um *Raspberry Pi* é considerado um exagero para a aplicação, o ESP-32 está na média.

### 3.4. Estrutura de Pinagem do ESP-32

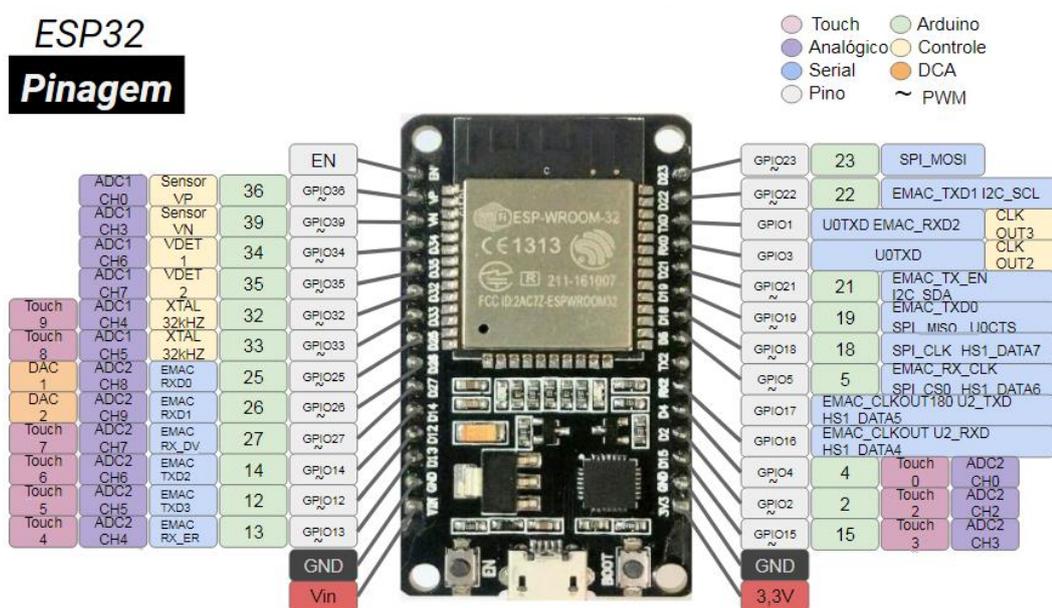
Figura 12 - Imagem de um ESP-32.



Fonte: [www.saravati.com.br](http://www.saravati.com.br).

Figura 13 - Imagem que representa as funções de cada pino do ESP-32.

Ao analisar a estrutura da pinagem de um ESP-32 é possível identificar vários pinos com recurso para PWM, 10 pinos de leitura capacitiva, 3 UART, 18 canais ADC, 2 I2S, 3 SPI e 34 portas GPIO.



Fonte: [www.curtocircuito.com.br](http://www.curtocircuito.com.br)

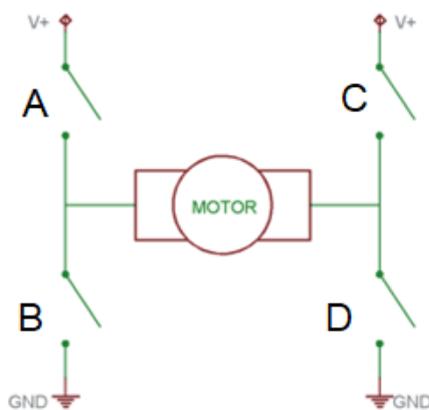
### 3.5. Ponte H

Uma Ponte H é um circuito eletrônico usado para controlar a direção e a velocidade de motores elétricos, especialmente em robótica e outros projetos eletrônicos. A Ponte H é composta por quatro transistores ou dispositivos de comutação, geralmente dispostos em uma configuração de arranjo em ponte, daí o nome "Ponte H".

A função da Ponte H é permitir que o motor elétrico seja controlado em ambas as direções (horário e anti-horário) e em diferentes velocidades. Isso é possível porque a Ponte H é capaz de inverter a polaridade da corrente elétrica que flui pelo motor, permitindo que ele gire em direções opostas. A Ponte H também é capaz de regular a quantidade de corrente que flui pelo motor, permitindo que ele seja controlado em diferentes velocidades, através do recurso de PWM vindo de um microcontrolador.

A Ponte H é amplamente utilizada em projetos de robótica, como robôs móveis e braços robóticos, bem como em outras aplicações que exigem o controle preciso da velocidade e direção de motores elétricos. Alguns exemplos de aplicações incluem automação industrial, sistemas de transporte e elevadores.

Figura 14 - Desenho eletrônico de uma Ponte H e seus contatos.

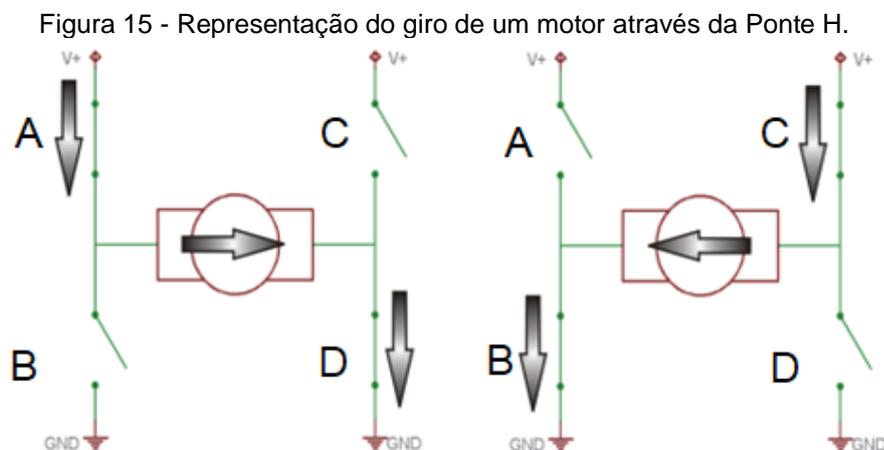


Fonte: PATSKO (2006, p.2).

### 3.7. Controle de Rotação e Direção

Para o motor elétrico funcionar é necessário fechar dois contatos, dependendo da direção que se deseja girar. Por exemplo, se fecharmos os contatos A e D, o motor

irá rodar no sentido horário, e se abirmos os contatos A e D e fecharmos os contatos B e C, o motor irá girar no sentido anti-horário. Assim, o sentido de giro do motor é apenas uma questão que fechar adequadamente as chaves da Ponte H.

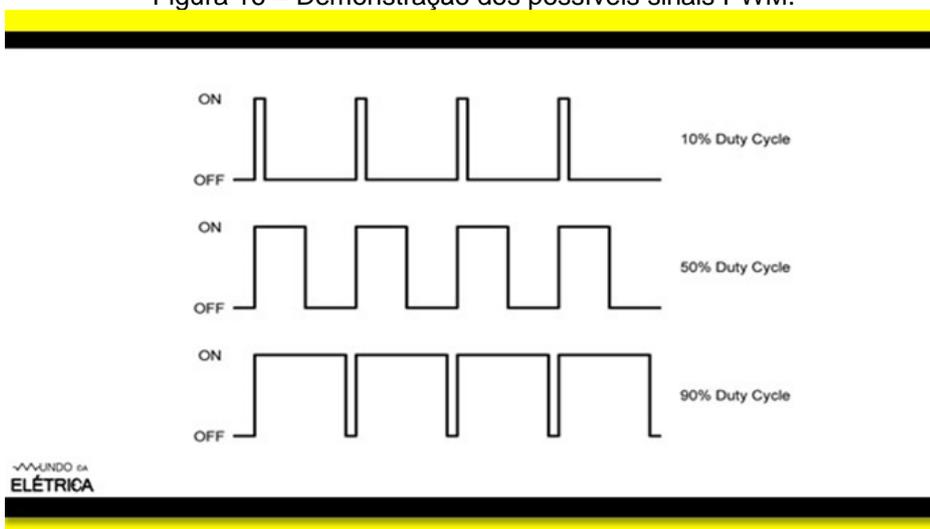


Fonte: PATSKO (2006, p.2).

### 3.8. Controle de Velocidade

Para a Ponte H controlar a velocidade do motor usa-se um sinal de PWM, que é enviado pelo microcontrolador. Esse tipo de sinal consiste em emitir uma série de pulsos com uma determinada largura em um período. Se a largura do pulso em nível alto for igual ao período do sinal, o motor irá girar na sua velocidade máxima. Entretanto, se a largura do pulso for menor do que o período, a velocidade do motor será proporcional a largura do pulso.

Figura 16 – Demonstração dos possíveis sinais PWM.



Fonte: [www.mundodaeletrica.com.br](http://www.mundodaeletrica.com.br)

### 3.9. Desenvolvimento do robô proposto

Primeiramente, o ESP-32 é um microcontrolador de baixo custo e alta performance, amplamente utilizado no desenvolvimento de projetos de IoT e robótica. Ele possui diversos recursos, como conectividade Wi-Fi e Bluetooth, capacidade de processamento e armazenamento de dados, além de ser facilmente programável através do ambiente de desenvolvimento Arduino.

Os componentes eletrônicos, como Ponte H, atuadores e módulos de comunicação, foram escolhidos de acordo com as necessidades do projeto. Por exemplo, para um robô totalmente autônomo, seria necessário utilizar sensores de distância para evitar obstáculos ou um módulo de GPS para navegação, neste caso o controle é feito através do *software* desenvolvido para isso.

Durante as pesquisas sobre diferentes plataformas robóticas, observou o uso de diferentes motores elétricos com redutores. Contudo, ou eram muito pequenos ou eram muito caros. O desafio era encontrar um motor elétrico com alta capacidade de torque, baixo peso e pequenas dimensões e com um custo razoável para o projeto. A solução encontrada foi usar uma furadeira / parafusadeira de mercado, que além do motor elétrico DC, já vem com um redutor planetário e um mandril para tracionar o eixo. Essa solução se mostrou bastante criativa, pois resolve uma série de problemas mecânicos e ainda mantém o custo baixo.

No entanto, é importante ressaltar que a furadeira não foi projetada para funcionar como motor de um robô, e pode apresentar limitações em relação a torque e velocidade, além de consumir mais energia do que um motor dedicado. A bateria deve ser escolhida de acordo com as especificações do projeto, considerando a capacidade de carga do robô plataforma e as tensões necessárias para alimentar todos os componentes eletrônicos.

Em resumo, o desenvolvimento de um robô usando o ESP-32 como controlador, componentes eletrônicos, furadeira como motor e uma bateria foi uma tarefa desafiadora, mas viável. É importante escolher os componentes corretos e dimensioná-los adequadamente para garantir o funcionamento adequado do robô.

### 3.10. Chassi de Madeira

Embora um chassi de madeira possa ser uma opção viável em alguns casos, geralmente ele não é tão útil quanto outros materiais mais adequados para a construção de robôs e outros projetos eletrônicos. Nos apêndices o leitor encontrará os desenhos mecânicos com as dimensões para construção do robô plataforma. Uma das principais razões pelas quais um chassi de madeira pode não ser tão útil é devido à sua resistência e durabilidade limitadas em comparação a outros materiais, como plástico, metal ou fibra de carbono. A madeira pode ser danificada facilmente por impactos e vibrações, o que pode levar a uma falha de algum componente ou até mesmo danificar o próprio chassi do projeto e tem a chance de uma falha completa no funcionamento do robô.

Contudo, neste projeto foi utilizado a solução de um chassi em MDF devido aos altos custo para construção de um chassi em alumínio e a verba utilizada até o momento não permitia construção em metal.

Figura 17 – Imagem do corpo do robô plataforma montado.



Fonte: Do próprio autor, 2023.

Além disso, a madeira pode ser suscetível a mudanças de temperatura e umidade, o que pode afetar suas propriedades mecânicas. Isso pode resultar em deformações e mudanças no tamanho e na forma do chassi, o que pode causar problemas de encaixe, de funcionamento e até mesmo de estabilidade.

A madeira pode não ser tão adequada para a integração com outros componentes eletrônicos, como placas de circuito, sensores e atuadores. Outros materiais, como plástico ou metal, podem oferecer uma melhor compatibilidade e suporte para esses componentes, o que pode tornar mais fácil a construção e a manutenção do projeto. Nessa hora é necessário levar em conta a durabilidade, resistência, peso, absorção da umidade e a conectividade e fixação dos componentes.

Em resumo, enquanto a madeira pode ser uma opção viável para alguns projetos, ela geralmente não é tão útil quanto outros materiais mais adequados para a construção de robôs e outros projetos eletrônicos, devido à sua resistência e durabilidade limitadas, susceptibilidade a mudanças de temperatura e umidade, dificuldade de trabalho e menor compatibilidade com outros componentes eletrônicos.

Embora a madeira apresente alguns fatores que tornam sua escolha menos atraente, no nosso caso a madeira se tornou uma opção interessante por conta dos seguintes fatores:

- **Disponibilidade:** é um material relativamente fácil de encontrar e tem um custo mais acessível em comparação com outros materiais utilizados na construção do chassi.
- **Facilidade:** A madeira é mais fácil de trabalhar do que outros materiais, como alumínio ou fibra de carbono. Isso pode tornar a construção de um chassi de madeira menos complicada.
- **Personalização:** A madeira pode ser facilmente cortada em diversos tamanhos, podendo reaproveitar algumas partes que em outros casos teria que ser descartada. Isso torna o dimensionamento do robô plataforma mais simples, já que podemos alterar facilmente algumas peças.

### 3.11. Motor e Redutor

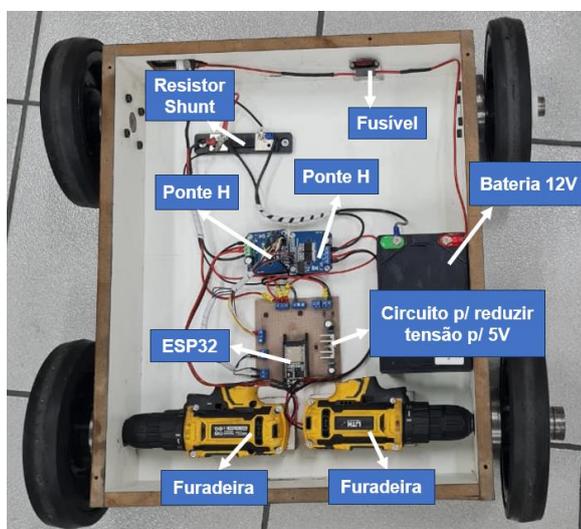
Para esse projeto foi definido o uso de dois motores para mover cada uma das rodas traseiras do robô, a escolha por dois motores se deu pelo fato de que teremos mais força e com isso o robô será capaz de andar com mais peso do que conseguiria com apenas um motor, além disso é possível fazer o robô virar para direita ou para esquerda. Esse controle do sentido de curva é feito da seguinte forma para virar para a direita travamos o motor do lado esquerdo enquanto o motor do lado direito gira

livremente e para virar para esquerda é só travar o motor do lado direito e girar o motor do lado esquerdo.

Para esse projeto escolhemos a furadeira LITH-LT7035 para ser usada como motor. A furadeira possui as seguintes especificações:

- Tensão da bateria: 12V;
- Corrente da bateria: 1,5Ah;
- Rotação de: 0 a 1350 RPM;
- Torque máximo: 24Nm;
- Mandril: 10mm.

Figura 18 - Imagem para ver a parte interna do robô plataforma com os componentes internos montados.



Fonte: Do próprio autor, 2023.

Figura 19 - Imagem para mostrar a face do robô.



Fonte: Do próprio autor, 2023.

### 3.12. Instalação das Furadeiras

As paredes laterais do chassi possuem 102 mm de altura e a Furadeira possui uma largura de 51 mm no ponto mais largo, logo se faz necessário a utilização de um calço com espessura de 15 mm para que a roda que será presa na Furadeira fique centralizada na lateral do robô. Para conseguir apoiar a Furadeira de forma adequada o calço foi feito de madeira e com as medidas de 60x120 mm. Com o calço pronto, ele foi parafusado na base do robô.

Como podemos ver na figura 20, a Furadeira possui seis furos (não passantes) que estão os parafusos que mantem os dois lados do equipamento unidos, quatro desses seis furos foram aumentados para 3 mm de diâmetro e se tornaram passantes, isso foi feito para que fosse possível prender as furadeiras nos calços através de quatro parafusos de madeira em cada furadeira.

Para transformar uma furadeira em um motor do robô plataforma nós tivemos que adaptá-la para que pudesse fornecer energia para as rodas. O método utilizado foi usinar uma ponta sextavado, que garante uma excelente tração se os inconvenientes de um eixo redondo, que pode escorregar. Esses eixos estão parafusados nas rodas, de um jeito que o movimento rotativo da Furadeira é transferido diretamente para as elas. Também foi executado uma adaptação para que a Furadeira seja acionada por uma bateria externa, tendo assim uma fonte de energia para o robô plataforma. Nesse caso, foi necessário construir um sistema de controle de velocidade para a Furadeira utilizando o ESP-32, isso possibilitou a criação de um programa para controlar o robô plataforma com mais precisão.

Figura 20 – Modelo da Furadeira utilizada no projeto. Em vermelho os furos de fixação do corpo da furadeira.

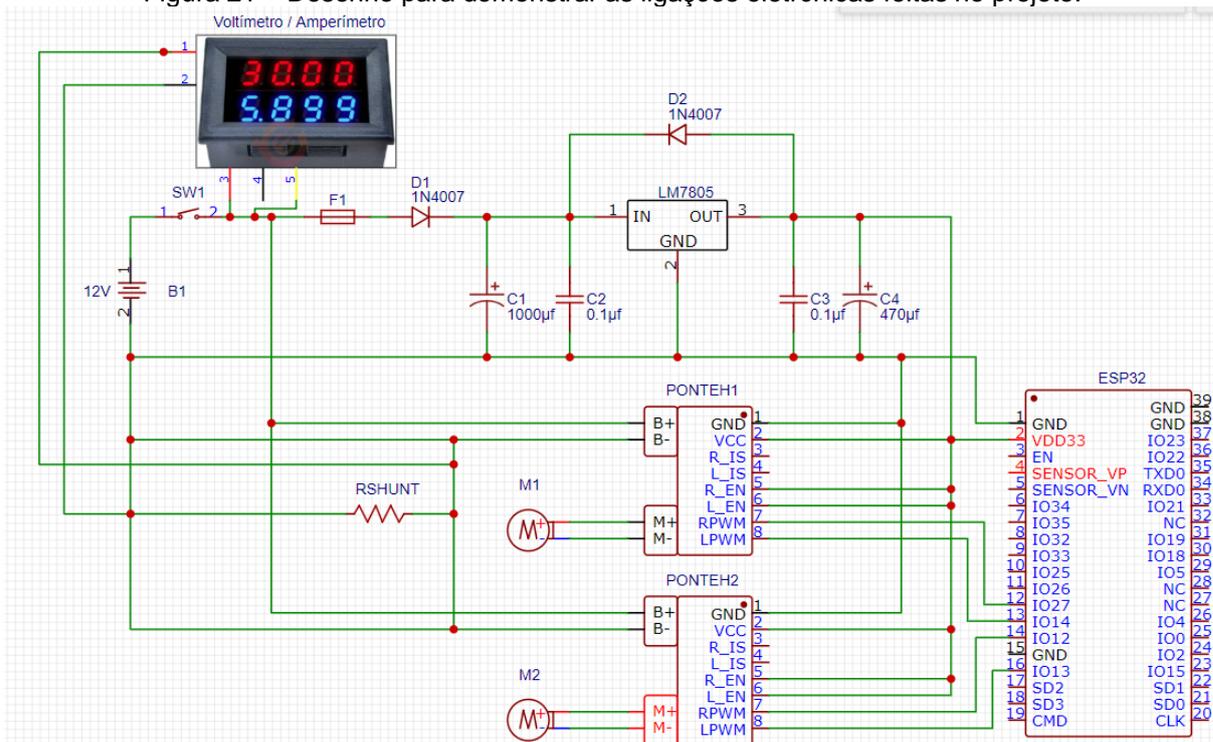


Fonte: [www.mixxferramentas.com](http://www.mixxferramentas.com)

### 3.13. Desenvolvimento do Circuito Eletrônico

Para que o robô fosse alimentado adequadamente foi desenvolvido um circuito eletrônico para alimentar o ESP-32, os dois motores DC e as duas Pontes H BTS90. Porém, os dois circuitos trabalham com tensões diferentes. Logo, é necessário fornecer duas tensões. O ESP-32 precisa de 5V para funcionar e os motores precisam de uma tensão de 12V, enquanto a Ponte H tem dois circuitos, um alimentado por 5V e o outro por 12V. Instalamos um instrumento para medição de tensão e corrente elétrica da bateria, sendo um voltímetro e amperímetro.

Figura 21 – Desenho para demonstrar as ligações eletrônicas feitas no projeto.



Fonte: Do próprio autor, 2023.

O motor é conectado em M+ e M-, enquanto o positivo e negativo da bateria são conectados respectivamente em B+ e B-.

Figura 22 – Imagem para mostrar os pontos de conexão da Ponte H BTS7960 com o motor e a fonte.



Fonte: Do próprio autor, 2023.

Para reduzir a tensão de 12V para 5V utilizamos o regulador de tensão LM7805 e para garantir que a tensão chegue constante foram colocados capacitores de 1000µf/16V e 0,1µf/50V na entrada do regulador, e para que a tensão que sai do regulador de tensão chegue constante no ESP-32 foram colocados capacitores de

0,1 $\mu$ f / 50V e 470 $\mu$ f / 16V. O ESP-32 é energizado conectando o positivo e o negativo respectivamente nos pinos 3.3 V e GND.

Para ativar o circuito de 5 V da Ponte H é preciso que os pinos VCC, R\_EN e L\_EN sejam conectados no positivo do circuito e o pino GND no negativo do circuito, isso nas duas Pontes H. E finalmente, os pinos 12, 13, 14 e 27 do ESP-32 são conectados nos pinos RPWM e LPWM das Pontes H, dois pinos do ESP-32 por Ponte H.

Figura 23 – Imagem para mostrar os pinos da Ponte H.



Fonte: Do próprio autor, 2023.

### 3.14. Desenvolvimento do Software de Controle

Um programa em Python foi desenvolvido na IDE Thonny Python para controlar o robô plataforma para realizar testes. Na primeira parte do código foram importadas as seguintes bibliotecas: Usocket (ou socket), Pin, PWM, network, esp e gc.

A biblioteca Usocket (ou socket) é responsável por criar soquetes, que são canais de comunicação, usados para enviar ou receber dados em uma rede. No código fonte em Python, a cláusula Try-Except é usada para tentar importar primeiro a biblioteca Usocket e, se isso não for possível, importar a biblioteca de socket padrão como alternativa. No pacote machine temos as bibliotecas Pin e PWM. A primeira é usada para controlar os pinos de entrada/saída (I/O) do microcontrolador. Já na segunda biblioteca temos a função para controlar o sentido e a velocidade dos motores DC com o uso de sinais PWM.

A biblioteca Network fornece funções para configurar e gerenciar conexões de rede em dispositivos microcontroladores. A biblioteca GC é usada para coletar lixo e

no código a coleta completa de lixo está ativada. Por fim, a biblioteca ESP disponibiliza funções relacionadas à configuração e gerenciamento do próprio *chip* ESP-32. No código, o gerenciamento do ESP-32 está sendo usada para desativar as mensagens de depuração.

Figura 24 - Demonstração das bibliotecas utilizadas no desenvolvimento do programa.

```

1  try:
2      import usocket as socket
3  except:
4      import socket
5
6  from machine import Pin, PWM
7  import network
8  import esp
9
10 esp.osdebug(None)
11
12 import gc
13 gc.collect()
14

```

Fonte: Do próprio autor, 2023.

Um ponto de acesso é criado no ESP-32 com o nome "TCC Robo", utilizando a biblioteca Network. A rede Wi-Fi é ativada e quando isso ocorre, a mensagem "Connection successful" é exibida para confirmar a sua ativação.

Figura 25 - Ilustração do ponto de acesso criado no ESP-32.

```

15 ssid = 'TCC Robo'
16
17 ap = network.WLAN(network.AP_IF)
18 ap.active(True)
19 ap.config(essid=ssid)
20
21 while ap.active() == False:
22     pass
23
24 print('Connection successful')
25 print(ap.ifconfig())
26

```

Fonte: Do próprio autor, 2023.

Os pinos dos motores são definidos, a variável velocidade é criada e recebe o valor "0", pois ao iniciar o robô precisa estar parado.

Figura 26 - Definição dos pinos dos motores.

```
27  
28 esquerda1 = PWM(Pin(12))  
29 esquerda2 = PWM(Pin(13))  
30 direita1 = PWM(Pin(14))  
31 direita2 = PWM(Pin(27))  
32  
33 velocidade = 0  
34
```

Fonte: Do próprio autor, 2023.

Temos a criação da função "*movimentar\_robô*" que recebe dois argumentos: "movimento" e "vel". O primeiro recebe uma string e é utilizado para definir o tipo de movimento desejado e, com isso, ativar os motores a fim de realizar esse movimento. O segundo recebe um valor PWM (0 a 1023) que define a velocidade com a qual o motor irá girar.

Figura 27 - Função criada para movimentar o Robô Plataforma.

```
35 def movimentar_roboto(movimento, vel):
36     if movimento == 'andar frente':
37         esquerda1.duty(vel)
38         esquerda2.duty(0)
39         direita1.duty(vel)
40         direita2.duty(0)
41     if movimento == 'andar tras':
42         esquerda1.duty(0)
43         esquerda2.duty(vel)
44         direita1.duty(0)
45         direita2.duty(vel)
46     if movimento == 'virar direita':
47         esquerda1.duty(0)
48         esquerda2.duty(0)
49         direita1.duty(vel)
50         direita2.duty(0)
51     if movimento == 'virar esquerda':
52         esquerda1.duty(vel)
53         esquerda2.duty(0)
54         direita1.duty(0)
55         direita2.duty(0)
56     if movimento == 'parar':
57         esquerda1.duty(0)
58         esquerda2.duty(0)
59         direita1.duty(0)
60         direita2.duty(0)
```

Fonte: Do próprio autor, 2023.

Figura 28 - Parte do programa que representa a página html.

```

62 def web_page():|
63
64     html = ""
65     <html>
66         <head>
67             <title>Servidor Web ESP32</title>
68             <meta charset="UTF-8">
69             <meta name="viewport" content="width=device-width, initial-scale=1">
70             <link rel="icon" href="data:,">
71             <style>
72                 h1 {
73                     font-size: 60px; /* Tamanho original */
74                 }
75                 body {
76                     text-align: center;
77                 }
78                 .button {
79                     padding: 10px 20px;
80                     font-size: 16px;
81                     margin: 4px 4px;
82                 }
83                 .button1 {
84                     padding: 25px 25px;
85                     font-size: 36px;
86                     margin: 4px 40px;
87                     position: relative;
88                 }
89                 .button1::before{
90                     content: '';
91                     position: absolute;
92                     width: 0.4em;
93                     height: 0.4em;
94                     border: solid rgb(0, 0, 0);
95                     border-width: 0 0.1em 0.1em 0;
96                     transform: translate(-50%, -20%) rotate(225deg);
97                 }
98                 .button2::before{
99                     transform: translate(-30%, -50%) rotate(135deg);
100                }
101                .button3::before{
102                    transform: translate(-70%, -50%) rotate(315deg);
103                }
104                .button4::before{
105                    transform: translate(-50%, -80%) rotate(45deg);
106                }
107            </style>
108        </head>
109        <body>
110            <h1>Controlar Robô</h1>
111
112            <p><a href="/?velocidade=minima"><button class="button">MÍNIMA</button></a>
113            <a href="/?velocidade=media"><button class="button">MÉDIA</button></a>
114            <a href="/?velocidade=maxima"><button class="button">MÁXIMA</button></a></p><br>
115            <p><a href="/?andar=frente"><button class="button1 button1"></button></a></p>
116            <p><a href="/?virar=esquerda"><button class="button1 button2"></button></a>
117            <a href="/?virar=direita"><button class="button1 button3"></button></a></p>
118            <p><a href="/?andar=tras"><button class="button1 button4"></button></a></p><br>
119            <p><a href="/?parar"><button class="button">PARAR</button></a></p>
120        </body>
121    </html>""
122    return html
123

```

Fonte:

Do próprio autor, 2023.

Essa parte do código está usando a biblioteca socket para criar um servidor TCP básico que escuta conexões de entrada na porta 80 (usada para HTTP).

Figura 29 - Demonstração da criação de um novo ciclo para representar a conexão entre o servidor e o cliente.

```
124 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
125 s.bind(('', 80))
126 s.listen(5)
```

Fonte: Do próprio autor, 2023.

Um novo socket é criado assim que o ciclo *while* é iniciado para representar a conexão entre o servidor e o cliente. Os dados que serão enviados pelo cliente são definidos com o valor de 1024 bytes por vez e por fim os dados em bytes são convertidos em *string* utilizando a parte do código da figura 29.

Figura 30 – Forma com que os dados em bytes são convertidos em *string*.

```
128 while True:
129     conn, addr = s.accept()
130     print('Got a connection from %s' % str(addr))
131     request = conn.recv(1024)
132     request = str(request)
```

Fonte: Do próprio autor, 2023.

As variáveis que correspondem aos botões da página HTML são criadas. O método `.find()` é usado na variável *request* para procurar na página HTML a *string* que está entre parênteses.

Figura 31 - Criação dos botões que correspondem as variáveis.

```
134 vel_min = request.find('/?velocidade=minima')
135 vel_med = request.find('/?velocidade=media')
136 vel_max = request.find('/?velocidade=maxima')
137 andar_frente = request.find('/?andar=frente')
138 virar_esquerda = request.find('/?virar=esquerda')
139 virar_direita = request.find('/?virar=direita')
140 andar_tras = request.find('/?andar=tras')
141 parado = request.find('/?parar')
```

Fonte: Do próprio autor, 2023.

Quando os botões são apertados eles recebem um valor e quando esse valor corresponde ao que está na condição (if) a velocidade ou o movimento a ser realizado são definidos.

Figura 32 - Parte do programa que atribui valores aos botões.

```

142     if vel_min == 6:
143         velocidade = 120
144         print("velocidade: ", velocidade)
145     if vel_med == 6:
146         velocidade = 260
147         print("velocidade", velocidade)
148     if vel_max == 6:
149         velocidade = 400
150         print("velocidade", velocidade)
151     if andar_frente == 6:
152         movimentar_roboto('andar frente', velocidade)
153         print(velocidade)
154     if parado == 6:
155         movimentar_roboto('parar', velocidade)
156         velocidade=0
157         print(velocidade)
158     if andar_tras == 6:
159         movimentar_roboto('andar tras', velocidade)
160         print(velocidade)
161     if virar_direita == 6:
162         movimentar_roboto('virar direita', velocidade)
163         print(velocidade)
164     if virar_esquerda == 6:
165         movimentar_roboto('virar esquerda', velocidade)
166         print(velocidade)

```

Fonte: Do próprio autor, 2023.

A função "web\_page" é chamada para definir o que será enviado como resposta para o cliente e as próximas linhas definem o formato da resposta como HTML e as enviam para o cliente, após a resposta ser enviada o socket atual é fechado e outro é aberto no começo do *while*.

Figura 33 - Função que é chamada para definir a resposta do cliente.

```

168     response = web_page()
169     conn.send('HTTP/1.1 200 OK\n')
170     conn.send('Content-Type: text/html\n')
171     conn.send('Connection: close\n\n')

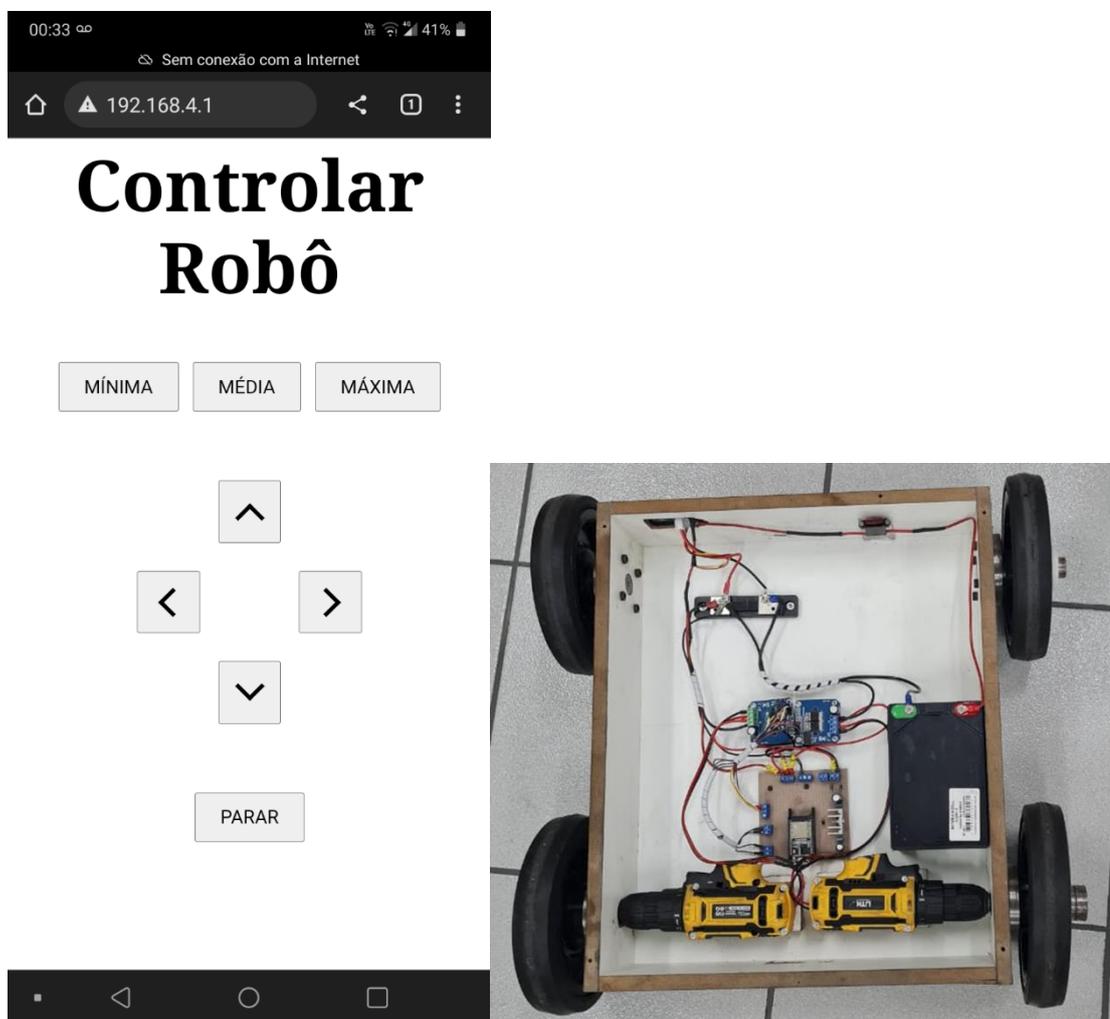
```

Fonte: Do próprio autor, 2023.

A página HTML gerada pelo código é constituída de três botões para controlar a velocidade da plataforma robótica móvel, de quatro botões para andar para frente, para trás, virar para direita, virar para esquerda e por último um botão para frear o

robô. É possível acessar a página HTML digitando “192.168.4.1” no navegador após se conectar à rede wi-fi do robô.

Figura 34 - Print da tela do celular com a página HTML aberta com o robô ao lado para demonstrar o sentido das setas em relação ao robô.



Fonte: Do próprio autor, 2023.

## 4. Testes e Análise dos Resultados

Alguns testes foram feitos para avaliar se o robô plataforma poderia atingir a potência nominal e para avaliar a eficiência do sistema mecânico juntamente do sistema eletrônico.

Foram definidos alguns parâmetros para a realização dos testes, como exemplo:

- Velocidade mínima, velocidade que faz o robô plataforma começar a se locomover;
- Velocidade máxima, maior velocidade que o conjunto mecânico suporta sem prejudicar suas partes.
- Pesos diferentes para testar a movimentação do robô, sendo eles apenas o peso do robô, 10 kg, 50 kg e 100 kg.
- Capacidade de realizar curvas.

Para a movimentação do robô plataforma foram definidas algumas direções de movimentação e níveis de potência. Para as direções definimos: para frente, para esquerda, para direita, para trás. E, uma opção de parar o robô plataforma de maneira instantânea.

Os testes em linha reta foram realizados em maio de 2023 na casa de um dos autores em um terreno regular e azulejado, mas com tamanho limitado fazendo com que não fosse possível realizar o teste de curvatura nesse local.

Figura 35 - Local do Teste em Linha Reta com Diferentes Pesos do Robô Plataforma.



Fonte: Do próprio autor, 2023.

Para realizar o teste com o peso de 10 kg foram utilizados dois sacos de arroz com 5 kg. Já para fazer o teste com o peso de 50 kg colocamos em cima do robô um saco de cimento de 50 kg. E para o teste com o peso de 100 kg utilizamos dois sacos de arroz de 5 kg, um saco de cimento de 50 kg e dois sacos de cimento de 20 kg.

Figura 36 - Robô Plataforma com os pesos de 10 kg e 50 kg em cima.



Fonte: Do próprio autor, 2023.

Figura 37 - Robô Plataforma com o peso de 100 kg em cima.



Fonte: Do próprio autor, 2023.

Para testar os regimes de velocidades definimos de acordo com a escala do PWM, que possui 1023 níveis, e que serviram de parametrização para definir as velocidades dos testes. A velocidade mínima foi colocada como 90, velocidade média como 260 e velocidade máxima como 400. A velocidade máxima (400) representa

cerca de 39,1% da potência máxima que poderia ser exercida sobre o robô plataforma, o projeto ficou limitado a esse nível por conta da parte de alguns limites da parte mecânica.

A velocidade mínima é capaz de mover apenas o peso do robô, e isso com certa dificuldade, sendo definida antes de trocar a bateria da furadeira por uma bateria mais potente e pesada. Portanto, a velocidade mínima não é a mesma que foi apresentada no início, já que o peso do robô plataforma aumentou por conta do peso extra da bateria. Sendo necessário mais potência para realização do movimento, assim, a velocidade mínima real é maior que a velocidade mínima nominal. Para corrigir esse problema a velocidade mínima foi alterada para 120, na escala PWM, após essa alteração o robô foi capaz de andar em todas as direções sem dificuldade isso em um terreno regular.

A velocidade média é capaz de andar em todas as direções definidas, isso apenas com o peso do próprio conjunto mecânico mais o elétrico. Se acrescentarmos um peso de 10 kg em cima do corpo do robô plataforma é possível perceber algumas diferenças, ele continua conseguindo realizar todos os movimentos, mas para virar é perceptível uma leve diferença em comparação quando não tem peso extra. Ao acrescentar uma carga de 50 kg o robô plataforma ainda consegue se movimentar para frente e para trás, mas virar para a esquerda e/ou direita não é mais possível. A limitação está por conta conjunto mecânico, que tem tração em apenas duas rodas.

Ao colocarmos na velocidade máxima obtivemos diferentes resultados, o robô plataforma ainda consegue realizar todos os movimentos quando está sem peso só que com uma velocidade bem maior, a resposta com as cargas de 10 kg e 50 kg foram as mesmas do teste com velocidade média, mas com um detalhe a mais, já que com o aumento da velocidade temos o aumento da potência, neste caso, na velocidade máxima é possível carregar uma carga de 100 kg sem dificuldade de se mover, mas apenas para frente ou para trás.

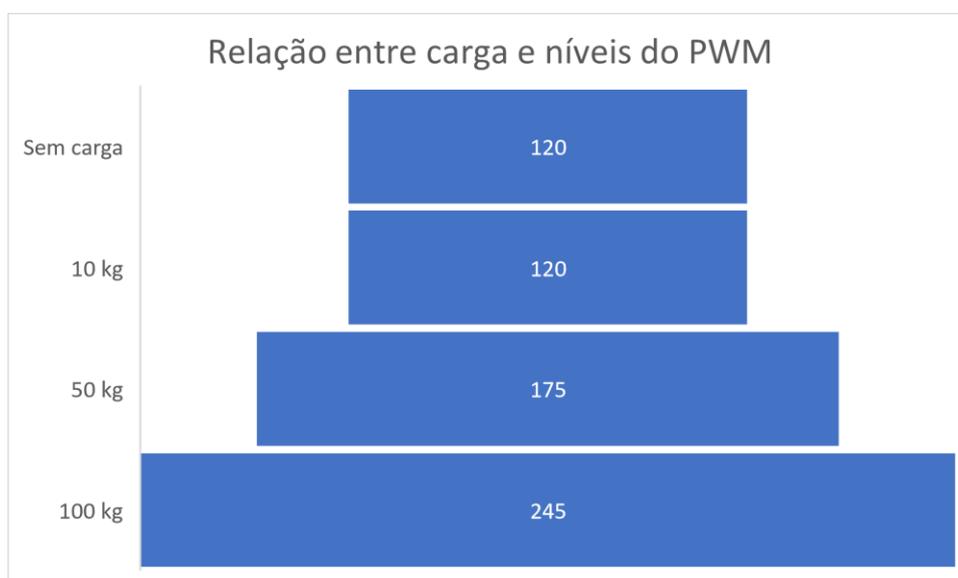
Em todos os testes realizados com os diferentes níveis de velocidade foi possível observar que o robô plataforma diminui sua efetividade ao colocarmos peso em cima. Seria um bom corpo para movimentação de peças leves que não ultrapassem o limite de 15 kg. Um problema observado e analisado é que as vezes, mesmo sem peso, o robô plataforma não vira para a direita ou para a esquerda, mesmo sem carga. Com base nas análises foi observado que esse problema acontece tanto por conta do

formato do corpo do robô, quanto pela possível distribuição desigual de peso, já que alguns componentes não estão fixados no corpo do robô plataforma, ficando soltos em seu interior e tendo algumas variações de posição.

Quadro 2 - Tabela dos resultados dos testes de carga e consumo de corrente dos motores na menor velocidade.

<b>Carga (kg)</b>	<b>PWM</b>	<b>Menor Velocidade (m/min)</b>	<b>Distância Percorrida (m)</b>	<b>Corrente de Consumo da bateria (A).</b>
Sem carga	120	16	4	0 – 0,11
10 kg	120	15	4	0 – 0,11
50 kg	175	11,6	4	0,34
100 kg	245	10,3	4	0,77

Figura 38 - Gráfico expressando a relação entre os níveis de PWM necessários para movimentação de acordo com a carga que o robô está carregando em sua menor velocidade.

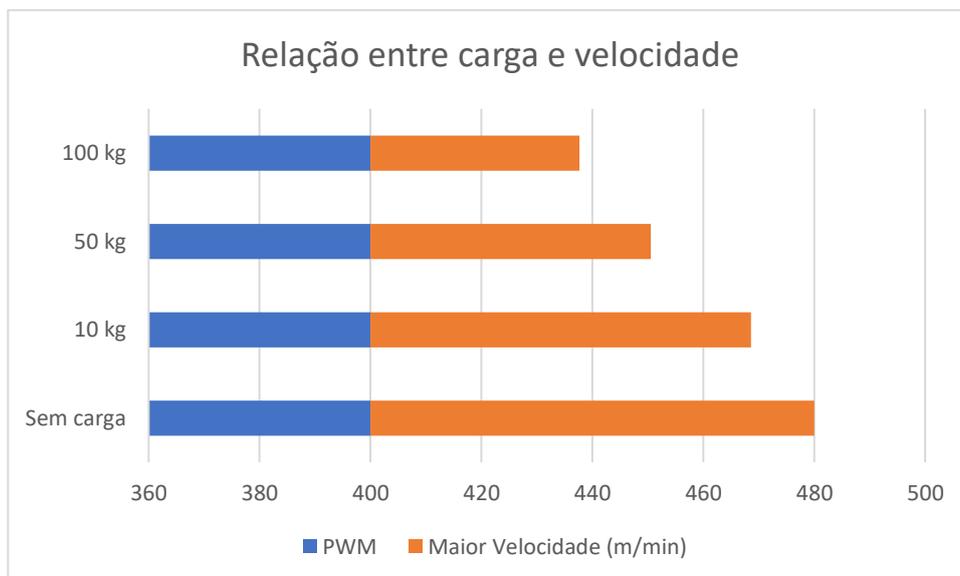


Fonte: Do próprio autor, 2023.

Quadro 3 - Tabela dos resultados dos testes de carga e consumo de corrente dos motores na maior velocidade.

<b>Carga (kg)</b>	<b>PWM</b>	<b>Maior Velocidade (m/min)</b>	<b>Distância Percorrida (m)</b>	<b>Corrente de Consumo da Bateria (A)</b>
Sem carga	400	80	8	0,45
10 kg	400	68,57	8	0,52
50 kg	400	50,53	8	0,73
100 kg	400	37,65	8	1,14

Figura 39 - Gráfico que mostra a relação da velocidade e da carga, quanto maior a carga menor a velocidade.



Fonte: Do próprio autor, 2023.

Na parte do consumo de corrente na tabela 1 a velocidade mínima com 10 kg e sem carga, constatou-se no amperímetro um valor de 0,00 A quase que constante, teve breves momentos que subiu para 110 mA, mas logo voltava para 0,00 A.

A escala PWM nessa aplicação vai de 0 a 1023, porém o maior valor usado foi de 400, isso se deve a limitação do conjunto físico que com velocidades maiores que essa teria problemas para manter as furadeiras fixas ao frear. Mas se mandássemos 1023 para os motores teríamos uma rotação de 1350 rpm (rotação nominal da furadeira), e como sabemos o diâmetro da roda que é 200 mm, podemos calcular a velocidade máxima que o robô pode atingir utilizando essas duas variáveis.

Para converter rpm em frequência só precisamos dividir o rpm por 60, logo temos que a frequência é:

$$f = \frac{1350}{60}$$

$$f = 22,5 \text{ Hz}$$

Com a frequência e o raio da roda podemos calcular a velocidade em m/s utilizando a seguinte fórmula:

$$v = 2\pi \times R \times f$$

$$v = 2\pi \times 0,1 \times 22,5$$

$$v = 14,137 \text{ m/s ou } 848,22 \text{ m/min}$$

Essa velocidade de 848,22 m/min é a velocidade máxima sem levar em conta o peso do robô, então a velocidade máxima real será um valor próximo disso, mas um pouco menor.

#### 4.1. Teste de Capacidade de Realização de Curvas

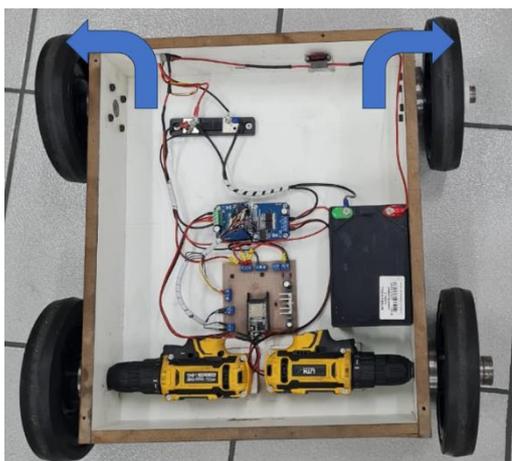
Os testes para executar curvas foram realizados em maio de 2023 em um terreno irregular. Foi utilizado o estacionamento da FATEC, que é asfaltado, mas não totalmente plano e regular.

Figura 40 - Local do Teste de Curvatura do Robô Plataforma.



Fonte: Do próprio autor, 2023.

Figura 41 - Representação do sentido de giro do robô no primeiro teste.



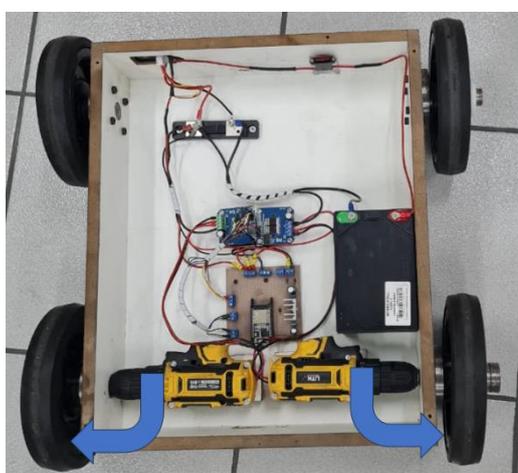
```

46  if movimento == 'virar direita':
47      esquerda1.duty(0)
48      esquerda2.duty(0)
49      direita1.duty(vel)
50      direita2.duty(0)
51  if movimento == 'virar esquerda':
52      esquerda1.duty(vel)
53      esquerda2.duty(0)
54      direita1.duty(0)
55      direita2.duty(0)

```

Fonte: Do próprio autor, 2023.

Figura 42 - Representação do sentido de giro do robô no segundo teste.



```

46  if movimento == 'virar direita':
47      esquerda1.duty(0)
48      esquerda2.duty(0)
49      direita1.duty(0)
50      direita2.duty(vel)
51  if movimento == 'virar esquerda':
52      esquerda1.duty(0)
53      esquerda2.duty(vel)
54      direita1.duty(0)
55      direita2.duty(0)

```

Fonte: Do próprio autor, 2023.

Quadro 4 - Tabela dos testes de curvatura do robô.

	<b>Esquerda (m)</b>	<b>Direita (m)</b>
Primeiro teste (figura 38)	13	27,5
Segundo Teste (figura 39)	9	12

As medições das curvaturas em metros podem apresentar um erro de mais ou menos 0,5 m, porque a trena não era grande o suficiente para medir as curvaturas de uma só vez. Por isso, foi necessário medir de 3 m em 3 m para chegar nos resultados

apontados na tabela 4. Logo, há erros nas medidas porque a trena pode ter sido recolocada em um ponto um pouco fora do ideal para prosseguir com a medição.

#### **4.2. Análise dos Resultados**

Os resultados indicam que o robô apresenta diferentes raios de curvatura dependendo da direção e sentido de rotação das rodas. Observou-se que a tração é um fator crucial que influencia o diâmetro das curvas realizadas por um robô. A capacidade de tração refere-se à força que as rodas exercem na superfície em que o robô está se movendo, acreditamos que por conta do terreno que o robô foi testado ser bastante irregular pode ter afetado negativamente os resultados, aumentando a diferença dos diâmetros das curvas. Observou-se que se com a tração dianteira seria possível um torque maior, e por isso maior será a aderência das rodas no solo e menor será o escorregamento, tendo assim curvas mais precisas.

Se o robô tiver uma boa tração, suas rodas serão capazes de agarrar a superfície de maneira mais eficiente, permitindo que o robô execute curvas com raios menores. Isso significa que, com a mesma configuração e controle, um robô com maior tração será capaz de realizar curvas com um diâmetro menor.

Foi necessário colocar um peso extra dentro da plataforma para compensar o peso da bateria e dos outros componentes, sem esse contrapeso o robô ficava com a roda suspensa em alguns momentos, com isso foi notado que o peso dentro do robô pode influenciar nas curvas devido ao seu efeito na distribuição da carga e na estabilidade do robô durante o movimento. O peso dentro do robô pode ser distribuído de diferentes maneiras, afetando o centro de massa. Se a carga estiver desequilibrada ou concentrada em uma área específica, pode ocorrer um deslocamento do centro de massa, isso acaba resultando em uma mudança na maneira como o robô se comporta durante as curvas, afetando o diâmetro delas. Por exemplo, se o peso estiver deslocado para um lado do robô, ele tenderá a inclinar-se para esse lado durante a curva. Assim dependendo do sentido da tração juntamente com a distribuição de peso o robô pode fazer uma curva mais aberta ou mais fechada. Conforme o robô vai carregando peso ele vai perdendo a capacidade de realizar curvas, com isso o ideal seria carregar pouco peso caro o objetivo seja fazer curvas, se o objetivo for andar em linha reta a carga não é um problema.

## 5. Conclusão

O projeto do robô plataforma descrito mostra uma solução criativa e econômica para a criação de uma plataforma universal controlada por um microcontrolador ESP-32, utilizando componentes eletrônicos simples e acessíveis. Com capacidade de suportar mais de 100 kg, o robô plataforma apresentou boa estabilidade e resistência, atendendo às expectativas do projeto.

Embora a utilização de furadeiras como motor tenha sido uma solução interessante em termos de custo e facilidade de acesso, foi observado que o robô plataforma apresentou limitações em relação a cargas muito elevadas, como dificuldade para fazer curvas. Por conta disso, seria ideal avaliar opções de motores e baterias mais adequadas para aplicações com maior exigência de desempenho.

O uso do chassi de madeira é uma escolha interessante para um projeto mais econômico, mas é importante considerar que materiais mais resistentes, como o alumínio ou aço, podem ser mais adequados para aplicações que exijam maior robustez. A utilização de uma Ponte H para alimentar os motores é uma escolha comum em projetos de robótica e automação, seu uso permite o controle de direção e velocidade dos motores de forma eficiente o que facilitou muito o controle da potência do robô plataforma.

A tração é um fator fundamental que influencia o comportamento de um veículo durante as curvas. Em geral, a tração traseira tende a resultar em curvas mais abertas em comparação com a tração dianteira. Isso ocorre devido à distribuição de tarefas entre as rodas dianteiras e traseiras.

O desenvolvimento de um *software* em Python para controlar o robô plataforma foi uma boa opção pois a simplicidade da linguagem faz com que essa escolha se torne uma solução flexível e de fácil implementação. De qualquer forma, o robô plataforma demonstrou ser bastante versátil, e pode ser utilizada para diversas finalidades, como transporte de materiais, suporte para equipamentos, entre outros. A combinação do ESP-32 com o *software* em Python permitiu um melhor controle do robô plataforma, facilitando a operação e manobra em diferentes situações.

Por fim, é importante ressaltar que este projeto pode e deve ser expandido e aprimorado com a adição de novos componentes e funcionalidades, como sensores, atuadores, sistemas de comunicação e navegação, possibilitando uma grande variedade de aplicações. Vale destacar que este projeto foi desenvolvido para estudar

a possibilidade de criar uma plataforma robótica universal e de baixo custo. O desenvolvimento físico desse estudo trouxe informações relevantes para propor melhorias para os próximos projetos, além de fornecer novas ideias para problemas antigos, como é o caso dos motores com redutores, com baixo custo, mas alto torque.

Abaixo listamos as principais limitações do projeto que foram observadas ao longo dos testes.

- Carga máxima: embora o robô plataforma seja capaz de suportar cargas elevadas, é importante observar que tem um limite para a capacidade de carga, e sobrecargas vão acabar prejudicando o desempenho e a durabilidade do equipamento;
- Limitações do motor: a utilização de furadeiras como motor pode apresentar limitações em relação a torque e velocidade, o que pode afetar o desempenho do robô plataforma em determinadas situações;
- Controle manual: o controle manual do robô plataforma pode ser limitado em termos de precisão e segurança, especialmente em ambientes com obstáculos ou em que seja necessário navegar em trajetórias complexas.
- Para permitir uma maior manobrabilidade em curvas, o robô sobre 4 rodas precisa ter as 4 rodas motorizadas.

Em resumo, o projeto pode servir como base para aplicações futuras e se faz necessária a implementação de melhorias e a consideração das limitações são etapas importantes para que o desempenho do robô plataforma apenas aumente.

Como um dos objetivos do projeto era ter o baixo custo e ser mais acessível, decidimos montar uma tabela para mostrar os principais gastos deste robô. Concluímos que conseguimos alcançar a proposta do baixo custo já que um robô para esses fins, no mercado, é muito além de 4 mil reais.

Quadro 5 - Tabela dos principais custos do projeto.

Duas Furadeiras	R\$ 369,80
Bateria	R\$ 269,10
Drive Ponte H	R\$ 183,50
Rolamentos	R\$ 188,70
Quatro rodas	R\$ 240
Amperímetro	R\$ 52,23
Madeira MDF	R\$ 250,00
ESP-32	R\$ 36
Eixos	R\$ 1.800,00
Outros	R\$ 30,00
<b>TOTAL</b>	<b>R\$ 3.419,33</b>

### 5.1. Propostas futuras

Certamente existem diversas melhorias que podem ser implementadas no robô plataforma descrito para aprimorar seu desempenho e funcionalidade. Algumas possibilidades incluem:

- Utilização de motores dedicados para aplicações com cargas elevadas, possibilitando uma maior capacidade de tração e torque, e permitindo que o robô plataforma faça curvas com maior facilidade;
- Adição de sensores de distância para evitar obstáculos e aumentar a segurança na navegação;
- Integração de um sistema de navegação por GPS para permitir a navegação autônoma do robô plataforma em ambientes externos;
- Incorporação de um sistema de comunicação sem fio para permitir o controle remoto do robô plataforma ou a transmissão de dados para um computador ou servidor.
- Para utilizar a plataforma atual do robô sem aumentar os custos com mais motores, sugere-se substituir as duas rodas dianteiras por duas rodas loucas.

## 6. Referências

ALBUQUERQUE, Rafael. Python embarcado. Embarcados, 2022. Disponível em: <<https://embarcados.com.br/python-embarcado/>>. Acesso em: 01/10/2022.

ARDUINO COACH. O que é ESP-32, para que serve, como começar e é melhor que Arduino? Disponível em: <<https://www.arduino.coach/o-que-e-ESP-32-para-que-serve-como-comecar-e-melhor-que-arduino.HTML>>. Acesso em: 24/03/2022.

BUHMANN, J. M.; BURGARD, W.; CREMERS, A. B.; FOX, D.; HOFMANN, T.; SCHNEIDER, F. E.; STRIKOS, J.; THRUN, S. The Mobile Robot Rhino, EUA, out 2000. Ai Magazine. Disponível em: <[https://www.researchgate.net/publication/2461298\\_The\\_Mobile\\_Robot\\_Rhino](https://www.researchgate.net/publication/2461298_The_Mobile_Robot_Rhino)>. Acesso em: 25/11/2021

BUZZI, Luiz Henrique, “Projeto e Concepção de uma Plataforma Robótica Móvel Integrada com o ROS, Trabalho de Conclusão de Curso apresentado à Universidade Federal de Santa Catarina, 2019.

CURTOCIRCUITO. Conhecendo o ESP-32, 2022. Disponível em: <<https://curtocircuito.com.br/blog/conhecendo-ESP-32>>. Acesso em: 23/03/2022.

DANILUSKI, Guilherme Arcas, ROSA, Lucca Baratera, “Robô Guiado Mapeador de Ambientes Internos”, Trabalho de Conclusão de Curso apresentado à Fatec Santo André, 2018.

De volta ao negócio: Graças ao Novo Robô Agrícola de Agricultores Continentais podem se concentrar novamente no essencial. Continental, 2021. Disponível em: <<https://www.continental.com/en/products-and-innovation/innovation/agriculture/agricultural-robot-contadino/>>. Acesso em: 25/11/2021

ELETROGATE. RTOS: Sistema Operacional de Tempo Real. Disponível em: <<https://blog.eletrogate.com/rtos-sistema-operacional-de-tempo-real/#:~:text=Um%20Sistema%20Operativo%20em%20Tempo,resposta%20%C3%A9%20elevada%20ou%20n%C3%A3o>>. Acesso em: 04/04/2022.

ESP-32/ESP8266 MicroPython Web Server – Control Outputs. Disponível em: <<https://randomnerdtutorials.com/ESP-32-esp8266-micropython-web-server/>>. Acesso em: 16/01/2023.

Indústria 4.0: Entenda seus conceitos e fundamentos. Portal da Indústria, 2021. Disponível em: <<https://www.portaldaindustria.com.br/industria-de-a-z/industria-4-0/>>. Acesso em: 18/11/2021

LIMA, João. Conhecendo ESP-32. Curto Circuito. Disponível em: <<https://curtocircuito.com.br/blog/Categoria%20IoT/conhecendo-ESP-32>>. Acesso em: 23/03/2022.

MATSUMURA, Takao, “Desenvolvimento de uma Plataforma Aberta de Robô Móvel para Propósitos Gerais, Dissertação de Mestrado apresentado à Universidade Estadual Paulista Julio de Mesquita Filho, 2014.

MicroPython: ESP-32/ESP8266 Access Point (AP). Disponível em: <<https://randomnerdtutorials.com/micropython-ESP-32-esp8266-access-point-ap/>>. Acesso em: 16/01/2023.

MICROPYTHON. MicroPython Library Documentation. Disponível em: <<https://docs.micropython.org/en/latest/library/index.HTML>>. Acesso em: 20/07/2022.

MIRANDA, Lucas. Sistemas embarcados com Python. iMasters. Disponível em: <<https://imasters.com.br/desenvolvimento/sistemas-embarcados-com-python#:~:text=Comunica%C3%A7%C3%A3o%20do%20Python%20com%20sistemas,o%20usu%C3%A1rio%20automatize%20seus%20testes>>. Acesso em: 15/09/2022.

MIXXFERRAMENTAS. Parafusadeira Furadeira Reversível 12V 1.5Ah com Carregador Bivolt Lith LT7035, 2023. Disponível em: <<https://www.mixxferramentas.com.br/parafusadeira-furadeira-reversivel-12v-15ah-com-carregador-bivolt-lith-lt7035>>. Acesso em: 22/04/2023.

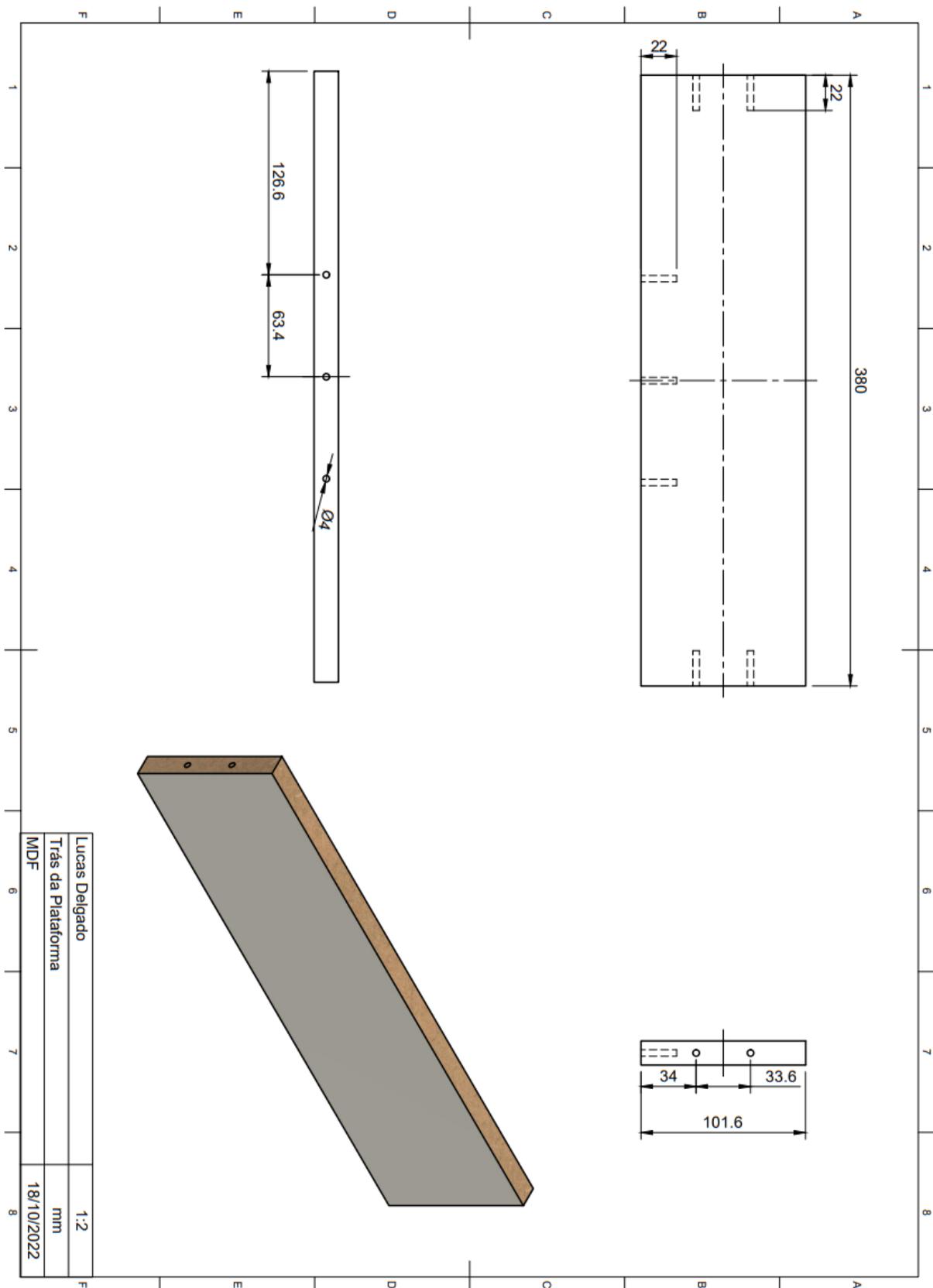
NEOIPSUM. Madeira na construção, 2023. Disponível em: <<https://neoipsum.com.br/madeira-na-construcao/>>. Acesso em: 03/04/2023.

OLHAR DIGITAL. Amazon implanta robôs autônomos de entrega em cidade da Califórnia. 2019. Disponível em: <<https://olhardigital.com.br/2019/08/08/noticias/amazon-implanta-robos-autonomos-de-entrega-em-cidade-da-california/#:~:text=A%20Amazon%20lan%C3%A7ou%20oficialmente%20seus,necessidade%20de%20vans%20e%20carros>>. Acesso em: 09/05/2023.

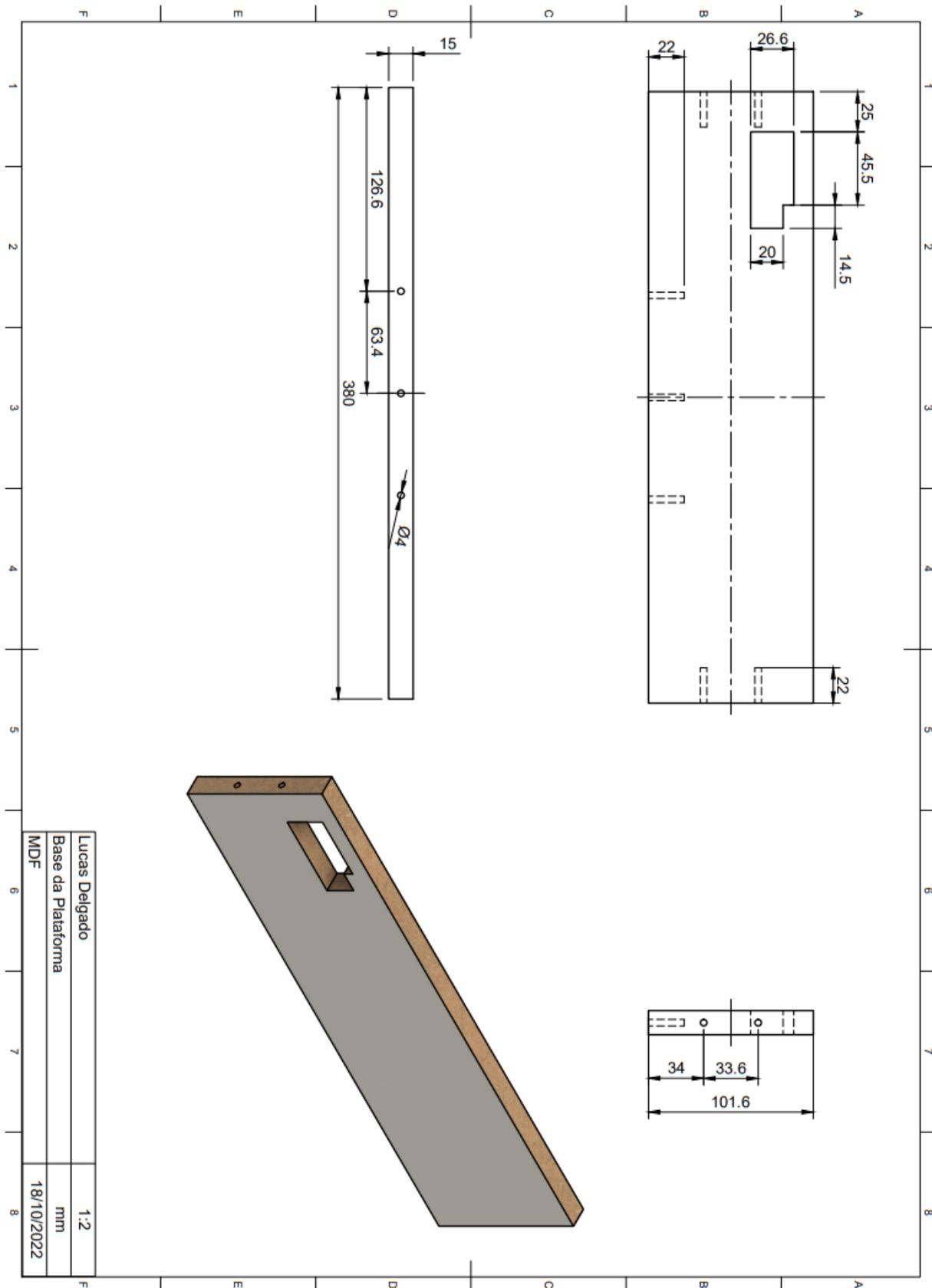
PPLWARE. Trombia Free: o robô autônomo que limpa as ruas. Disponível em: <<https://pplware.sapo.pt/motores/trombia-free-o-robo-autonomo-que-limpa-as-ruas/>>. Acesso em: 09/05/2023.

SILVA, Henrique. O que são sistemas embarcados? Embarcados. Disponível em: <<https://embarcados.com.br/o-que-sao-sistemas-embarcados/>>. Acesso em: 01/10/2022.

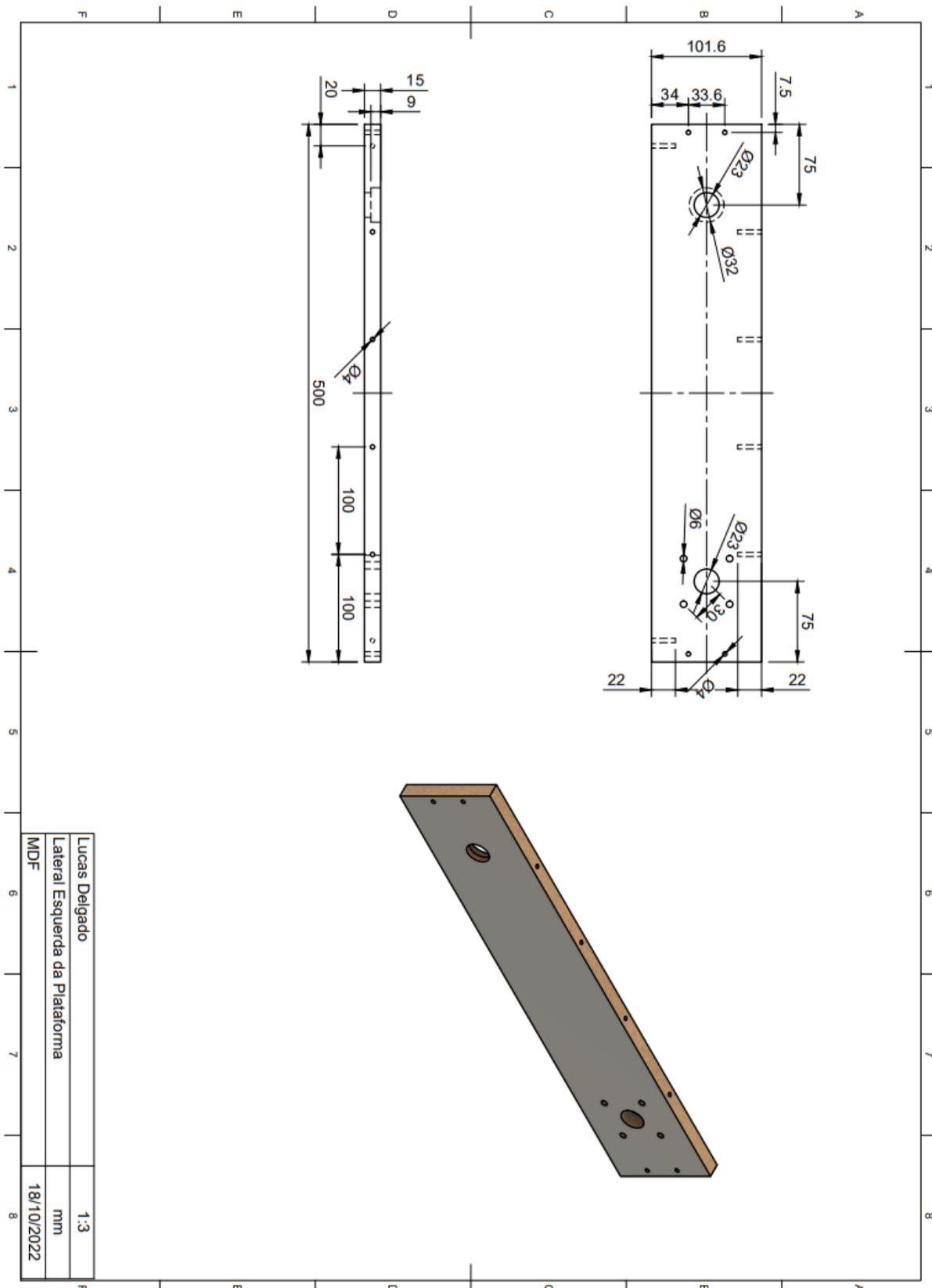
### Apêndice 1 – Dimensões da Peça Traseira



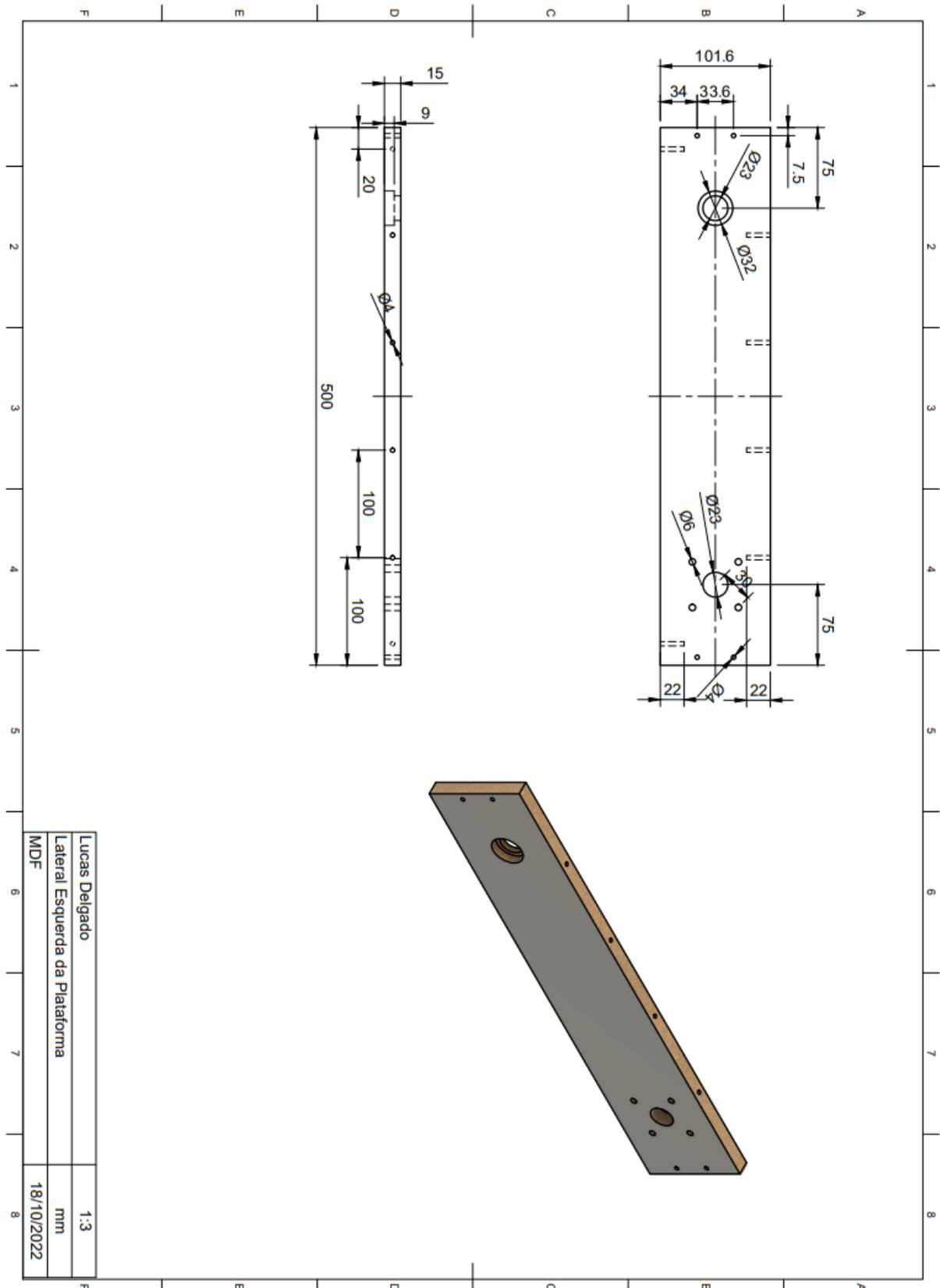
### Apêndice 2 – Dimensões da peça Dianteira



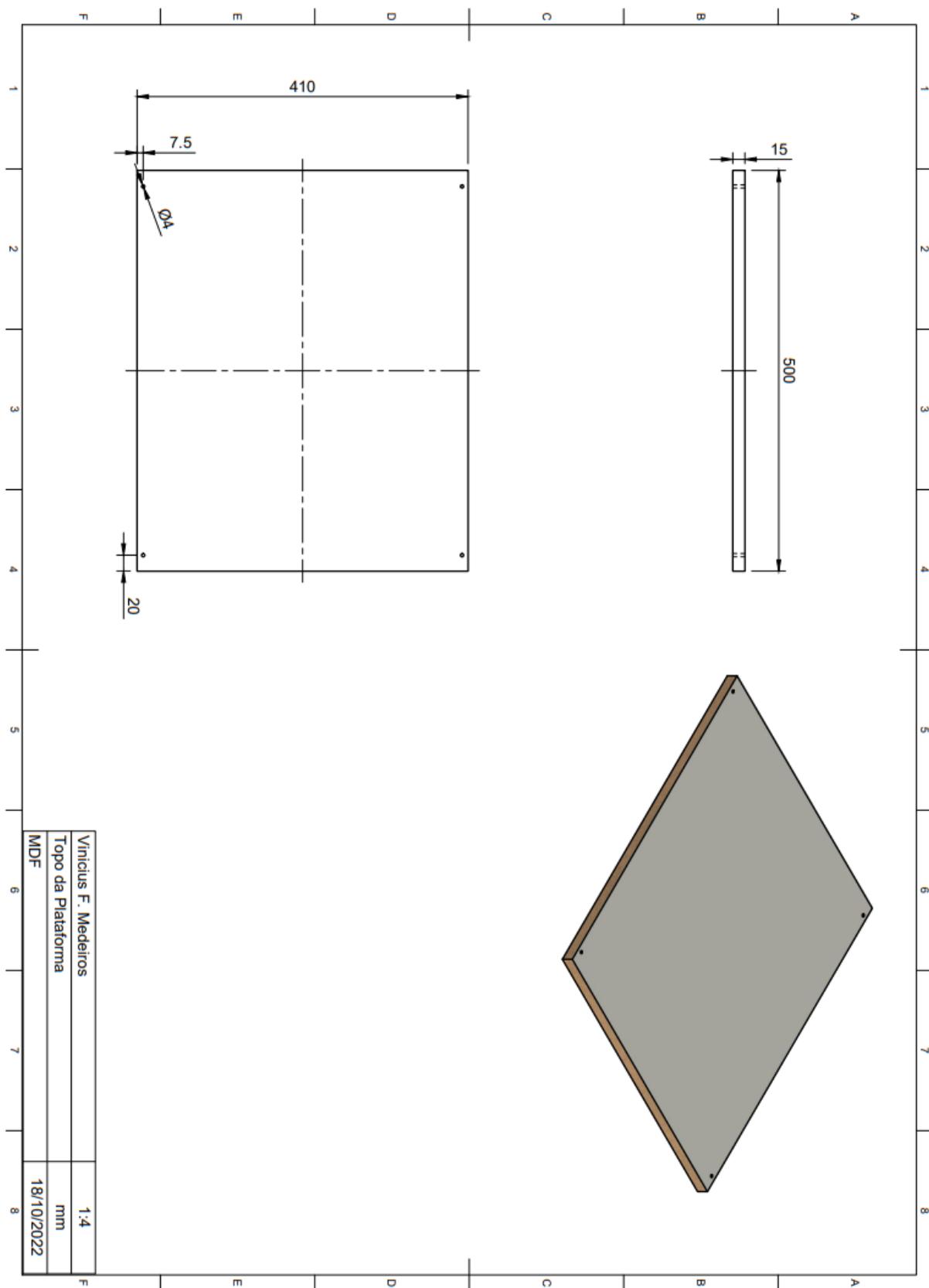
### Apêndice 3 – Lateral Esquerda do Robô



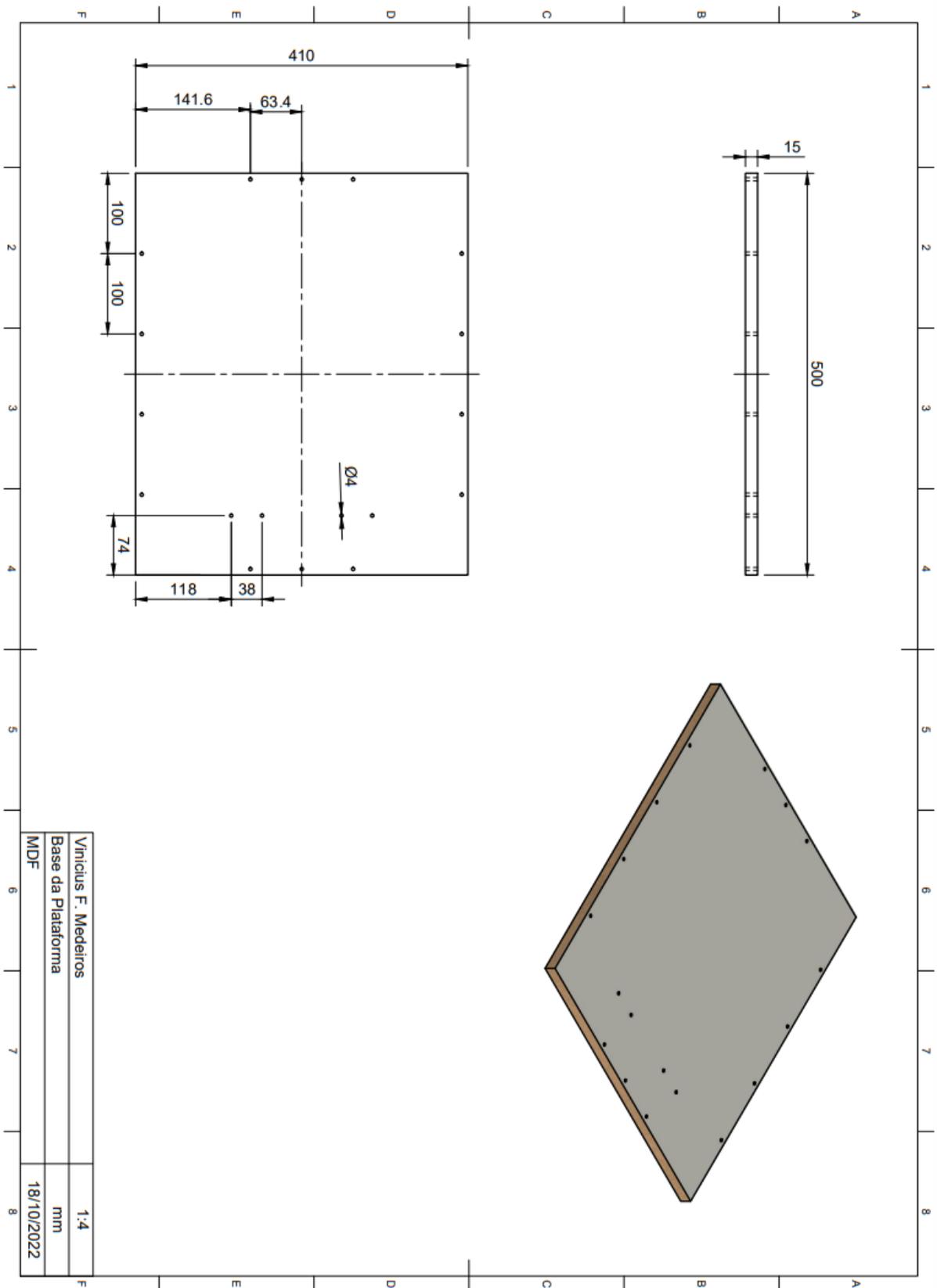
## Apêndice 4 – Lateral Direita do Robô



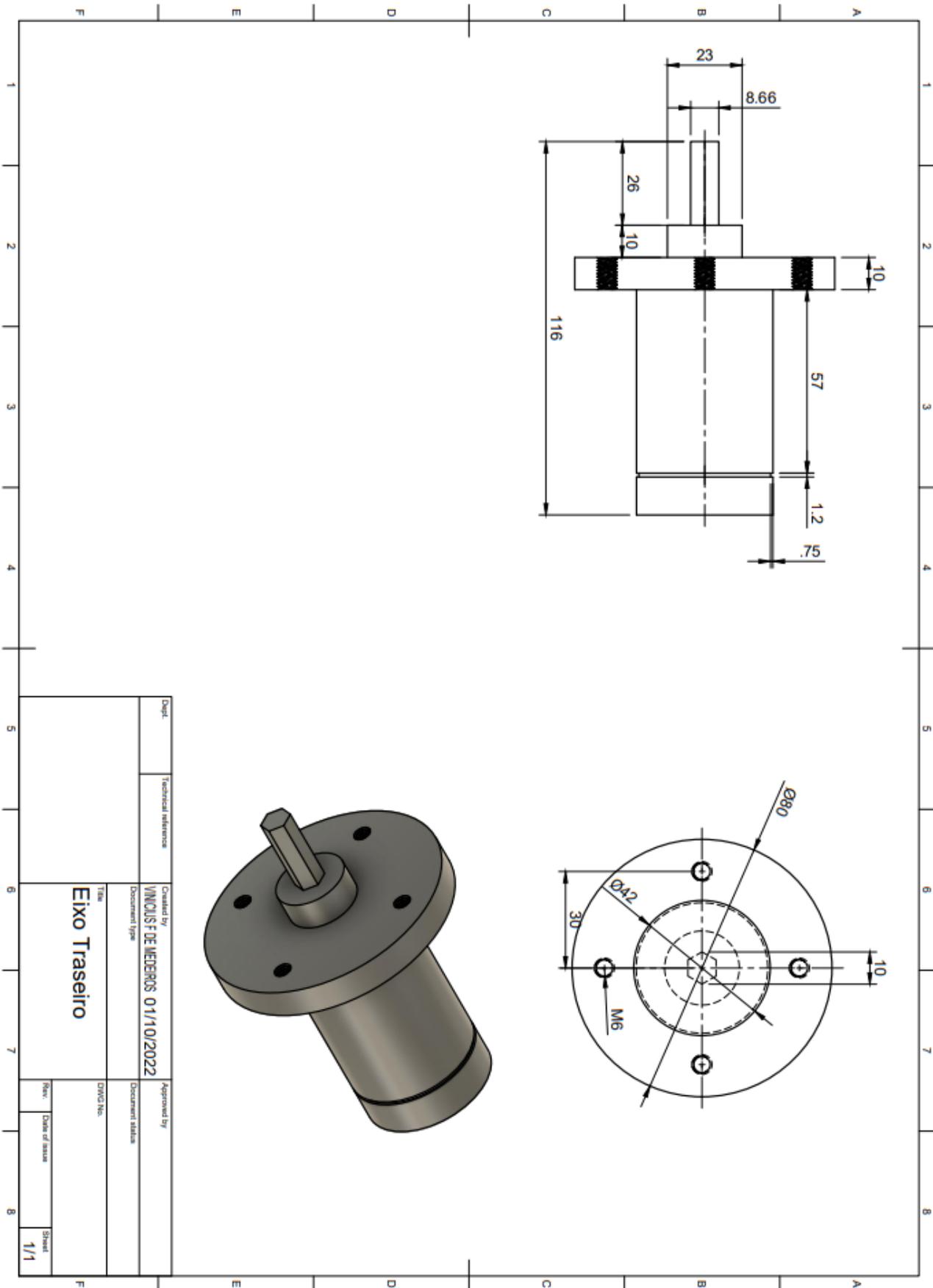
## Apêndice 5 – Dimensões da peça do Topo



### Apêndice 6 – Dimensões da peça da Base

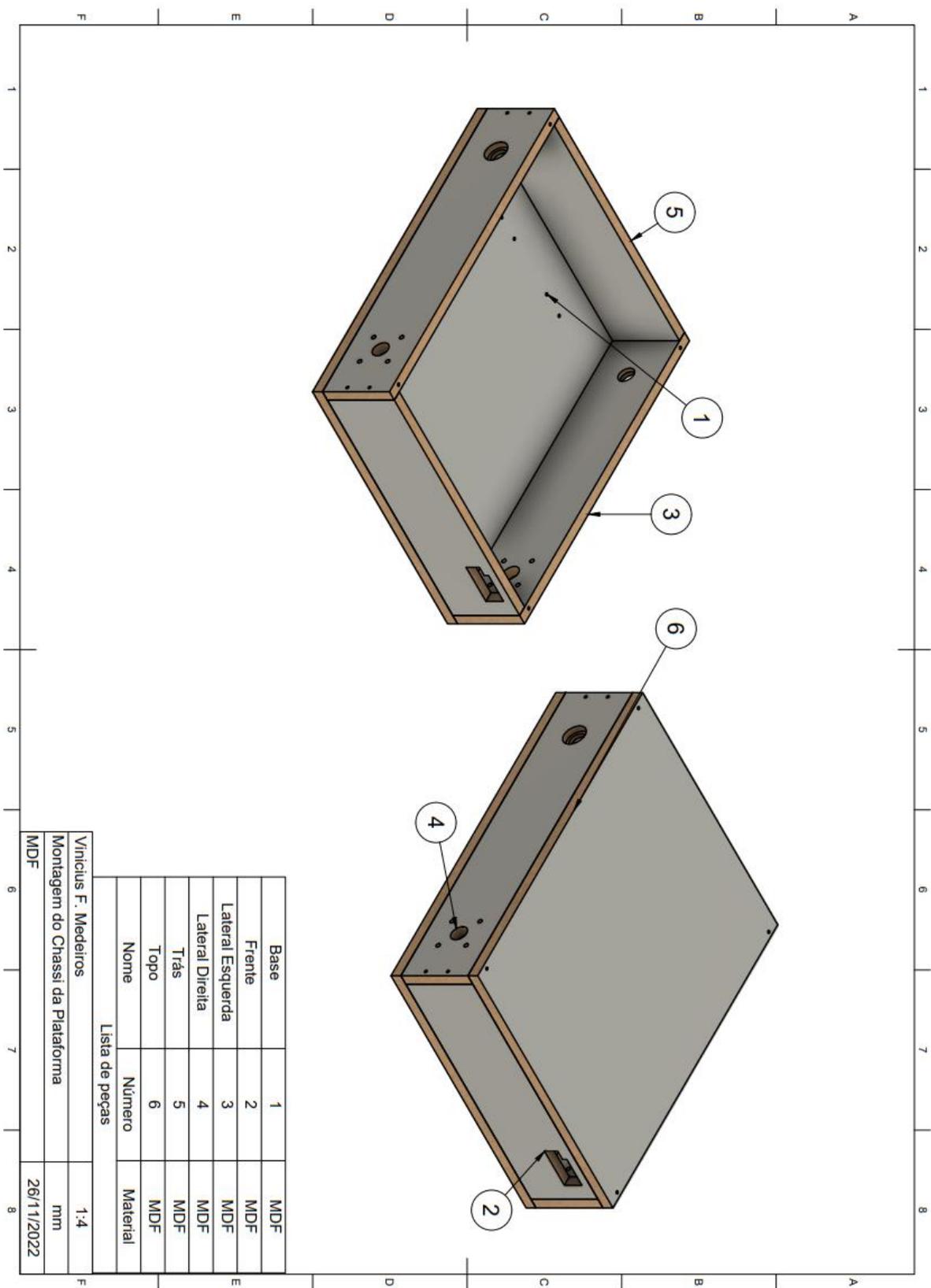


### Apêndice 7 – Dimensões do Eixo Traseiro





## Apêndice 9 – Montagem Simplificada do Robô



## Apêndice 10 – Programa Desenvolvido em Python

```

try:
    import usocket as socket
except:
    import socket

from machine import Pin, PWM
import network
import esp

esp.osdebug(None)

import gc
gc.collect()

ssid = 'TCC Robo'

ap = network.WLAN(network.AP_IF)
ap.active(True)
ap.config(essid=ssid)

while ap.active() == False:
    pass

print('Connection successful')
print(ap.ifconfig())

esquerda1 = PWM(Pin(12))
esquerda2 = PWM(Pin(13))
direita1 = PWM(Pin(14))
direita2 = PWM(Pin(27))

velocidade = 0

def movimentar_robo(movimento, vel):
    if movimento == 'andar frente':
        esquerda1.duty(vel)
        esquerda2.duty(0)
        direita1.duty(vel)
        direita2.duty(0)
    if movimento == 'andar tras':
        esquerda1.duty(0)
        esquerda2.duty(vel)
        direita1.duty(0)
        direita2.duty(vel)
    if movimento == 'virar direita':
        esquerda1.duty(0)

```

```

    esquerda2.duty(0)
    direita1.duty(vel)
    direita2.duty(0)
if movimento == 'virar esquerda':
    esquerda1.duty(vel)
    esquerda2.duty(0)
    direita1.duty(0)
    direita2.duty(0)
if movimento == 'parar':
    esquerda1.duty(0)
    esquerda2.duty(0)
    direita1.duty(0)
    direita2.duty(0)

def web_page():

    html = """
    <html>
      <head>
        <title>Servidor Web ESP32</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1">
        <link rel="icon" href="data:,">
        <style>
          h1 {
            font-size: 60px; /* Tamanho original */
          }
          body {
            text-align: center;
          }
          .button {
            padding: 10px 20px;
            font-size: 16px;
            margin: 4px 4px;
          }
          .button1 {
            padding: 25px 25px;
            font-size: 36px;
            margin: 4px 40px;
            position: relative;
          }
          .button1::before{
            content: '';
            position: absolute;
            width: 0.4em;
            height: 0.4em;
            border: solid rgb(0, 0, 0);
            border-width: 0 0.1em 0.1em 0;

```

```

        transform: translate(-50%, -20%) rotate(225deg);
    }
    .button2::before{
        transform: translate(-30%, -50%) rotate(135deg);
    }
    .button3::before{
        transform: translate(-70%, -50%) rotate(315deg);
    }
    .button4::before{
        transform: translate(-50%, -80%) rotate(45deg);
    }
}
</style>
</head>
<body>
    <h1>Controlar Robô</h1>

    <p><a href="/?velocidade=minima"><button
class="button">MÍNIMA</button></a>
    <a href="/?velocidade=media"><button
class="button">MÉDIA</button></a>
    <a href="/?velocidade=maxima"><button
class="button">MÁXIMA</button></a></p><br>
    <p><a href="/?andar=frente"><button class="button1
button1"></button></a></p>
    <p><a href="/?virar=esquerda"><button class="button1
button2"></button></a>
    <a href="/?virar=direita"><button class="button1
button3"></button></a></p>
    <p><a href="/?andar=tras"><button class="button1
button4"></button></a></p><br>
    <p><a href="/?parar"><button
class="button">PARAR</button></a></p>
</body>
</html>""
return html

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('', 80))
s.listen(5)

while True:
    conn, addr = s.accept()
    print('Got a connection from %s' % str(addr))
    request = conn.recv(1024)
    request = str(request)
    print('Content = %s' % request)
    vel_min = request.find('/?velocidade=minima')
    vel_med = request.find('/?velocidade=media')
    vel_max = request.find('/?velocidade=maxima')
```

```
andar_frente = request.find('/?andar=frente')
virar_esquerda = request.find('/?virar=esquerda')
virar_direita = request.find('/?virar=direita')
andar_tras = request.find('/?andar=tras')
parado = request.find('/?parar')
if vel_min == 6:
    velocidade = 120
if vel_med == 6:
    velocidade = 260
if vel_max == 6:
    velocidade = 400
if andar_frente == 6:
    movimentar_robo('andar frente', velocidade)
if parado == 6:
    movimentar_robo('parar', velocidade)
    velocidade=0
if andar_tras == 6:
    movimentar_robo('andar tras', velocidade)
if virar_direita == 6:
    movimentar_robo('virar direita', velocidade)
if virar_esquerda == 6:
    movimentar_robo('virar esquerda', velocidade)

response = web_page()
conn.send('HTTP/1.1 200 OK\n')
conn.send('Content-Type: text/html\n')
conn.send('Connection: close\n\n')
conn.sendall(response)
conn.close()
```