

**FACULDADE DE TECNOLOGIA DE SÃO BERNARDO DO CAMPO
“ADIB MOISÉS DIB”**

BEATRIZ TOLENTINO MARQUES
BRENDA DE MORAES MERLIN
CAROLINE BORGES CONSENTINO
KETHLEEN MAYUMI NAKAGAWA

ENTREGA DE PEDIDOS NOS RESTAURANTES COM O USO DE AGV

**BEATRIZ TOLENTINO MARQUES
BRENDA DE MORAES MERLIN
CAROLINE BORGES CONSENTINO
KETHLEEN MAYUMI NAKAGAWA**

ENTREGA DE PEDIDOS NOS RESTAURANTES COM O USO DE AGV

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Bernardo do Campo “Adib Moisés Dib” como requisito parcial para a obtenção do título de Tecnóloga em Automação Industrial.

Orientador: Prof. Esp. Jorge Luis Sarapka

São Bernardo do Campo – SP
Junho/2023

**BEATRIZ TOLENTINO MARQUES
BRENDA DE MORAES MERLIN
CAROLINE BORGES CONSENTINO
KETHLEEN MAYUMI NAKAGAWA**

ENTREGA DE PEDIDOS NOS RESTAURANTES COM O USO DE AGV

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Bernardo do Campo “Adib Moisés Dib” como requisito parcial para a obtenção do título de Tecnóloga em Automação Industrial.

Orientador: Prof. Esp. Jorge Luis Sarapka

Trabalho de Conclusão de Curso apresentado e aprovado em:_____/ /2023

Banca Examinadora:

Prof. Esp. Jorge Luis Sarapka, FATEC SBC – Orientador

Prof. Dr. _____, FATEC SBC – Avaliador

Prof. Esp. _____, FATEC SBC – Avaliador

Dedicamos aos nossos amigos e familiares que nos apoiaram durante nossa trajetória acadêmica.

Agradecemos ao Prof. Esp. Jorge Luis Sarapka pela ajuda durante a elaboração deste trabalho.

“A tarefa não é tanto ver aquilo que ninguém viu, mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê”.

ARTHUR SCHOPENHAUER

RESUMO

O presente projeto consiste na confecção de um sistema de automatização dos pedidos realizados pelos clientes dentro dos restaurantes, incluindo nesse sistema a automatização do transporte do produto a ser consumido até a mesa do cliente solicitante por meio de um protótipo AGV seguidor de linha, controlado pelo Arduino, e uma plataforma online para solicitação e recebimento dos pedidos. Com o uso da eletrônica e da lógica de programação, o projeto garante a qualidade do atendimento, além de proporcionar uma experiência nova ao cliente, padronizando o processo da entrega dos pedidos, auxiliando os funcionários nos estabelecimentos quanto ao trabalho exaustivo e diminuindo possíveis acidentes e erros causados pela ação humana. Para a realização desse projeto, o embasamento teórico das pesquisas bibliográficas realizadas acrescentou o conhecimento necessário para o desenvolvimento prático, além da metodologia científica aplicada para direcionar o melhor caminho para obter o resultado esperado. As fundamentações deste trabalho apresentadas em tópicos são: automatização no processo de pedidos, veículo guiado automaticamente (AGV), sensores, motor de corrente contínua, impressão 3D, *software* Inventor, SkethUp e Olaclick.

Palavras-chave: Automatização. Robô. AGV. Restaurante. Pedido.

ABSTRACT

This project consists of the preparation of a system for automating the orders placed by customers within restaurants, including in this system the automation of the transport of this product to be consumed to the table of the requesting customer, through a prototype AGV line follower, controlled by the Arduino, and an online platform for requesting and receiving orders. With the use of electronics and programming logic, the project ensures the quality of service, in addition to providing a new experience to the customer, standardizing the process of delivery of orders, assisting employees in establishments with exhaustive work and reducing possible accidents and errors caused by human action. For the realization of this project, the theoretical basis of the bibliographic research carried out added the necessary knowledge for practical development, in addition to the applied scientific methodology to direct the best way to obtain the expected result. The foundations of this work presented in topics are: automation in the order process, automatically guided vehicle (AGV), sensors, direct current motor, 3D printing, *software* Inventor, SkethUp and OIaclick.

Keywords: Automation. Robot. AGV. Restaurant. Request.

LISTA DE FIGURAS

Figura 1.1 – Robô garçom de Kunshan.....	19
Figura 1.2 – Arduino UNO.....	21
Figura 1.3 – Sensores para Automação.....	24
Figura 1.4 – Sensores ultrassônicos.....	24
Figura 1.5 – Cone sonoro.....	25
Figura 1.6 – Tipos de sensores ópticos.....	26
Figura 1.7 – Sensor óptico por barreira.....	27
Figura 1.8 – Sensor óptico difuso	27
Figura 1.9 – Sensor óptico reflexão.....	28
Figura 1.10 – Motor de corrente contínua com e sem escovas.....	29
Figura 1.11 – Impressão 3D pelo método FDM.....	31
Figura 1.12 – <i>Software</i> Inventor.....	33
Figura 2.1 – Protótipo do Robô AGV	36
Figura 2.2 – Protótipo do Robô AGV cotado.	37
Figura 2.3 – Fluxograma geral do processo.....	38
Figura 2.4 – <i>Layout</i> do restaurante... ..	39
Figura 2.5 – Fluxograma do subprograma de colisão.....	40
Figura 2.6 – Fluxograma do subprograma do seguidor de linha.....	41
Figura 3.1 – Projeto finalizado.....	46
Figura 3.2 – Bases em acrílico do chassi.....	48
Figura 3.3 – Base com espaçadores.....	48
Figura 3.4 – Motor acoplado na base.....	49
Figura 3.5 – Base com as rodas montadas na base inferior.....	50
Figura 3.6 – Base principal.....	51
Figura 3.7 – Base principal com espaçadores.....	52

Figura 3.8 – Base principal fixada com base do chassi.....	52
Figura 3.9 – Cúpula inferior.....	53
Figura 3.10 – Montagem cúpula inferior e base principal.....	53
Figura 3.11 – Cúpula superior.....	54
Figura 3.12 – Pontos de montagem das cavilhas.....	55
Figura 3.13 – Montagem da prateleira.....	55
Figura 3.14 – Montagem completa.....	56
Figura 3.15 – Motores com fiação isoladas.....	57
Figura 3.16 – Conexões do módulo L298N.....	57
Figura 3.17 – Conexão no borne L298N.....	58
Figura 3.18 – Sensores ópticos e sensor magnético.....	59
Figura 3.19 – Tela para comunicação <i>bluetooth</i>	61
Figura 3.20 – Conexão eletrônica física.....	62
Figura 3.21 – Conexão eletrônica.....	62
Figura 3.22 – Criação do menu editável Olaclick.....	65
Figura 3.23 – Configuração dos horários de atendimento.....	65
Figura 3.24 – Cardápio digital.....	66

LISTA DE TABELAS

Tabela 2.1 – Cronograma do segundo semestre de 2022.....	42
Tabela 2.2 – Cronograma do primeiro semestre de 2023.....	43
Tabela 2.3 – Materiais utilizados à confecção do projeto.....	44
Tabela 3.1 – Indicação do tempo de impressão e peso das peças.....	50

LISTA DE ABREVIATURAS E SIGLAS

AGV	Veículo Guiado Automaticamente
PDA	Assistente Pessoal Digital
LED	Diodo Emissor de Luz
GPS	Sistema de Posicionamento Global
IDE	Ambiente de Desenvolvimento Integrado
USB	Barramento Serial Universal
CAD	Desenho Assistido por Computador
PLA	Polilactídeo
ABS	Acrilonitrila Butadieno Estireno

SUMÁRIO

INTRODUÇÃO	14
1 FUNDAMENTAÇÃO TEÓRICA	16
1.1 Automação no processo de pedidos	16
1.1.1 Aplicativo para pedidos.....	17
1.1.2 Cardápio digital em <i>tablets</i>	17
1.1.3 Comandas eletrônicas utilizadas pelos garçons	17
1.1.4 Caixa ou totens de autoatendimento	18
1.2 Veículo Guiado Automaticamente (AGV)	18
1.3 Arduino	20
1.3.1 Linguagem C++	22
1.4 Sensores	23
1.4.1 Sensores ultrassônicos.....	24
1.4.2 Sensores ópticos	25
1.5 Motor de corrente contínua	29
1.5.1 Vantagens de um motor de corrente contínua.....	30
1.5.2 Desvantagens de um motor de corrente contínua.....	30
1.6 Impressão 3D	30
1.6.1 Tipos de impressões.....	30
1.6.2 GTMax3D Core A2v2	31
1.6.3 Materiais para impressão	31
1.6.4 <i>Software Cura</i>	32
1.7 Inventor	33
1.8 SketchUp	33
1.9 OIaclick	34
2 METODOLOGIA	35
2.1 O que é Metodologia	35
2.2 O tema-problema	35
2.3 Justificativa	37

2.4	Fluxogramas	38
2.5	Etapas teóricas e práticas para o desenvolvimento do projeto	42
3	DESENVOLVIMENTO DO PROJETO	46
3.1	Projeto finalizado.....	46
3.2	Desenvolvimento da montagem do robô	47
3.2.1	Montagem do chassi.....	47
3.2.2	Impressão 3D	50
3.2.3	Montagem eletrônica	56
3.3	Programação do microcontrolador.....	63
3.4	Criação do menu digital na plataforma Olaclick.....	64
3.5	Integração das partes e testes de funcionamento.....	66
3.6	Obstáculos e soluções.....	67
	CONSIDERAÇÕES FINAIS	69
	REFERÊNCIAS.....	70
	APÊNDICES	73

INTRODUÇÃO

Nas últimas décadas a implementação de robôs autônomos, tais como AGV (*Automated Guided Vehicle* – Veículo Guiado Automaticamente), apresentou acentuado crescimento dentro dos setores industriais. Seu objetivo baseia-se na redução de custos com a automatização do processo de transporte dos produtos. As indústrias tendem a ser pioneiras nas aplicações de novas tecnologias e com o tempo são utilizadas em outros setores, como o de serviços. Assim, buscam o mesmo objetivo, porém com adaptações para a devida aplicação.

Dessa forma, o setor terciário, composto pelas atividades de comércios e serviços, como os restaurantes, tornou-se mais um ambiente para a implementação dessa tecnologia. Com isso, o presente projeto que se intitula Entrega de pedidos nos restaurantes com o uso de AGV, tem como objetivo o desenvolvimento de um sistema que inclui a automatização do pedido do cliente nos restaurantes, feito através de *tablets*, até o transporte do produto à sua mesa, que será transportado por um robô controlado por microcontrolador.

A inserção de robôs AGVs nesse setor trará melhorias na qualidade de atendimento e na eficiência do processo, devido ele não apresentar cansaço físico e erros de operação, que podem, muitas vezes, gerar uma má avaliação dos clientes para o estabelecimento. Sendo assim, essa inovação contribuirá no auxílio desse processo, reduzindo o número de funcionários, trazendo lucro ao proprietário e redução do tempo de pedido.

Por se tratar de uma implementação ainda não tão comum nesse setor, surgirá nas pessoas uma curiosidade para conhecer um local diferenciado, aumentando a clientela do restaurante.

O trabalho está dividido nos seguintes capítulos:

Capítulo 1 – Fundamentação Teórica: descreverá sobre conceitos técnicos com embasamento teórico, sendo esses conceitos a base para o desenvolvimento da parte prática a ser realizada.

Capítulo 2 – Metodologia: será relatado os métodos utilizados que permitiram criar uma rotina de desenvolvimento do conteúdo teórico e prático.

Capítulo 3 – Desenvolvimento do Projeto: nesse item será utilizado a fundamentação teórica como base para desenvolver a parte prática, apresentando a programação realizada, a comunicação entre os sistemas envolvidos no projeto e a montagem do protótipo com os materiais citados no desenvolvimento teórico.

E finalmente, as Considerações finais: são descritos os objetivos propostos e atingidos, justificativa, pontos fortes e fracos, conquistas alcançadas relações entre as teorias e os fatos verificados e possíveis sugestões para futuros projetos.

1 FUNDAMENTAÇÃO TEÓRICA

O presente capítulo abordará pesquisas com embasamento teórico para auxiliar com o desenvolvimento e com a sustentação do projeto **ENTREGA DE PEDIDOS NOS RESTAURANTES COM O USO DE AGV**.

1.1 Automatização no processo de pedidos

Quando se fala sobre os clientes que frequentam estabelecimentos como bares e restaurantes, estamos falando de um público que além de procurar qualidade na comida servida, procuram também locais onde se oferecem uma alta qualidade de serviço e atendimento. Vive-se na era da tecnologia, onde basta pesquisar o nome do estabelecimento na *internet* e já é possível ver em destaque as avaliações das pessoas que já o frequentaram, e com isso, quando entrega-se um mau serviço de atendimento, a imagem do estabelecimento é prejudicada, o que resulta na perda de muitos clientes, podendo levar a falência da empresa.

Hoje, a tecnologia está presente em diversas empresas, mostrando-se um fator chave para aumentar a eficiência e melhorar os resultados. Uma prova do impacto da revolução digital é a automatização dos serviços nos restaurantes, sendo uma tendência poderosa e se tornou indispensável para gerenciar com sucesso estes estabelecimentos.

Com isso, pode-se ver melhorias significativas quanto ao atendimento que acaba se tornando mais ágil, fluído e menos propenso a erros e desperdícios (SANTOS, 2014).

Segundo Goomer (2019), uma de muitas empresas responsáveis pela criação de cardápios digitais e auxílio para que os restaurantes desenvolvam seu próprio aplicativo, disponibilizou uma pesquisa descrevendo vantagens e desvantagens de quatro tipos de soluções para sistemas de pedidos nos restaurantes. A seguir será relatado o funcionamento brevemente desses sistemas.

1.1.1 Aplicativo para pedidos

O cliente poderá realizar o *download* do aplicativo do restaurante no seu celular, tendo acesso as opções do cardápio e podendo efetuar o pedido. Esse sistema é utilizado tanto em ambientes locais, ou seja, no próprio restaurante, como também por *delivery* (sistema de entrega no local que foi solicitado), o que traz grande benefício para o estabelecimento.

Porém, uma das dificuldades encontradas para o uso desse tipo de aplicação é o engajamento com os clientes, devido a questão um tanto quanto burocrática que é desde o processo inicial, com a instalação do aplicativo em seu celular, navegação pelo cardápio e depois a finalização do pedido. Todas essas etapas são feitas pelo próprio cliente que pode não estar acostumado com o sistema de solicitação do local, podendo causar certa morosidade durante esse processo (GOOMER, 2019).

1.1.2 Cardápio digital em *tablets*

Ao chegar no estabelecimento, o cliente se depara com *tablets* (dispositivos em formato de prancheta com um sistema operacional que pode ser usado para ter acesso a *internet*) em sua mesa, sendo ele totalmente configurado para as utilizações do restaurante, permitindo que o cliente acesse o cardápio e faça seu pedido, sem precisar se incomodar em realizar o *download* de aplicativos em seu celular.

Sendo um *tablet* por mesa, esse sistema permite diversos atendimentos com comandas individuais. Todavia, a curto prazo ainda demanda um alto custo do investimento realizado nesse tipo de equipamento (GOOMER, 2019).

1.1.3 Comandas eletrônicas utilizadas pelos garçons

Nesse tipo de processo, o cliente não terá acesso a um equipamento para realizar a solicitação do seu pedido. Assim, o próprio garçom utilizará o PDA (*Personal Digital Assistant* – Assistente Pessoal Digital), sendo um computador de pequeno porte, porém de alta capacidade computacional, para guardar os registros. O

estabelecimento poderá optar por um celular com o aplicativo do restaurante instalado, no lugar do PDA, também utilizado pelo garçom para registrar os pedidos.

Independente de qual for o equipamento utilizado pelo funcionário, esse terá capacidade de comunicação direta com o sistema da cozinha.

Esse tipo de tecnologia é mais ágil do que utilizar o antigo sistema, caneta e papel, pois o funcionário passa a ter mais liberdade de manusear o equipamento e registrar o pedido do cliente. Contudo, ainda é necessário o serviço de uma pessoa para a realização desse processo de solicitação e entrega do pedido (GOOMER, 2019).

1.1.4 Caixa ou totens de autoatendimento

Para solicitações em caixas e totens de autoatendimento, o cliente precisa se deslocar ao atendente no balcão, que registra o pedido por meio de painéis eletrônicos ou até totens de autoatendimento para realizar o pedido, evitando filas. Somente após isso, ele poderá se acomodar em alguma mesa e apreciar sua refeição.

Em redes de *Fast Foods* – comida rápida, como McDonald's, Burger King entre outros, são muito comuns esses tipos de solicitações (GOOMER, 2019).

1.2 Veículo Guiado Automaticamente (AGV)

Na década de 1950, o primeiro AGV (*Automated Guided Vehicle* – Veículo Guiado Automaticamente) chegou nas indústrias com o objetivo de transitar pela fábrica, levando materiais para outro local por meio de uma rota definida. Geralmente, esses objetos transportados eram volumosos, o que acabou aumentando a segurança na realização desse processo, além de facilitar o trabalho dos operários.

Inicialmente, esses robôs seguiam uma trilha de fios distribuídos pelo chão da fábrica, ou seja, condutores embutidos que geravam um campo magnético. Com o passar do tempo, os métodos utilizados evoluíram (INVIA ROBOTICS, 2019).

O AGV é um tipo de robô móvel, capaz de se mover ao longo de trajetórias pré-determinadas, realizando tarefas de forma autônoma e se tornando muito eficaz em locais onde há uma necessidade de realização de tarefas repetitivas (ALMEIDA, 2016).

Este veículo possui sua alimentação feita por baterias. Dessa forma, é possível que ele funcione durante o dia inteiro, sem que haja a intervenção humana, devido a sua locomoção autônoma (KIM e TANCHOCO, 1999).

Segundo a SSI SCHAEFER (2015), as vantagens que o AGV possui são: melhoria na ergonomia e nas condições de trabalho, aumento da segurança operacional, transporte seguro de diversos bens e embalagens, aumento da produtividade, minimização de danos no transporte e entregas erradas.

Segundo G1 (2014), um restaurante em *Kunshan* (China) dobrou o número de clientes com a implementação de dez robôs capazes de prepararem e servirem as refeições nas mesas. O proprietário *Song Yugang* conta que cada robô teve um investimento de aproximadamente um ano de salário pago a um funcionário, um investimento que a longo prazo traria economia ao dono, além de que os robôs não ficam doentes e podem trabalhar sem parar. Na época, a China estava passando por um aumento com os custos com funcionários o que motivou ainda mais os proprietários. Na Figura 1.1 pode-se ver um exemplo do robô responsável por realizar a entrega das bandejas de forma autônoma.

Figura 1.1 – Robô garçom de Kunshan



Fonte: www.g1.globo.com, 2022

Esses veículos podem ser conduzidos de quatro formas diferentes. A empresa deverá estudar e escolher a melhor alternativa para atender a sua necessidade.

- Fio indutivo no piso: transmite uma corrente de determinada intensidade e frequência;
- Fita magnética: circuito constituído por uma banda metálica fixada no piso;
- Meio óptico: o veículo percorre uma linha no piso mediante um dispositivo óptico de detecção nele instalado;
- A laser: varre o espaço em busca de referência para a movimentação do veículo.

O AGV seguidor de linha é geralmente um robô que possui rodas e sensores ópticos, que identificam se o robô está andando sobre a faixa. Caso o sensor detecte que houve um desvio no percurso, sua saída se torna nível baixo e assim é possível fazer lógicas para que o objetivo seja cumprido (ALMEIDA, 2016).

1.3 Arduino

Arduino é uma placa de prototipagem baseada no microcontrolador ATMEGA 328 que possibilita uma rápida prototipagem e que pode ser programada para processar dados de entrada e saída entre um dispositivo e componentes externos nele conectados. Chamado de plataforma de computação física ou embarcada, o Arduino nada mais é do que um sistema que pode interagir com seu ambiente por meio de *hardware* e *software*. É usada no protótipo que prova o conceito do sistema.

Pode ser usado para desenvolver objetos interativos autônomos ou pode ser conectado a um computador, rede, à internet para receber, enviar e responder aos dados do Arduino.

A Figura 1.2 ilustra o Arduino Uno:

Figura 1.2 – Arduino UNO



Fonte: www.planetaelectronico.com, 2022

Os Arduinos podem ser conectados a LEDs – Diodos Emissores de Luz, *displays* de matriz de pontos, botões, interruptores, motores, sensores de diversos tipos, receptores GPS (*Global Positioning System* – Sistema de Posicionamento Global) ou qualquer outro dispositivo que emita dados ou possa ser controlado. Na placa pode-se observar os pinos de entradas e saídas do Arduino, para que se possa conectá-los a outros circuitos ou sensores.

A placa do Arduino é composta de um microcontrolador ATMEL, um cristal ou oscilador (que é responsável por enviar pulsos de tempo em uma frequência especificada, que permite sua operação na velocidade correta) e um regulador linear de 5 V. Ele também pode contar com uma saída USB (*Universal Serial Bus* – Barramento Serial Universal), que permite conectá-lo a um computador para *upload* ou recuperação dos dados.

Para realizar a programação utiliza-se o IDE (*Integrated Development Environment* – Ambiente de Desenvolvimento Integrado) do Arduino, um *software* livre no qual escrevemos o código na linguagem que o Arduino entende, que é baseada na linguagem C++. O IDE permite a escrita de um programa computacional, que é um conjunto de instruções passo a passo que pode ser enviado para o Arduino, que executará essas instruções e interagirá com o que estiver conectado a ele. No mundo Arduino, os programas são chamados de *sketches* (MCROBERTS, 2011).

1.3.1 Linguagem C++

A linguagem C++ denomina-se como uma programação compilada multi paradigma, sendo uma das linguagens mais populares desde 1990 por ser portátil, flexível e ter uma padronização de seus compiladores (DELGADO, 2018).

Segundo Noletto (2021), essa linguagem pode ser considerada tanto de alto quanto de baixo nível, devido as bibliotecas que ela possui. Com essa adaptação de programação, ela é considerada como código de uso geral.

O símbolo “++” significa um acréscimo a Linguagem C. Dessa maneira, é encontrado determinados recursos da Linguagem C e novos métodos implementados, fazendo com que esse tipo de linguagem seja a quarta mais popular do mundo.

A linguagem possui bibliotecas pré-definidas para facilitar a utilização de funcionalidades comumente utilizadas, sendo elas:

- `#include<stdio.h>`: Biblioteca que utiliza os códigos dev c++;
- `#include<math.h>`: Biblioteca que utiliza operações matemáticas;
- `Main ()`: Função principal para identificar o programa;
- `Int`: Armazena variáveis inteiras;
- `Float`: Armazena variáveis flutuantes (números com virgulas);
- `Char`: Armazena caracteres;
- `Printf`: Imprime na tela um texto ou valor salvo em alguma variável do programa;
- `Scanf`: Grava um valor digitado no endereço destinado a essa variável;
- `For`: Laço de repetição, possibilita que seja repetido enquanto a condição for verdadeira;
- `If/Else`: Comando condicional onde faz com que o programa inclua condições para que algo seja executado, caso essa condição seja falsa, as linhas dentro do comando serão ignoradas.

1.4 Sensores

Os sensores são os componentes mais utilizados no mundo da eletroeletrônica. Qualquer processo automatizado apresenta diversos sensores por todo o seu sistema. Eles estão presentes no nosso cotidiano em diversas situações, como carros, eletrodomésticos e, principalmente, nas indústrias (CAPELLI, 2006).

O termo “sensor” é utilizado para equipamentos sensíveis a alguma forma de energia, seja luminosa, térmica ou cinética (THOMAZINI e ALBUQUERQUE, 2020).

Segundo Mattede (2006), os sensores são dispositivos que apresentam a função de detectar e responder algum estímulo, como temperatura, pressão, velocidade, posição entre outros.

Os sensores podem ser divididos de acordo com o tipo de sinal na saída, sendo analógicas ou digitais. Os sensores analógicos geram um sinal de saída de acordo com a variação da variável física, ou seja, apresentam diversos valores durante a sua medição. Diferente dos sensores analógicos, os digitais apresentam somente dois estados na saída, “*on*” ou “*off*” (ligado ou desligado). A variação do estado baixo para alto é alterado quando o sensor detecta um valor específico da variável física, que pode ser ajustado (THOMAZINI e ALBUQUERQUE, 2020).

Segundo DJP Automação (2020), existem diversos tipos de sensores industriais, conforme mostra a Figura 1.3. Entretanto, cada um apresenta seu funcionamento baseado na medição.

Figura 1.3 – Sensores para Automação



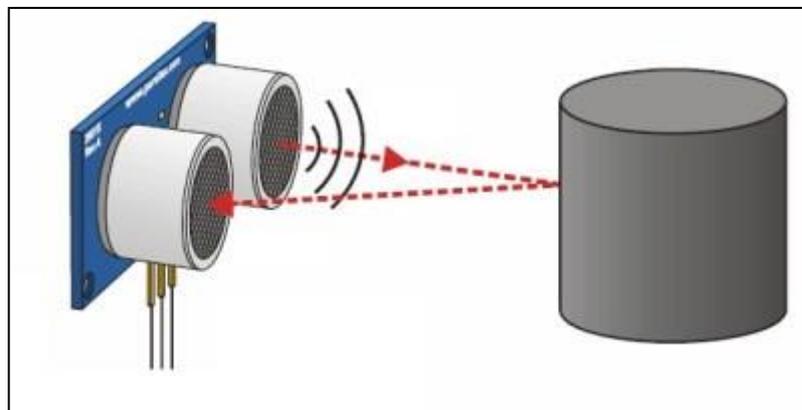
Fonte: www.scanelectronic.com.br, 2022

1.4.1 Sensores ultrassônicos

O princípio de funcionamento dos sensores ultrassônicos baseia-se na emissão e reflexão de ondas sonoras entre o objeto e o receptor. Assim, esse sensor detecta tudo que reflete o som e, conseqüentemente, mede a distância, que podem variar de 20 mm a 20 metros.

Os pulsos ultrassônicos são emitidos e refletidos por um objeto, assim ora o sensor funciona como emissor, ora como receptor, conforme mostra a Figura 1.4. Caso algum objeto, independente da forma, cor e constituição, incida no receptor, a saída é acionada (CAPELLI, 2006).

Figura 1.4 – Sensores ultrassônicos

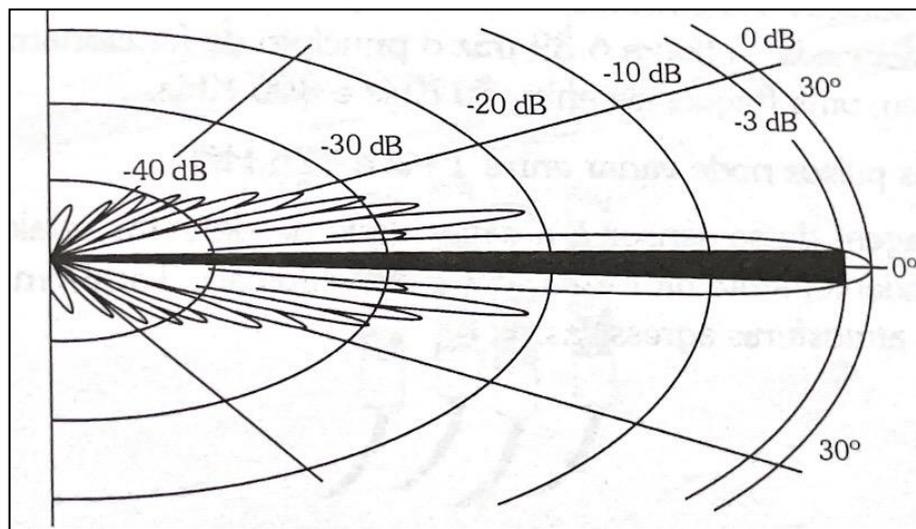


Fonte: www.vidadesilicio.com.br, 2022

Detecção de objeto, rompimento de fio, detecção de pessoas, medição de altura e largura são algumas das aplicações para o sensor ultrassônico (SILVEIRA, 2016).

As desvantagens que esse sensor possui são: necessidade de alinhamento angular, caso o cone formado pela emissão e reflexão do sinal ocorra fora do seu limite de detecção, observado na Figura 1.5, o objeto pode não ser detectado; baixa velocidade e custo alto em relação aos demais sensores (CAPELLI, 2006).

Figura 1.5 – Cone sonoro



Fonte: CAPELLI, 2006, pg 154

1.4.2 Sensores ópticos

O princípio de funcionamento dos sensores ópticos baseia-se na emissão de feixes de luz do emissor entre violeta (380 nm) e vermelha (780 nm), o qual é recebido por um elemento fotossensível chamado de receptor. É um dispositivo que converte os raios de luz em sinais eletrônicos. Seu principal objetivo é medir uma quantidade de luz emitida e transformá-la em medida para que se possa fazer a comparação e medição do deslocamento de algum objeto a sua frente (CAPELLI, 2006).

Esse tipo de sensor tem como finalidade fazer o mapeamento do local e guardar as informações no próprio sistema do Arduino, isso permite que o AGV se mova de uma forma autônoma sem que haja necessidade de interferências ou introduções de novos meios (SINOVA, 2022).

As principais vantagens que o sensor óptico possui são: alta sensibilidade, alcance, resolução e grande faixa de operação, onde pequenos objetos podem ser detectados a longas distâncias.

A Figura 1.6 ilustra a diversidade de sensores ópticos encontrados no mercado:

Figura 1.6 – Tipos de sensores ópticos



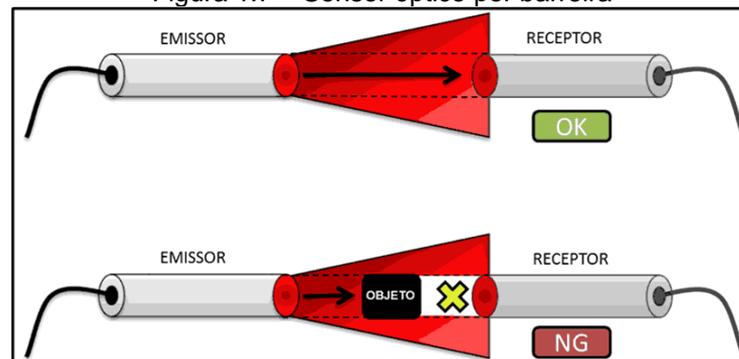
Fonte: djpautomacao.com, 2022

Os mesmos podem ser divididos em três tipos sendo eles: sistema por barreira, difusão e reflexão (CAPPELI, 2006).

Sistema por barreira: Utiliza-se dois sensores ópticos alinhados, onde um dispositivo é o emissor de luz e o outro alinhado frontalmente que recebe os feixes de luz denominado como receptor. Seu princípio de funcionamento baseia-se na interrupção do feixe entre o emissor e o receptor, no qual provoca uma ação chamada comutação eletrônica. Grande parte dos sensores de barreira utilizam luz proveniente da emissão de raios infravermelhos.

A Figura 1.7 ilustra o comportamento do sensor óptico por barreira quando não há objetos interrompendo o feixe de luz e, abaixo, quando há a interrupção:

Figura 1.7 – Sensor óptico por barreira



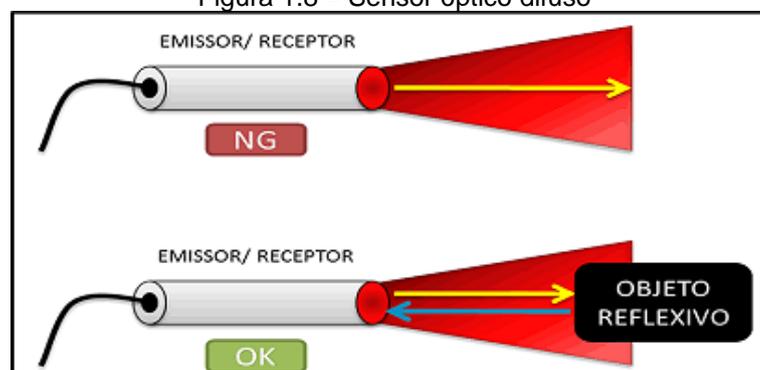
Fonte: www.citisystems.com.br, 2022

Desvantagem: Tem grande restrição com objetos transparentes, por apresentar falhas na detecção do objeto quando entra no range dos sensores.

Vantagens: Os objetos podem ser opacos ou pouco translúcidos; detectam objetos pequenos a longa distância, em ambientes sujos como pó, óleo, entre outros (THOMAZINI e ALBUQUERQUE, 2020).

Sistema por difusão: Utiliza-se dois sensores ópticos justapostos em um mesmo encapsulamento onde ambos são infravermelhos, sendo um emissor e o outro receptor, conforme a Figura 1.8. Os raios infravermelhos são emitidos pelo transmissor que reflete sob a superfície do objeto detectado e retorna em direção ao receptor, sendo assim o sensor difuso utiliza o próprio objeto como refletor. Seu range de detecção varia de acordo com a distância de comutação, onde a própria comutação provoca o chaveamento eletrônico do sensor.

Figura 1.8 – Sensor óptico difuso

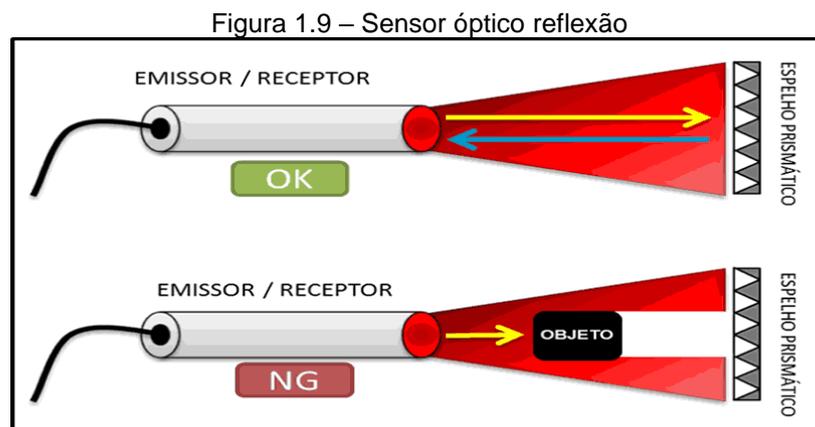


Fonte: www.citisystems.com.br, 2022

Desvantagens: A superfície do objeto detectado não pode ser totalmente fosca pois o sensor por difusão apresenta falhas na detecção. Outras variáveis são: cor, ângulo de impacto, forma e densidade.

Vantagens: Grande detecção com materiais metálicos; sensibilidade alta; menor custo de instalação e compra (LAMB, 2015).

Sistema por reflexão: O sensor óptico por reflexão tem seu funcionamento dado por meio da transmissão de um feixe de luz infravermelho emitido em direção a um espelho prismático. Quando o feixe de luz é cortado por algum objeto, a reflexão do feixe é interrompida entre o emissor e o espelho onde se permite reconhecer a presença de um objeto no mesmo instante, pois o receptor deixa de receber os sinais do emissor, conforme a Figura 1.9.



Fonte: www.citisystems.com.br, 2022

Desvantagens: Necessidade de um refletor externo; objetos que são totalmente transparentes não são detectados.

Vantagens: Instalação simples; objeto pode conter um certo grau de transparência; faixa de resposta maior que o sensor de barreira (THOMAZINI e ALBUQUERQUE, 2020).

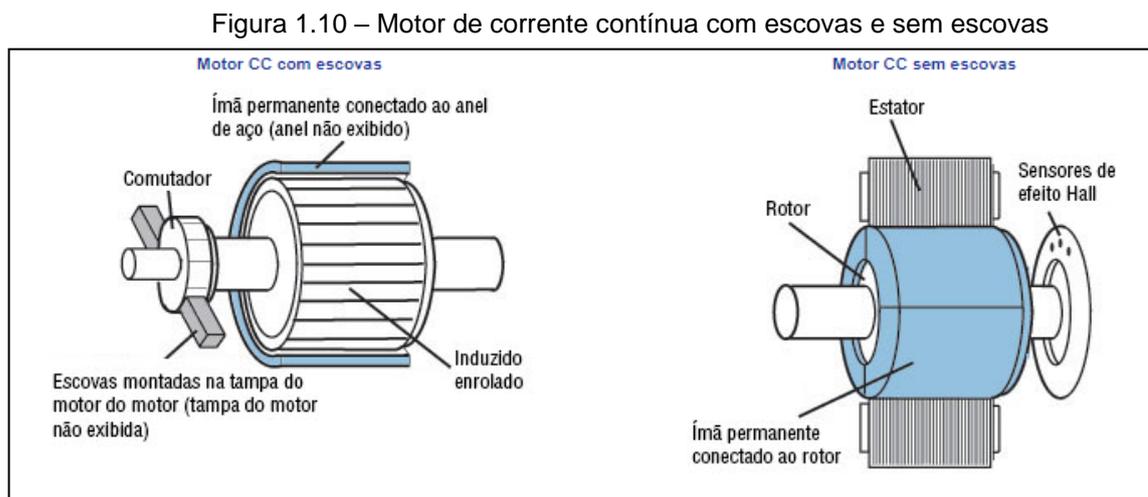
1.5 Motor de corrente contínua

Um motor de corrente contínua recebe esse nome pois é alimentado por uma corrente contínua que transforma energia elétrica em energia mecânica (CAPELLI, 2013 apud FERREIRA, 2019).

Esse tipo de motor é constituído por diversos equipamentos, onde os principais são o estator (enrolamento de campo ou ímã permanente), rotor (enrolamento de armadura), comutador e escovas.

Para que o motor comece a se mover, é preciso que o rotor inicie com um torque, sendo uma ação sobre um objeto fazendo com que ele rotacione. O estator gera um campo magnético que atravessa os enrolamentos de armadura. O comutador faz com que a corrente siga na mesma direção e por sua vez as escovas fazem o contato com esses enrolamentos introduzindo energia elétrica nele (HONDA, 2006).

A Figura 1.10 ilustra a diferença entre motores de corrente contínua com e sem escovas em sua estrutura.



Fonte: www.orientalmotor.com.br, 2022

1.5.1 Vantagens de um motor de corrente contínua

O motor de corrente contínua possui um ciclo contínuo, alto torque na partida sendo essas duas questões ocorrendo em baixas rotações. Além disso, é possível ocorrer variações da velocidade, porém com facilidade de controle (HONDA, 2006).

1.5.2 Desvantagens de um motor de corrente contínua

Os motores de corrente contínua costumam ser maiores e de custo elevado, tendo necessidade de realizar manutenções com maior frequência, além de possuírem restrições de aplicações, devido ao risco de arcos e faíscas que podem ocorrer no ambiente (HONDA, 2006).

1.6 Impressão 3D

A impressão 3D, também conhecida como manufatura aditiva, é uma família de processos que produz objetos em três dimensões, adicionando materiais em sucessivas camadas transversais de um modelo 3D (AUTODESK, 2022).

Com uma impressora 3D, um protótipo pode ser obtido através de desenhos feitos em algum *software* 3D, como o Inventor, e assim converter esse arquivo para um formato compatível com a máquina. Normalmente, os itens são feitos usando algum tipo de plástico como matéria-prima, mas outros materiais, incluindo metais, também podem ser usados.

Este tipo de equipamento funciona de forma simples, assim como uma impressora comum. Porém, ao invés de liberar jatos de tinta em quantidades e formatos capazes de formar imagens, as impressoras 3D injetam materiais para que assim o objeto seja formado (FIA BUSINESS SCHOOL, 2019).

1.6.1 Tipos de impressões

Para cada tipo de recursos e matéria-prima, utiliza-se técnicas de impressões diferentes e dentre as mais comuns pode-se citar: modelagem por fusão e depósito,

estereolitografia, sinterização seletiva a laser, sinterização direta de metal a laser e polyjet.

A técnica de modelagem por fusão e depósito (FDM), ilustrada na Figura 1.11, também conhecida como fabricação com filamento fundido, é uma das técnicas mais utilizadas devido ao seu baixo custo e relativa simplicidade. O processo utiliza filamentos plásticos para imprimir objetos. Funciona de maneira bem simples, os filamentos são aquecidos, enquanto um cabeçote se movimenta em dois eixos diferentes, injetando assim o material em uma bandeja. Em seguida ocorre a fusão do plástico até que se obtenha a forma desejada (FIA BUSINESS SCHOOL, 2019).

Figura 1.11 – Impressão 3D pelo método FDM



Fonte: www.marvin3d.com.br, 2022

1.6.2 GTMax3D Core A2v2

É um equipamento que oferece facilidade para todas as suas necessidades de impressão 3D, cria protótipos com baixo custo de produção, alta qualidade e rapidez (MAXCOMP, 2016).

1.6.3 Materiais para impressão

Com diversos tipos de filamentos usados nas impressoras 3D, pode-se criar objetos com propriedades completamente diferentes. Assim, com a mesma impressora, consegue-se obter desde peças decorativas até objetos usados em grandes projetos de engenharia. Podendo ter alta resistência mecânica e química,

além de aparência rústica. Dessa forma, pode-se citar dois materiais que são atualmente os mais utilizados pelas indústrias de impressão, sendo o filamento PLA (polilactídeo), e o Filamento ABS (Acrilonitrila Butadieno Estireno).

O PLA é um material de fácil impressão que possibilita a utilização em impressoras abertas ou fechadas, com a possibilidade de haver ou não uma mesa aquecida. Por possuir uma baixa contração, este material é indicado para aplicações onde há a necessidade de um dimensional controlado, ou peças que serão expostas à esforços de abrasão, pelo fato de possuir dureza superficial elevada, quando comparado com outros materiais. Isso faz com que ele seja muito rígido e, quando dobrado, se quebre com facilidade.

O ABS é um material muito utilizado nas indústrias. Possui uma resistência à temperatura e absorção de impactos, além de apresentar uma alta facilidade de acabamento após a impressão. Por ter a característica de dureza superficial baixa, se torna abrasivo e fácil de lixar. Diferente do PLA este material aceita maior deformação antes de romper. Ou seja, ao dobrá-lo ele vai demorar mais a romper (PORTELA, 2019).

1.6.4 *Software Cura*

O Cura 3D é um *software* fatiador para impressoras 3D. Ele é responsável por transformar um protótipo 3D em camadas que quando sobrepostas formam o objeto que será impresso.

O arquivo gerado é conhecido como G-Code, sendo o código que uma impressora 3D entende. Ou seja, é um programa que realiza a comunicação que transforma um arquivo do modelo criado em 3D para um formato de leitura da impressora (PORTELA, 2019).

1.7 Inventor

O Inventor é um programa desenvolvido pela Autodesk, onde o ambiente permite desenvolver protótipos de conjuntos e peças no plano tridimensional com ferramentas profissionais que simulam uma peça desenvolvida no mundo real.

A Figura 1.12 ilustra o ícone do *software* Inventor, disponível no *site* da Autodesk para acesso ao programa:

Figura 1.12 – *Software* Inventor



Fonte: www.autodesk.com.br, 2022

Suas principais características são a modelação paramétrica, modelação de montagens de conjuntos mecânicos, criação de desenhos, colaboração em vistas compartilhadas dentre outras funções essenciais para a criação de projetos (AUTODESK, 2022).

1.8 SketchUp

SketchUp é um *software* CAD (*Computer Aided Design* – Desenho Assistido por Computador). Atualmente pertence a empresa Trimble Navigation, utilizada para desenvolvimento de ambientes e objetos em três dimensões, podendo ser aplicada em diversas áreas de atuação como arquitetura e engenharia (RAZOR, 2019).

1.9 OIaclick

A plataforma OlaClick trata-se de uma *startup* gratuita que auxilia no processo de criação de menu digital de maneira prática, além de automatizar a gestão de pedidos, vendas e entregas. Após a criação do cardápio, personalizado pelo próprio estabelecimento, o acesso ao menu pode ser feito por *links* e *QR Codes*. Além do mais, o restaurante consegue ter uma gestão administrativa em tempo real de todos os pedidos feitos, monitorados através da aba “Histórico de Pedidos” (OLIVEIRA, 2022).

2 METODOLOGIA

Neste capítulo encontra-se a trajetória para o desenvolvimento e construção do projeto intitulado como Entrega de pedidos nos restaurantes com o uso de AGV, trata-se de uma pesquisa aplicada que é desenvolvida nas dependências da FATEC São Bernardo do Campo e nas residências das integrantes do grupo.

2.1 O que é Metodologia

Prodanov e Freitas (2013) destacam que a metodologia consiste em estudar, compreender e avaliar os métodos disponíveis para a realização da construção do protótipo. Enfatiza que os métodos são procedimentos amplos do raciocínio, enquanto as técnicas são procedimentos mais restritas que operacionalizam os métodos mediante emprego de instrumentos adequados.

Vianna (2001) enfatiza que qualquer pesquisa de cunho científico ou tecnológico possui sua base em fundamentos teóricos, com citações e sites de autores renomados, fazendo com que a teoria tenha um corpo de conhecimentos renomados.

Severino (2013) aponta que a metodologia é a preparação metódica e planejada de um trabalho científico ou tecnológico que se encontra inserida em uma sequência de etapas para sua construção, compreendendo o tema-problema e justificativa, levantamento bibliográfico, seleção da bibliografia, desenvolvimento da construção do projeto e redação do texto.

A construção da redação do TCC tem como base o Manual de Normalização de Projeto de Trabalho de Graduação da FATEC – SBC (2017) que se encontra embasado nas normas da ABNT.

2.2 O tema-problema

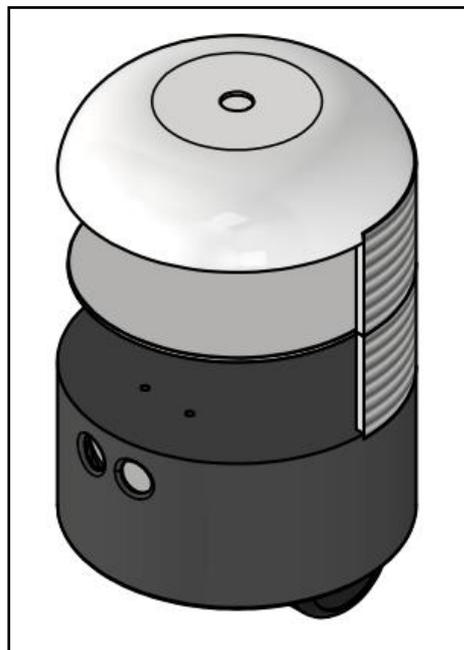
O tema-problema surgiu após uma visita em um restaurante, onde era possível realizar o pedido de forma completamente digital, através de *tablets*. Nos dias atuais,

o homem busca formas de utilizar a tecnologia como forma de facilitar a realização de tarefas do cotidiano.

Os AGVs são amplamente utilizados nas indústrias. Eles são equipamentos relativamente simples e de baixo custo, capazes de realizar o transporte de peças e objetos, entre pontos pré-definidos infinitas vezes.

Diante disso, podemos notar uma certa semelhança entre esse tipo de robô e o trabalho dos garçons nos restaurantes, que realizam um trabalho repetitivo, andando sempre de um ponto ao outro para transportar as refeições. Assim, ao implementar um robô para realizar esta função, conseguimos reduzir o desgaste físico do ser humano, além de reduzir os custos com mão de obra e proporcionar um atendimento padrão a todos os clientes. Para ilustrar a ideia, foi desenvolvido um protótipo de como ficará o projeto pela plataforma Inventor, conforme mostra a Figura 2.1.

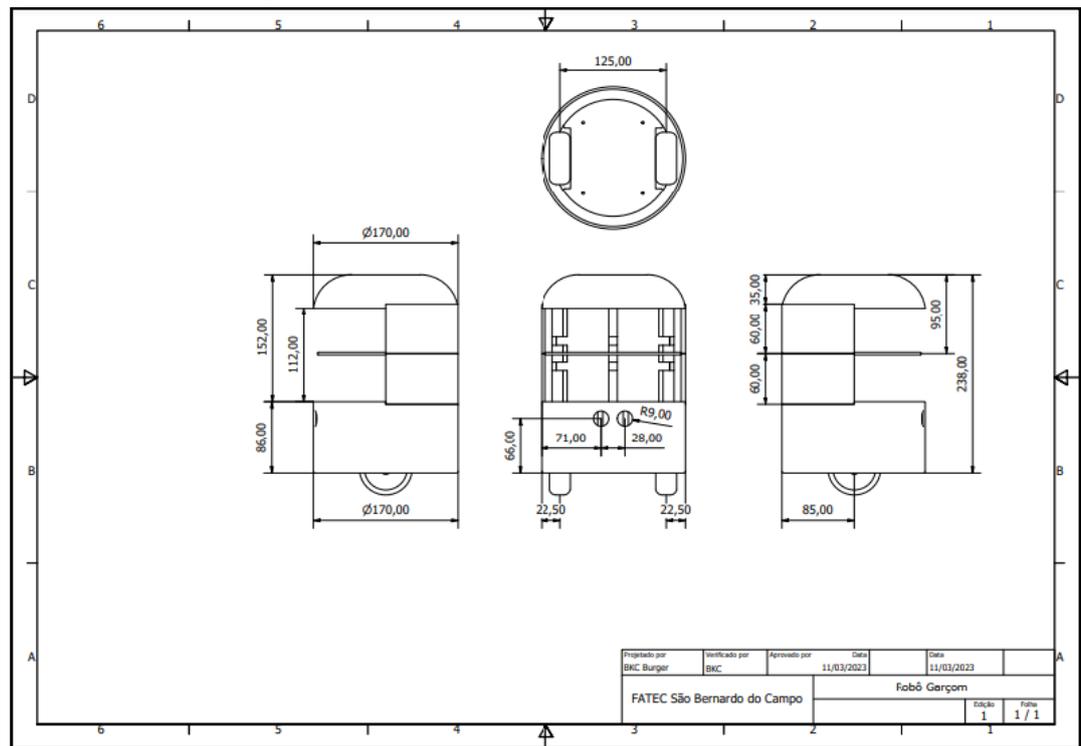
Figura 2.1 – Protótipo do Robô AGV



Fonte: Autoria própria, 2022

A Figura 2.2 mostra o protótipo com sua dimensão em plano bidimensional, também desenvolvido no *software* Inventor na qual sua construção será realizada pela impressora GTMax3D Core A2v2.

Figura 2.2 – Protótipo do Robô AGV cotado



Fonte: Autoria própria, 2022

2.3 Justificativa

O presente projeto tem por justificativa a automatização da logística de entrega de refeições dentro dos restaurantes, melhorando a qualidade de trabalho e ergonomia do funcionário, além de trazer maior retorno financeiro aos proprietários.

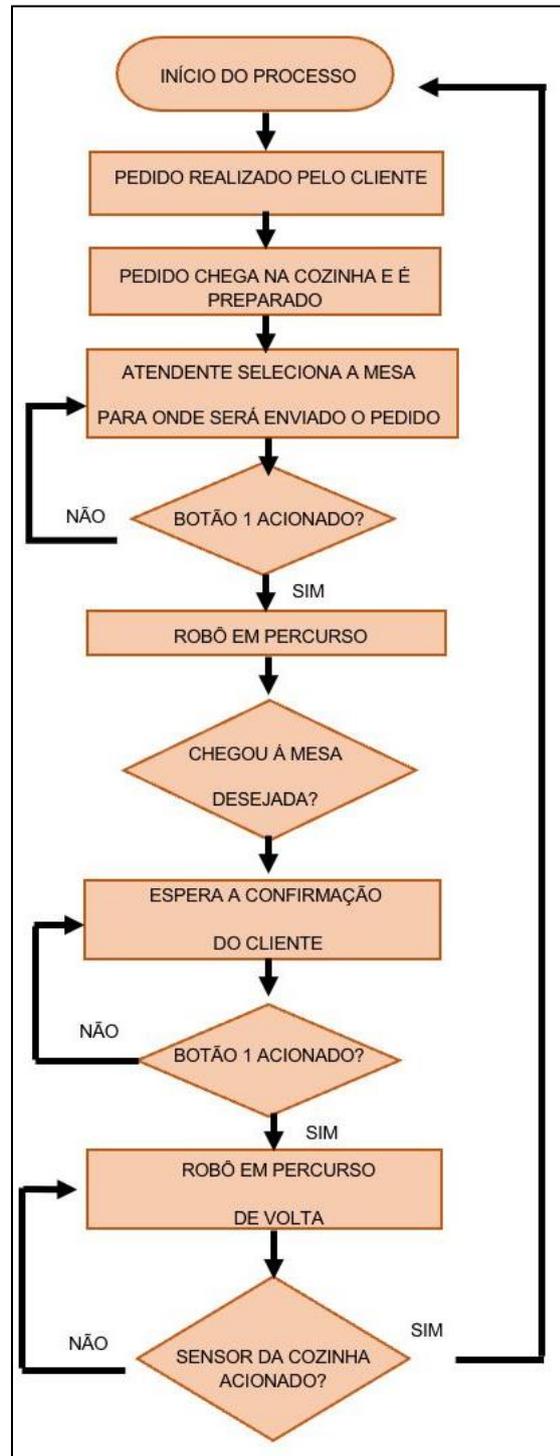
Essas vantagens ocorrerão, pois haverá a redução do número de funcionários destinados a registrar o pedido do cliente e levar até a cozinha. Além de que o colaborador não precisará realizar a entrega do pedido à mesa.

Contudo, o atendente poderá ser solicitado quando houver dificuldades para realização do pedido ou se ocorrer algum problema de logística, caso o robô apresente falhas durante o trajeto. Sendo assim, os funcionários teriam mais tempo para se dedicar em outros afazeres do estabelecimento.

2.4 Fluxogramas

Para uma visualização geral do programa, o fluxograma representa a estrutura de funcionamento do projeto, conforme ilustra a Figura 2.3.

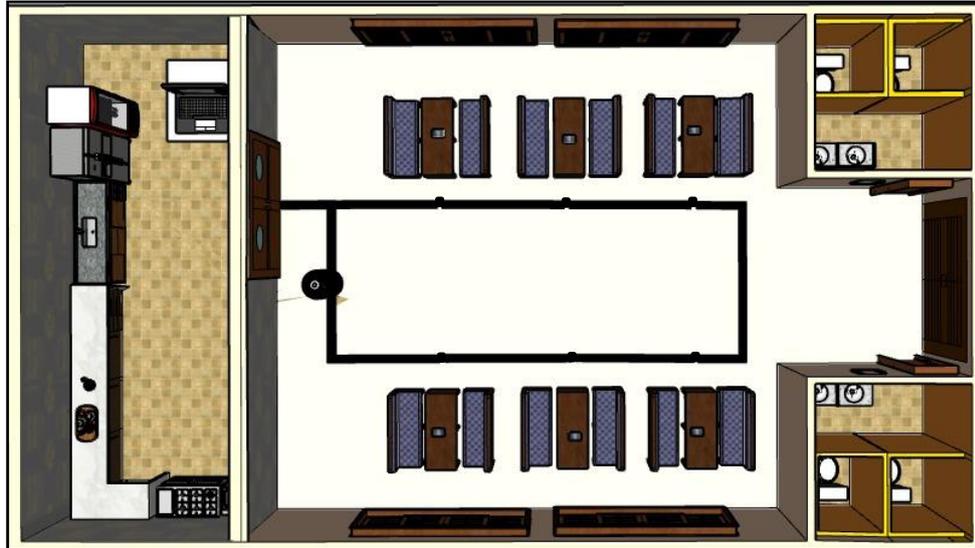
Figura 2.3 – Fluxograma geral do processo



Fonte: Autoria própria, 2022

Inicialmente o cliente entrará no restaurante, ilustrado seu *layout* pela Figura 2.4 desenvolvido através do *software* SketchUp, e escolherá a mesa para se acomodar.

Figura 2.4 – *Layout* do restaurante



Fonte: Autoria própria, 2022

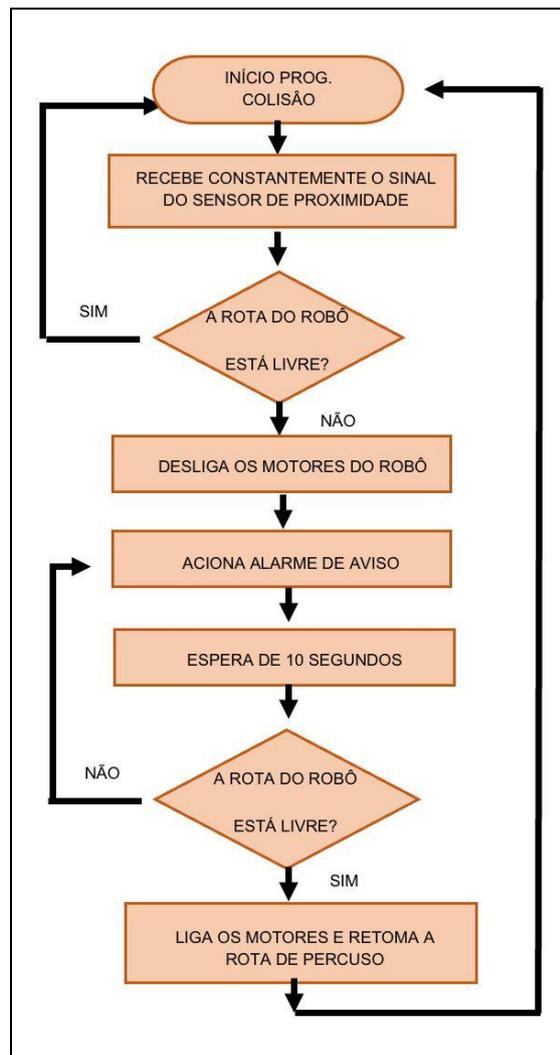
Em cada mesa haverá um *tablet* que possui um menu digital com endereçamentos diferentes, com o objetivo de informar à cozinha, na hora da entrega do pedido, a mesa correspondente. Dessa forma, após ocorrer a finalização do pedido, ele é visualizado na cozinha através de um computador, logado no cadastro do estabelecimento da plataforma OlaClick na aba “Pedidos do Dia”. Quando a refeição estiver pronta, o atendente a coloca no AGV e seleciona o número da mesa para que o robô siga com o pedido, sendo por botões ou através de um *smartphone*. Contudo, ele só seguirá após o funcionário acionar o “botão iniciar” para que o pedido chegue até a mesa desejada.

Ao chegar na mesa, é esperado a confirmação de que o pedido foi retirado pelo cliente da bandeja, pressionando o mesmo botão. Logo após isso, o robô volta para a cozinha, estando pronto para a próxima entrega.

Em paralelo a programação principal, responsável pela realização do trajeto do robô, há dois subprogramas.

A Figura 2.5 representa o fluxograma do subprograma responsável por realizar a monitoração constante de que o caminho a ser percorrido não está obstruído, tanto por pessoas quanto por objetos. Caso o sensor detecte algo em sua frente, ele mandará um sinal positivo que no exato momento desligará os dois motores, fazendo assim com que o robô pare naquele determinado ponto. Logo depois, será emitido um alarme sonoro para que os funcionários do restaurante percebam e corrijam o ocorrido. Dessa forma, a programação é parada por dez segundos para que haja a verificação novamente, através do sensor, se a passagem está livre. Caso a passagem ainda estiver obstruída, ela entra em um ciclo até que o problema esteja resolvido, e quando a passagem estiver livre, através da programação, é acionado os motores e a rotina principal é retornada.

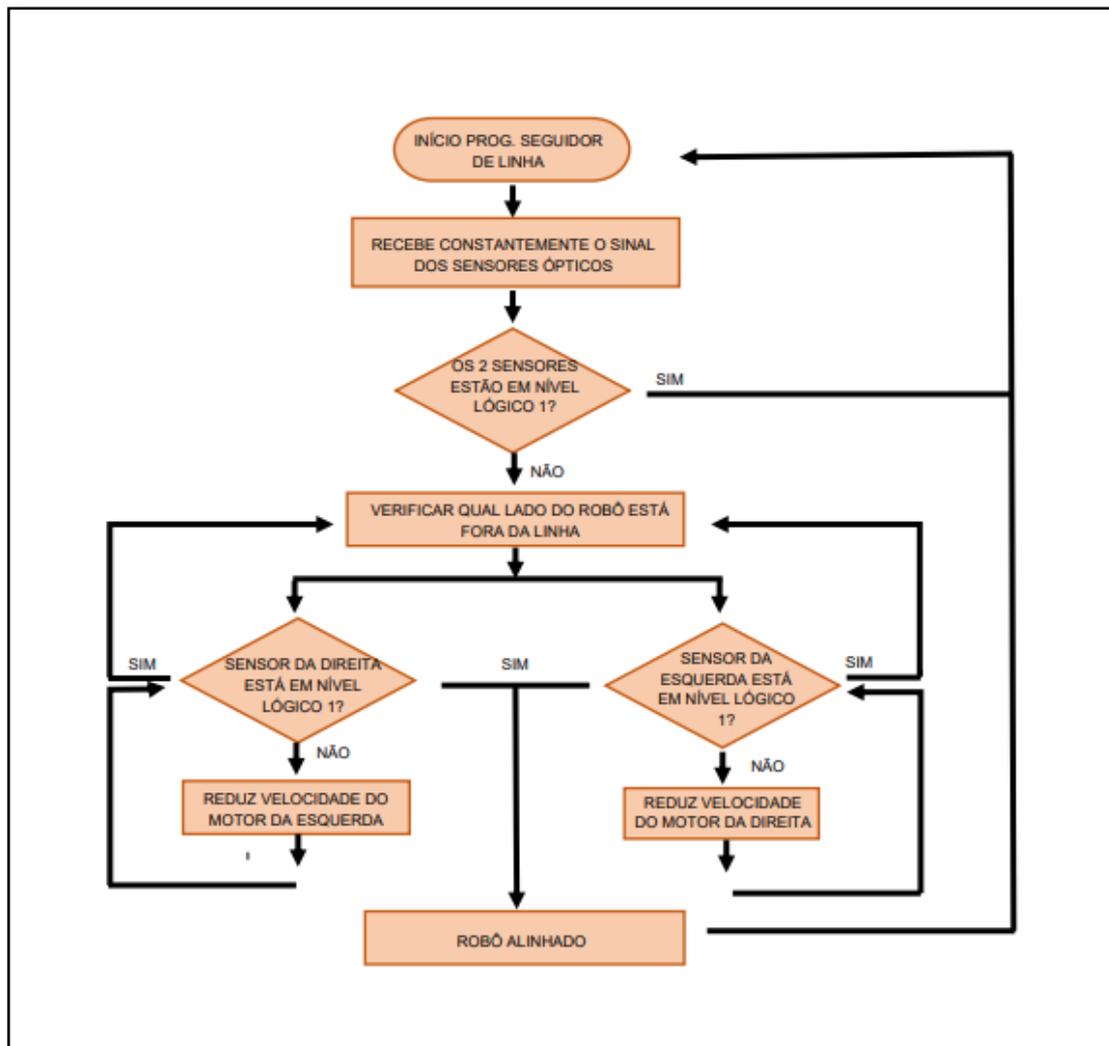
Figura 2.5 – Fluxograma do subprograma de colisão



Fonte: Autoria própria, 2022

A Figura 2.6 representa o subprograma responsável por manter o robô no percurso, seguindo a linha colocada no chão do estabelecimento. O robô conta com dois sensores ópticos, um localizado em sua extremidade esquerda e outro na direita. Estes sensores, quando o robô está alinhado, se encontram em estado com nível lógico “1”. Assim quando, por exemplo, ele detectar que o sensor da direita está em nível lógico “0”, significa que o robô está fora do percurso, inclinado para a esquerda. Logo, é necessário realizar o movimento de giro, onde é preciso aumentar a velocidade do motor da esquerda em relação ao motor da direita. E seguindo a mesma lógica, com os comandos invertidos caso esteja inclinado para a direita, ele realizará o movimento de giro até que o respectivo sensor chegue em nível lógico “1”, deixando assim o robô alinhado a linha e seguindo seu percurso.

Figura 2.6 – Fluxograma do subprograma do seguidor de linha



Fonte: Autoria própria, 2022

2.5 Etapas teóricas e práticas para o desenvolvimento do projeto

Após a descrição do projeto, será detalhado as etapas nas quais foram divididas com o intuito de se obter um cronograma como auxílio para o grupo.

Primeira etapa: reunião com as integrantes do grupo para escolha do tema e do orientador. Após isso, foi feito o convite na qual o orientador aceitou a participação no projeto.

Segunda etapa: reunião do grupo com o orientador para explicação detalhada do projeto e apresentação de dúvidas sobre métodos de pesquisa e referências. Foi acordado, obrigatoriamente, uma reunião por semana, nas segundas-feiras as 18h.

Terceira etapa: levantamento de objetivos e cronograma de atividades a serem desenvolvidas no segundo semestre de 2022, conforme a tabela 2.1.

Tabela 2.1 – Cronograma do segundo semestre de 2022

CRONOGRAMA DE ATIVIDADES					
Semanas	Seleção das fontes de pesquisa	Elaboração dos elementos pré e pós-textuais	Concepção da introdução	Criação dos slides	Apresentação do TCC à banca
3º agosto					
4º agosto					
1º setembro					
2º setembro					
3º setembro					
4º setembro					
1º outubro					
2º outubro					
3º outubro					
4º outubro					
1º novembro					
2º novembro					
3º novembro					
4º novembro					

Fonte: Autoria própria, 2023

Quarta etapa: levantamento bibliográfico em livros na biblioteca da FATEC SBC, em sites especializados e artigos encontradas na *internet*. Eles foram lidos e selecionadas aquelas teorias que se adequaram ao tema do projeto, encontrados no Capítulo 1 – Fundamentação Teórica.

Quinta etapa: elaboração do esboço do protótipo e do *layout* a ser seguido pelo robô. Desenvolvimento do esquemático de como funcionará a programação, identificando a dinâmica do local que será implantado o projeto, encontrados no Capítulo 2 – Metodologia.

Sexta etapa: levantamento do cronograma de atividades a serem desenvolvidas no primeiro semestre de 2023, conforme a tabela 2.2.

Tabela 2.2 – Cronograma do primeiro semestre de 2023

CRONOGRAMA DE ATIVIDADES						
Semanas	Ajustes desenho 3D para a impressão	Criação da programação do Arduino	Testes e pré montagem do AGV	Ajustes na montagem do projeto	Elaboração dos elementos textuais, slides e banner	Apresentação do TCC à banca
3º fevereiro						
4º fevereiro						
1º março						
2º março						
3º março						
4º março						
1º abril						
2º abril						
3º abril						
4º abril						
1º maio						
2º maio						
3º maio						
4º maio						
1º junho						
2º junho						
3º junho						
4º junho						

Fonte: Autoria própria, 2023

Sétima etapa: levantamento dos materiais e componentes que são utilizados na construção do projeto. Pesquisa em sites e lojas especializadas, buscando a melhor viabilidade de preço. Aquisição dos materiais conforme Tabela 2.3.

Tabela 2.3 – Materiais utilizados à confecção do projeto

Produto	Qtde.	Valor em reais
Kit chassi para robô duas rodas circular	1	73,00
Ponte H L298N	1	18,00
Clip/conector para bateria 9V	1	1,50
Módulo seguidor de linha TCRT5000	2	20,00
Bateria 9 V	1	15,00
Bateria 12 V	1	50,00
Sensor ultrassônico HC-SR04	1	15,00
Módulo sensor magnético	1	10,00
Pulsador botão push botton	4	10,00
Botão switch	1	2,00
Arduino Uno	1	80,00
<i>Shield</i> borne Arduino Uno	1	20,00
Módulo buzzer 5V	1	7,00
HC-05	1	30,00
Resistor 3k3 Ω	1	1,00
Resistor 5k6 Ω	1	1,00
Cola adesiva Fixador 3D	1	30,00
TOTAL		383,50

Fonte: Autoria própria, 2023

Oitava etapa: na parte mecânica é impresso e montado todas as peças para construção do robô juntamente com o chassi. Na parte eletrônica faz-se a ligação de todos os componentes eletrônicos ao Arduino.

Nona etapa: desenvolvimento da programação do Arduino.

Décima etapa: construção do menu digital pela plataforma Olaclick para automatização de pedidos.

Décima primeira etapa: integração das partes mecânica, eletrônica, programação e teste final de funcionamento.

Décima segunda etapa: obstáculos de soluções. Concluído o desenvolvimento do projeto fazem-se as Considerações finais, Resumo e Abstract.

3 DESENVOLVIMENTO DO PROJETO

Neste capítulo encontra-se passo a passo o desenvolvimento e construção do projeto que se intitula Entrega de pedidos nos restaurantes com o uso de AGV.

3.1 Projeto finalizado

Para melhor compreensão e visualização do projeto a Figura 3.1 ilustra-o finalizado.

Figura 3.1 – Projeto finalizado



Fonte: Autoria própria, 2023

Seu funcionamento dá-se da seguinte maneira: com o aplicativo “*Bluetooth for Arduino*” baixado em um *smartphone*, faz-se a conexão via *bluetooth* do módulo HC-05, responsável por criar a interface do *smartphone* com o Arduino, para enviar pela tela do *smartphone* a mesa que será selecionada. O Arduino recebe e processa essa informação vinda do módulo HC-05 ou dos botões respectivos de cada mesa e assim que o botão principal for acionado, os motores, controlados pela Ponte H, são acionados, fazendo com que o AGV inicie seu percurso. O AGV é guiado através dos sinais vindos dos sensores ópticos e com o auxílio do sensor ultrassônico que verifica

se o caminho não está obstruído. Para a detecção dos pontos de parada, utiliza-se o sensor magnético.

O desenvolvimento e construção do projeto se encontra amparado nos seguintes tópicos:

- desenvolvimento da montagem do robô;
- programação do microcontrolador;
- criação do menu digital na plataforma OIaclick;
- integração das partes e testes de funcionamento;
- obstáculos e soluções.

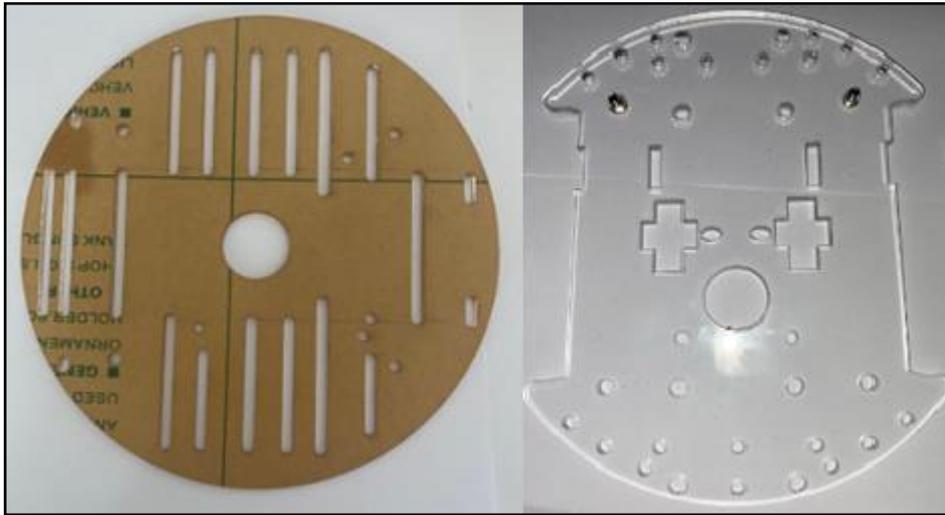
3.2 Desenvolvimento da montagem do robô

O desenvolvimento da montagem do robô pode ser subdividido em três etapas. Primeiro faz-se a montagem do chassi, parte responsável por acomodar todos os componentes eletrônicos, motores e as rodas. Em seguida faz-se a montagem da carcaça feita em impressão 3D, e por fim realiza-se a montagem de toda a parte eletrônica, onde é feita as devidas ligações entre os componentes.

3.2.1 Montagem do chassi

O chassi é composto por duas bases redondas em acrílico, medindo 140 mm de diâmetro e 3 mm de espessura. A base que se localiza na parte inferior possui 2 rasgos de 75 mm nas laterais para o encaixe das rodas. Na Figura 3.2 pode-se observar as bases do chassi.

Figura 3.2 – Bases em acrílico do chassi



Fonte: Autoria própria, 2023

Para que ocorra a fixação entre as duas bases, utiliza-se quatro espaçadores em nylon que possuem 6 mm de diâmetro por 30 mm de altura e com rosca M3, fixados com parafusos M3 x 6 mm. A Figura 3.3 ilustra a base inferior com os espaçadores fixados.

Figura 3.3 – Base com espaçadores

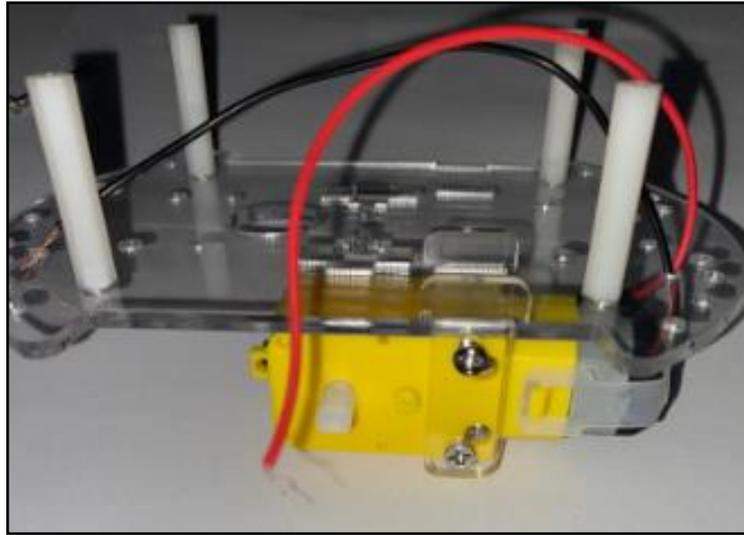


Fonte: Autoria própria, 2023

A montagem dos motores faz-se através do acoplamento deste componente com duas travas em acrílico, com o auxílio de dois parafusos M3 x 30 mm que passam pelos furos passantes encontrados nas laterais dos motores e são presos com porcas

pelo lado interior. Este processo é repetido duas vezes. A Figura 3.4 ilustra os motores fixados com a base inferior.

Figura 3.4 – Motor acoplado na base



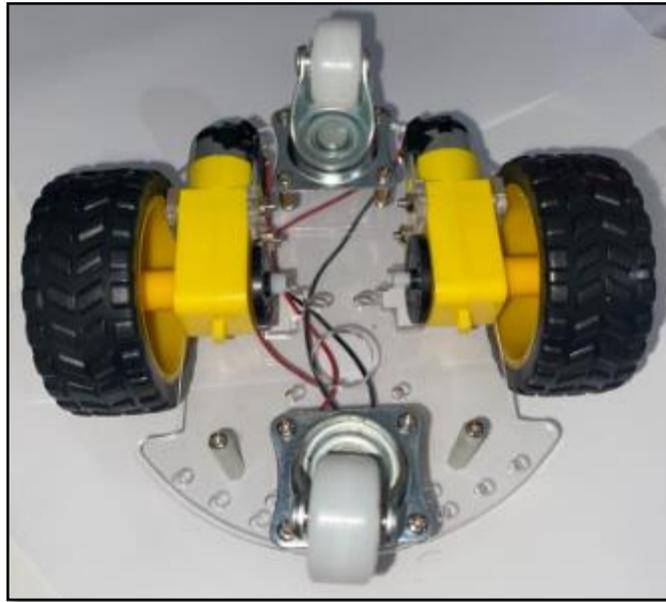
Fonte: Autoria própria, 2023

As rodas principais possuem diâmetro de 60 mm e são encaixadas em uma extremidade do eixo do motor e na outra extremidade é colocada uma trava.

O chassi também possui duas rodas adicionais que não são controladas pelo motor, chamadas “rodas bobas “. Elas possuem 25 mm de diâmetro e são montadas de forma paralela ao centro da base para que assim haja estabilidade ao robô.

As “rodas bobas” são montadas entre 4 espaçadores que possuem 10 mm de altura, para que fiquem alinhadas ao chão, e a base inferior. Assim, elas se fixam através de parafusos M3. A Figura 3.5 ilustra a base do chassi com a parte mecânica montada.

Figura 3.5 – Rodas montadas na base inferior



Fonte: Autoria própria, 2023

3.2.2 Impressão 3D

O conjunto de peças impressas e que constam nos Apêndices A ao E dividem-se em cinco partes, sendo elas: base principal, cúpula inferior, cúpula superior, prateleira e tampa. Todas as peças foram impressas em material PLA.

Conforme a tabela 3.1, observa-se o tempo e quantidade de material gasto referente a cada peça. Os dados fornecidos foram obtidos pelo *software* Cura.

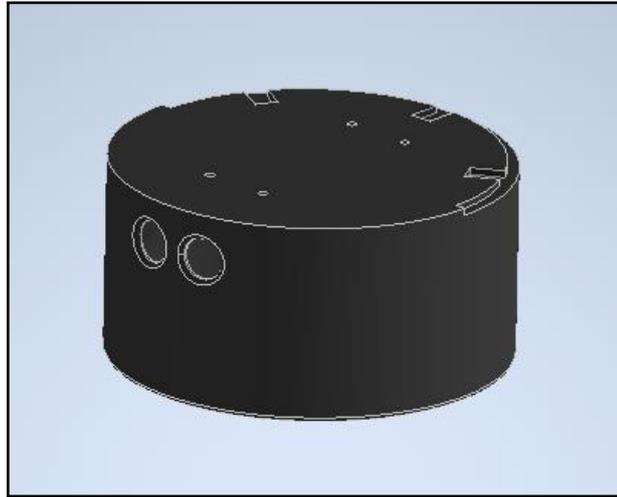
Tabela 3.1 – Indicação do tempo de impressão e peso das peças

PEÇA	TEMPO	PESO (Gramas)
Base principal	17 h, 4 min	298
Cúpula inferior	7 h, 50 min	115
Cúpula superior	7 h, 50 min	115
Prateleira	2 h, 50 min	65
Tampa	7 h, 33 min	146
TOTAL	43 h ,7 min	739

Fonte: Autoria própria, 2023

Para a sustentação e apoio das cúpulas, utiliza-se a base principal, possuindo diâmetro de 170 mm x 86 mm de altura x 3 mm de espessura, como ilustra a Figura 3.6.

Figura 3.6 – Base principal



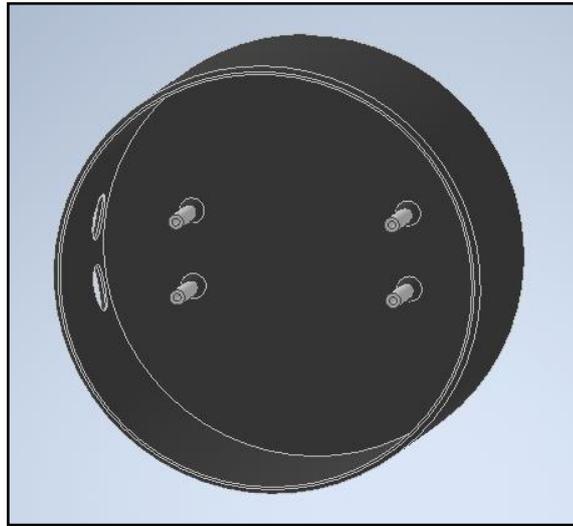
Fonte: Autoria própria, 2023

A peça possui dois furos de raio 9 mm na sua parte frontal e encontram-se deslocados a 14 mm do centro e 20 mm da face superior, possuindo a finalidade de alojar o sensor ultrassônico.

Na sua face superior, estão localizados quatro furos de 4 mm de diâmetro para a fixação dos espaçadores. Ademais, possui um rasgo para alojar a cúpula inferior. Sua geometria conta com um rebaixo de 3 mm, seguindo o seu diâmetro de 170 mm, e feixes de 10 mm de comprimento para o encaixe das aletas da cúpula inferior.

Para que ocorra a fixação da base principal impressa com a base do chassi, utiliza-se quatro espaçadores de nylon com diâmetro de 6 mm por 30 mm de altura com rosca M3, fixados com parafusos M3 x 6 mm. A Figura 3.7 ilustra a fixação dos espaçadores na base principal.

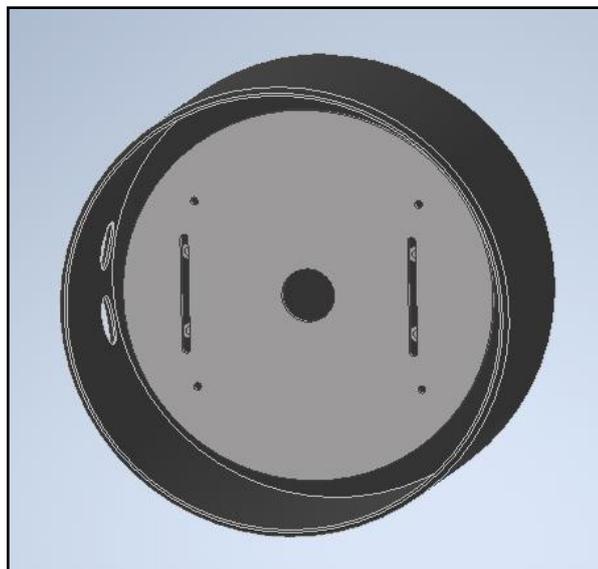
Figura 3.7 – Base principal com espaçadores



Fonte: Autoria própria, 2023

A Figura 3.8 ilustra como encontra-se as duas bases montadas e fixadas entre si.

Figura 3.8 – Base Principal fixada com base do chassi

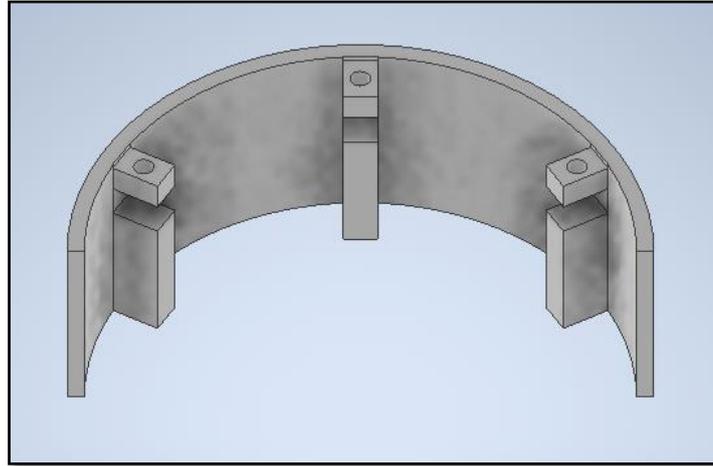


Fonte: Autoria própria, 2023

A cúpula inferior possui a finalidade de fazer a proteção e armazenamento dos alimentos transportados. Ela possui suas dimensões baseadas na base principal sendo assim, 170 mm de diâmetro x 60 mm de altura x 5 mm de espessura.

Para uma melhor rigidez, seu dimensional apresenta 3 aletas de reforço com espessura de 10 mm por 15 mm de comprimento a partir de sua face. Seus reforços possuem 3 furos para a fixação da cúpula superior, como ilustra a Figura 3.9.

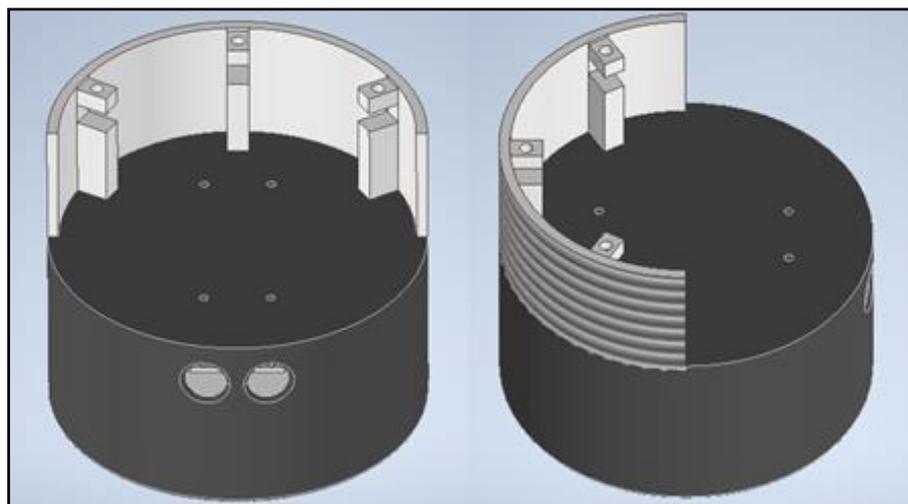
Figura 3.9 – Cúpula inferior



Fonte: Autoria própria, 2023

A fixação da cúpula inferior na base principal faz-se através do encaixe de sua forma – através do rasgo já existente na base que acompanha a geometria da cúpula inferior – e a adição de cola adesiva Fixador 3D. A Figura 3.10 ilustra a junção dos dois corpos em vista frontal e vista lateral do conjunto.

Figura 3.10 – Montagem cúpula Inferior e base principal

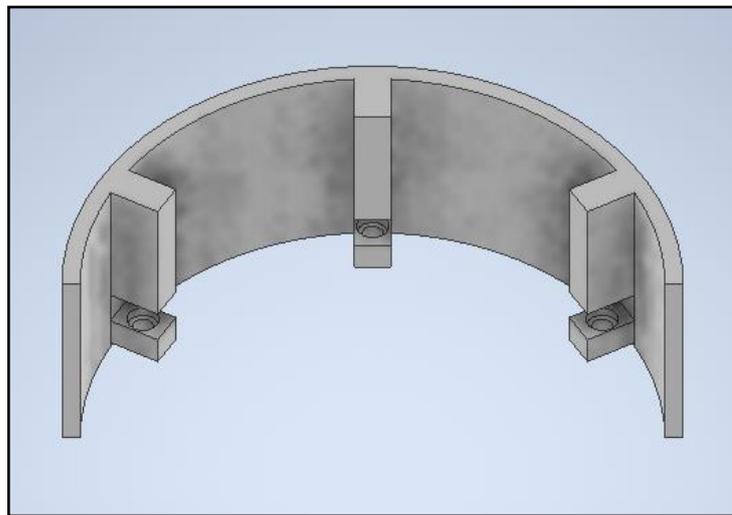


Fonte: Autoria própria, 2023

A cúpula superior possui a finalidade de fazer a proteção e armazenamento dos alimentos, assim como a cúpula inferior. Ela possui as mesmas dimensões da cúpula inferior sendo assim, 170 mm de diâmetro x 60 mm de altura x 5 mm de espessura.

Para uma melhor rigidez, seu dimensional conta com 3 aletas de reforço com espessura de 10 mm e 15 mm de comprimento a partir de sua face. Seus reforços possuem 3 furos passantes de 6 mm de diâmetro para a fixação a cúpula inferior.

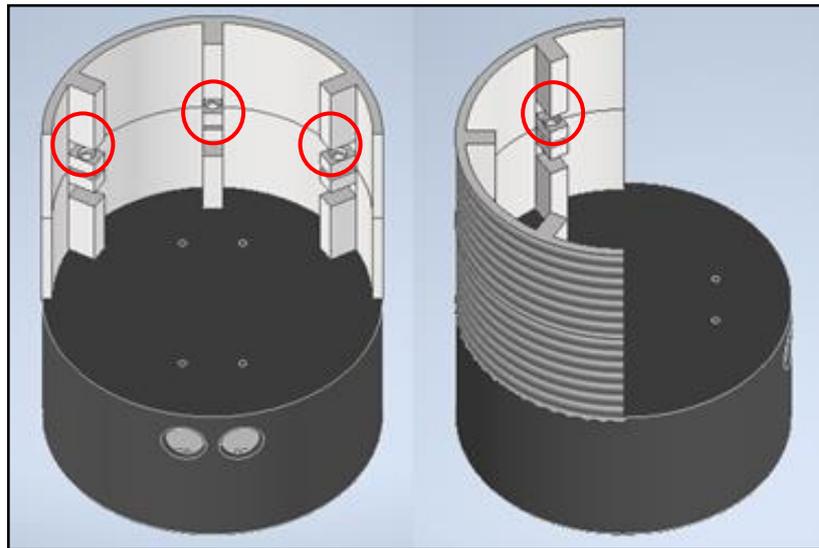
Figura 3.11 – Cúpula superior



Fonte: Autoria própria, 2023

A união da cúpula superior ao restante do conjunto é realizada através de cavilhas de madeira com diâmetro de 6 mm x 30 mm de comprimento. Para uma melhor fixação da montagem, utiliza-se cola adesiva Fixador 3D. A Figura 3.12 ilustra os pontos de montagem das cavilhas em vista frontal e lateral.

Figura 3.12 – Pontos de montagem das cavilhas

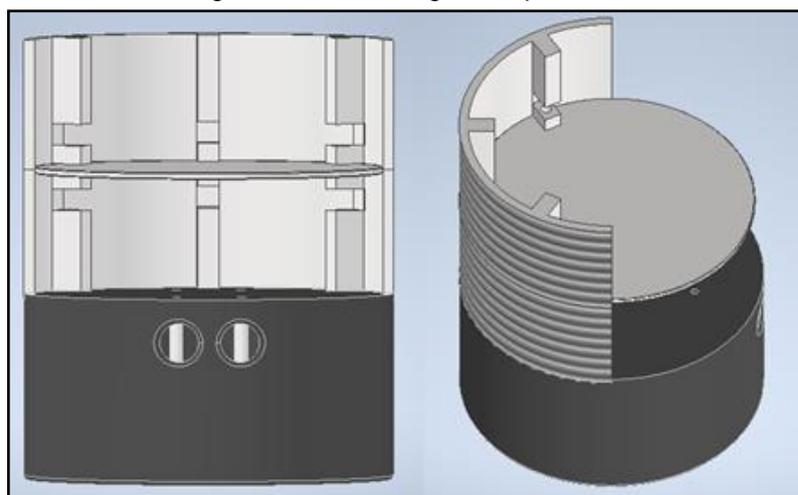


Fonte: Autoria própria, 2023

A prateleira tem por finalidade acomodar o item a ser transportado pelo AGV. Suas dimensões são 160 mm de diâmetro e 3 mm de espessura. Ademais, possui 3 furos passantes de 6,2 mm para passagem das cavilhas.

Sua fixação ao conjunto faz-se através de um rasgo já existente entre a montagem das cavilhas, que devem transpassar os furos da prateleira, e a utilização da cola adesiva Fixador 3D. A Figura 3.13 ilustra a montagem em vista frontal e lateral.

Figura 3.13 – Montagem da prateleira

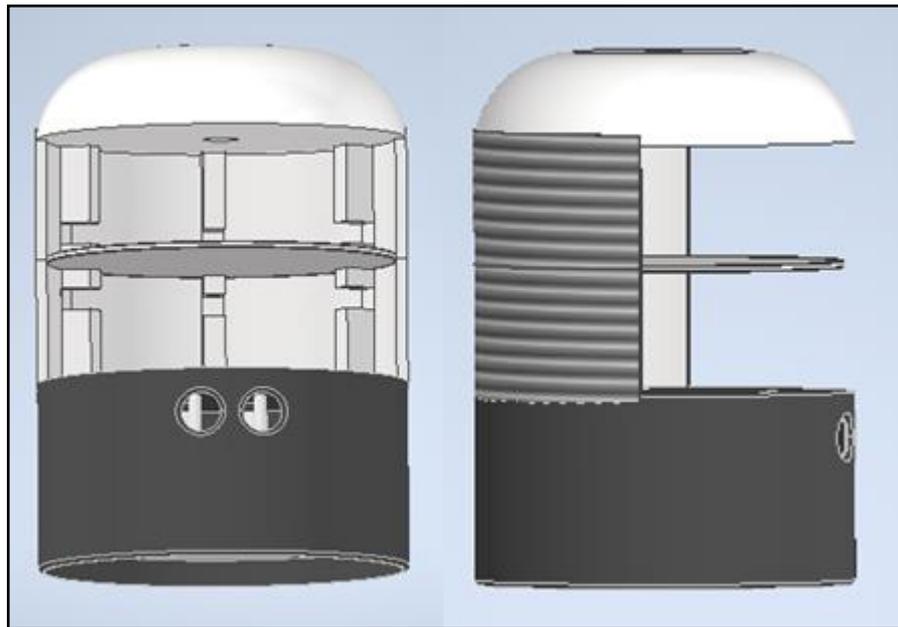


Fonte: Autoria própria, 2023

A tampa possui a finalidade de cobrir e dar o *designer* ao projeto. Suas dimensões são de 170 mm de diâmetro e 40 mm de espessura. Além disso, possui

um rasgo para alojar a cúpula superior e sua geometria conta com um rebaixo de 3 mm, seguindo o seu diâmetro de 170 mm, e feixes de 10 mm de comprimento para o encaixe das aletas da cúpula superior. Para a fixação, utiliza-se cola adesiva Fixador 3D. A Figura 3.14 ilustra a montagem em vista frontal e lateral.

Figura 3.14 – Montagem completa



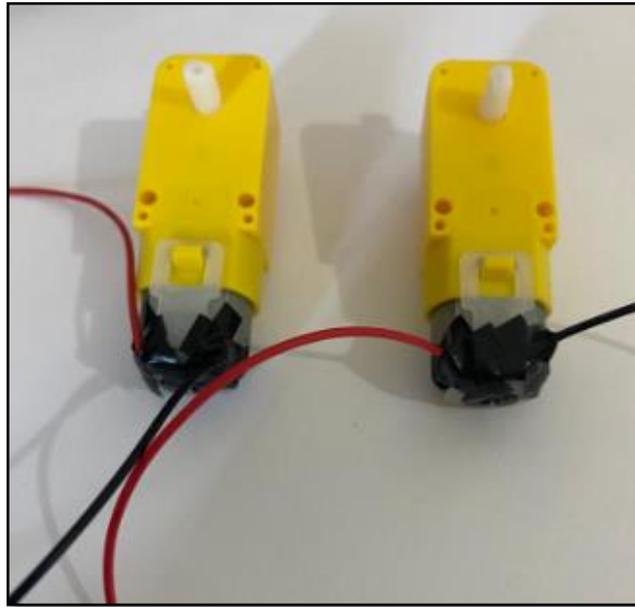
Fonte: Autoria própria, 2023

3.2.3 Montagem eletrônica

A conexão dos motores faz-se desencapando a ponta de dois fios, onde indicam o lado positivo e negativo. Assim, com o cobre a mostra, cada fio se conecta em um pino nas laterais que são localizados na parte inferior do motor.

Para essa conexão, não é necessário o uso de soldagem. Dessa forma, faz-se necessário apenas o isolamento onde se desencapou os fios para que não ocorra acidente e que não haja curto circuito. A Figura 3.15 ilustra como foi feito o isolamento dos fios dos motores, utilizando fita isolante.

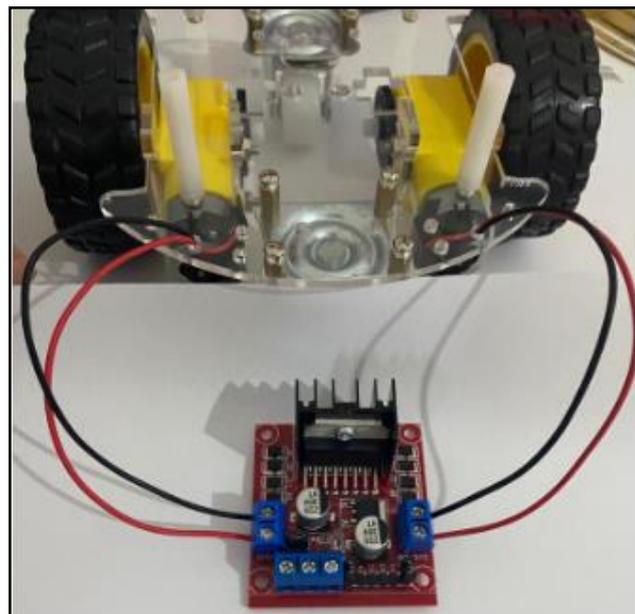
Figura 3.15 – Motores com fiação isoladas



Fonte: Autoria própria, 2023

Utiliza-se o módulo L298N para controle dos motores. Os fios, negativo e positivo vindos dos motores, são ligados aos pinos do módulo denominados: OUT 1, OUT2, OUT3 e OUT4. Para a alimentação da ponte H, o pino indicado como “12 V” é ligado ao fio positivo e o pino “GND” é ligado ao fio negativo, vindo da bateria. A Figura 3.16 ilustra a conexão de alimentação do módulo e a conexão dos motores.

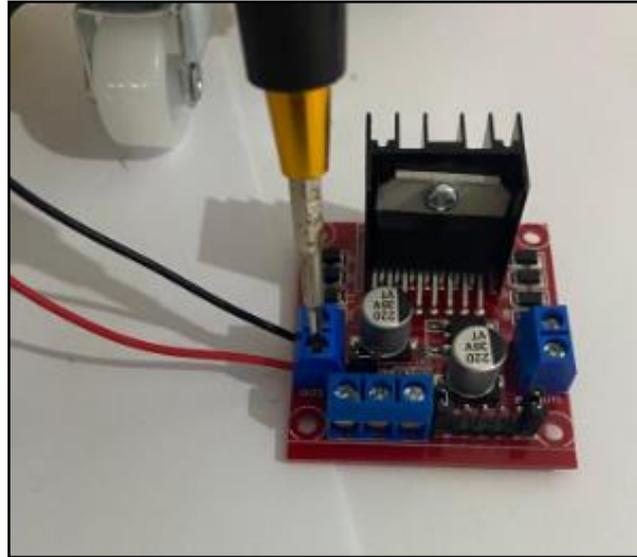
Figura 3.16 – Conexões do módulo L298N



Fonte: Autoria própria, 2023

A ligação dos fios ao módulo L298N faz-se feita através de bornes, fixando os de forma mecânica, apertando os parafusos encontrados no módulo com auxílio de uma chave Philips. A Figura 3.17 ilustra como é realizada esta conexão.

Figura 3.17 – Conexão no borne do L298N



Fonte: Autoria própria, 2023

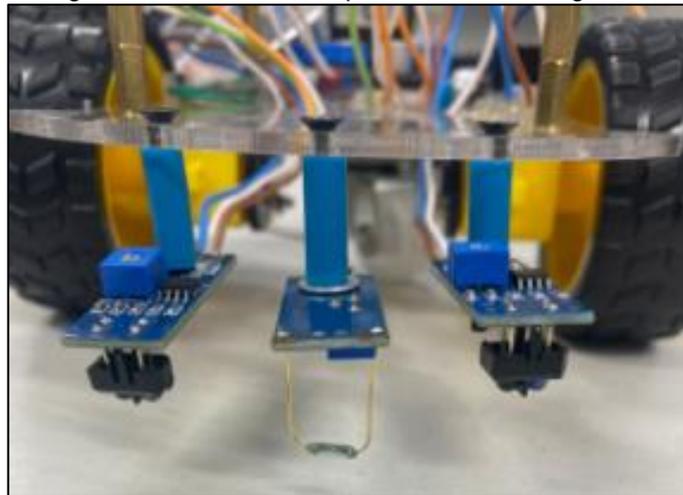
Nos pinos de entrada do L298N fazem-se as ligações com o Arduino, denominadas como N1, N2, N3 e N4. Dessa forma, faz-se a ligação dos pinos N1 e N2 nos pinos 6 e 7, respectivamente, do microcontrolador para controlar o motor da esquerda, e N3 e N4 nos pinos 12 e 13 para o controle do motor da direita. A ponte H conta com a função PWM, utilizada para o controle de velocidade dos motores, para isso é necessário retirar o jumper dos pinos ENA e ENB e conecta-los em uma entrada PWM do microcontrolador. A velocidade do motor pode ser variada em valores de 0 a 255.

O sensor ultrassônico é conectado através de quatro pinos ligados ao Arduino. O pino indicado como “Vcc” é ligado ao pino “5 V” e o pino “GND” é conectado ao “GND”, ambos do Arduino. Os pinos denominados como “TRIG” e “ECHO” do sensor são ligados nos pinos A4 e A5 do Arduino. Este sensor é colocado na parte frontal do robô, afim de detectar pessoas ou objetos que possam estar obstruindo o caminho. Na programação é determinado uma distância de 15 cm para que ele detecte e pare sua movimentação até que o caminho seja desobstruído.

Os módulos dos sensores ópticos conectam-se de forma similar ao sensor ultrassônico. Porém, os módulos dos sensores ópticos utilizam-se apenas três pinos. O pino “Vcc” é conectado ao pino “5 V” e o pino “GND” é conectado ao “GND”, ambos do Arduino. Esse módulo possui um pino de entrada digital e um pino de entrada analógica. No projeto faz-se a ligação das entradas digitais dos sensores nos pinos 4 e 5 do Arduino. Do mesmo modo é realizada a montagem do sensor magnético, sendo conectado na entrada 8 do Arduino.

Estes sensores são colocados na parte inferior do chassi, afim de detectar a linha a ser seguida e os imãs que são detectados nos pontos de paradas. A Figura 3.18 ilustra as conexões mencionadas.

Figura 3.18 – Sensores ópticos e sensor magnético



Fonte: Autoria própria, 2023

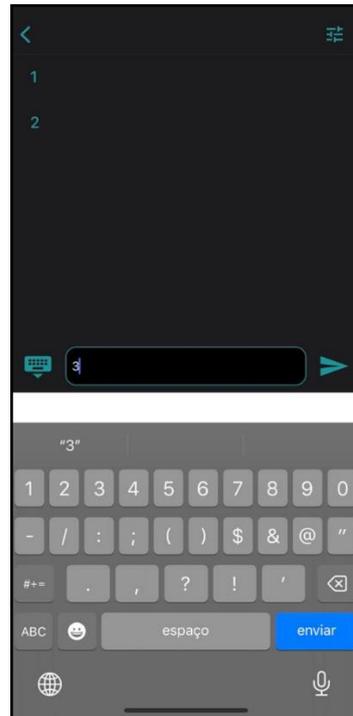
O módulo *buzzer* é responsável por emitir um aviso sonoro caso algo obstrua o caminho do AGV. Sua ligação faz-se através de três pinos, sendo: “Vcc”, “GND” e uma saída digital conectada ao pino 9 do Arduino.

O projeto conta com quatro botões do tipo *push botton*, sendo um deles para a função de iniciar trajeto e confirmar a retirada do pedido e os outros três que realizam a seleção da mesa desejada de forma física. Os botões são ligados nos pinos de entradas analógicas, que no projeto exercem a função de digitais, apenas porque há uma limitação de saídas na placa utilizada.

Para a conexão via *Bluetooth*, utiliza-se o módulo HC-05, sendo um módulo para a comunicação sem fio via *Bluetooth* com o *smartphone*. Ele possui sua alimentação de 3,3 V, para isso é utilizado o pino de alimentação do Arduino de 3,3 V e conectado a duas entradas do Arduino, sendo uma delas do tipo PWM.

O funcionamento do módulo HC-05 se dá através dos pinos RX e TX, onde RX é o receptor e TX o transmissor. A ligação é feita conectando o receptor do módulo ao transmissor do Arduino, e o receptor do Arduino ao transmissor do módulo HC-05. Para isso é utilizado o pino 2 do Arduino como receptor e o pino 3 como transmissor. Dessa forma, declara-se o transmissor como uma saída e o receptor como uma entrada.

Para a transmissão dos dados, conecta-se um *smartphone* ao módulo através do aplicativo “Bluetooth for Arduino”. Dessa forma, é enviado ao Arduino o número correspondente a mesa desejada do pedido. O valor recebido é salvo em uma variável tipo “char”, que condiz a um valor da tabela ASCII, logo é necessário realizar a conversão dessa variável para uma variável do tipo inteira. A conversão se dá subtraindo 48 do valor recebido, deixando assim o número digitado no aplicativo igual ao número a ser processado na programação e podendo ser comparado às condições do programa. A Figura 3.19 ilustra o *layout* da tela do *smartphone* para o envio da mesa que será entregue o pedido.

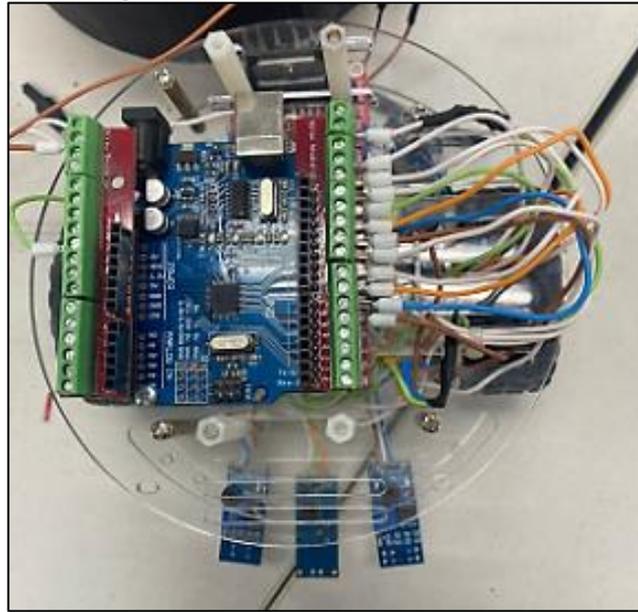
Figura 3.19 – Tela para comunicação *bluetooth*

Fonte: Autoria própria, 2023

O conjunto eletrônico é composto por duas fontes de alimentação. Sendo uma delas a bateria 9 V, responsável por realizar toda a alimentação do Arduino, juntamente com todos os periféricos ligados a ele, e uma fonte de 12 V para a alimentação dos motores. Por se tratar de fontes de alimentação diferentes, é necessário realizar uma ligação do GND da ponte H ao GND do Arduino para que estejam em um único plano de referência.

Para auxiliar a conexão dos periféricos ao Arduino, é utilizado uma placa de extensão de borne para Arduino Uno, assim os cabos são crimpados e ligados aos bornes, evitando que se soltem e gerem ruídos ou mau contato, além de manter o cabeamento organizado. A Figura 3.20 ilustra as conexões no Arduino.

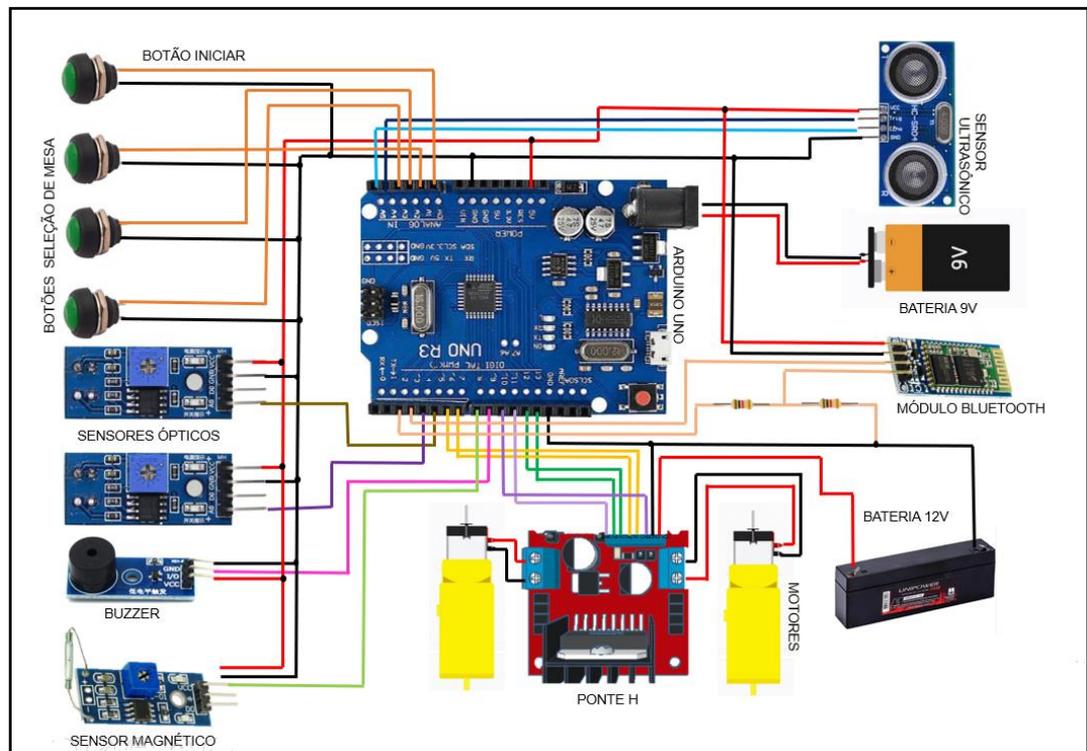
Figura 3.20 – Conexão eletrônica física



Fonte: Autoria própria, 2023

A Figura 3.21 ilustra o circuito geral com todos os componentes conectados, podendo ser observado cada entrada e cada saída ligada ao Arduino.

Figura 3.21 – Conexão eletrônica



Fonte: Autoria própria, 2023

3.3 Programação do microcontrolador

A programação do projeto é feita na placa de prototipagem Arduino Uno, utilizando o ambiente padrão: Arduino IDE. Para isso, baixa-se a versão mais recente do Arduino IDE, compatível com o computador que se utiliza para realizar a programação do projeto. A programação completa encontra-se no Apêndice F.

Primeiramente, faz-se o uso do cabo USB para a comunicação do microcontrolador com o computador. Após isso, no Arduino IDE na aba “ferramentas” e “portas”, seleciona-se a porta “COM” que está sendo usada para fazer a comunicação com o Arduino.

A seleção da placa utilizada faz-se através de “ferramentas”, onde se encontra a opção “placa”, que disponibiliza uma árvore de opções já predefinidas pelo *software*. Para o projeto, seleciona-se a placa “Arduino Uno”.

Para a utilização do sensor ultrassônico, baixa-se a biblioteca denominada como “New Ping”. Realiza-se o *download* e adiciona-se à pasta “*libraries*”, encontrada no disco local do computador, dentro da pasta do Arduino IDE. Para realizar a instalação da biblioteca, clica-se em “ferramentas” e “gerenciador de bibliotecas”. Após isso, na barra de pesquisa na janela, procura-se a opção “New Ping” e clica-se em “instalar”.

Assim o ambiente de programação já se encontra preparado para receber o código do projeto.

No código, primeiramente faz-se a inclusão da biblioteca a ser utilizada com o comando “*#include*”. Em seguida, encontram-se declaradas as variáveis que auxiliam a programação armazenando os dados no decorrer do *sketch*, sendo do tipo “*int*”, ou seja, guardam informações de números inteiros.

A função “*#define*” é responsável por definir nomes para as entradas e saídas do microcontrolador, denominando o nome e o número do pino respectivo após a função.

Após isso, faz-se a declaração dos pinos, denominando a função que cada pino exerce, podendo ser “*INPUT*”, sendo os pinos de sinais de entrada e “*OUTPUT*”, sendo sinais de saída, para isso é utilizada a função “*pinMode*”.

No decorrer do *sketch* são utilizados comandos de repetições e de instruções, como as funções “*while*”, “*if*” e “*else*”. A função “*while*” é um comando de repetição, fazendo com que um trecho do código esteja em *loop* até que alguma variável se torne verdadeira. Esse tipo de função encontra-se na lógica dos botões. Já as funções “*if*” e “*else*” são instruções de tomadas de decisão. Assim, caso a condição seja verdadeira realiza-se as ações dentro da função.

A programação conta com o programa principal, que é responsável pela leitura e processamento dos sensores e botões e decisões sobre quais os movimentos devem ser realizados. E também conta com quatro subprogramas responsáveis pelos tipos de movimentos que serão realizados, são eles: motores parados, movimento em frente, girar á direita e girar á esquerda.

Para a movimentação em frente, é necessário ligar os dois motores para frente, enquanto para que ele realize um movimento de giro é preciso que um lado se movimente mais rápido do que o outro, assim o lado que se deseja realizar o movimento se coloca uma velocidade menor em relação ao motor do lado oposto.

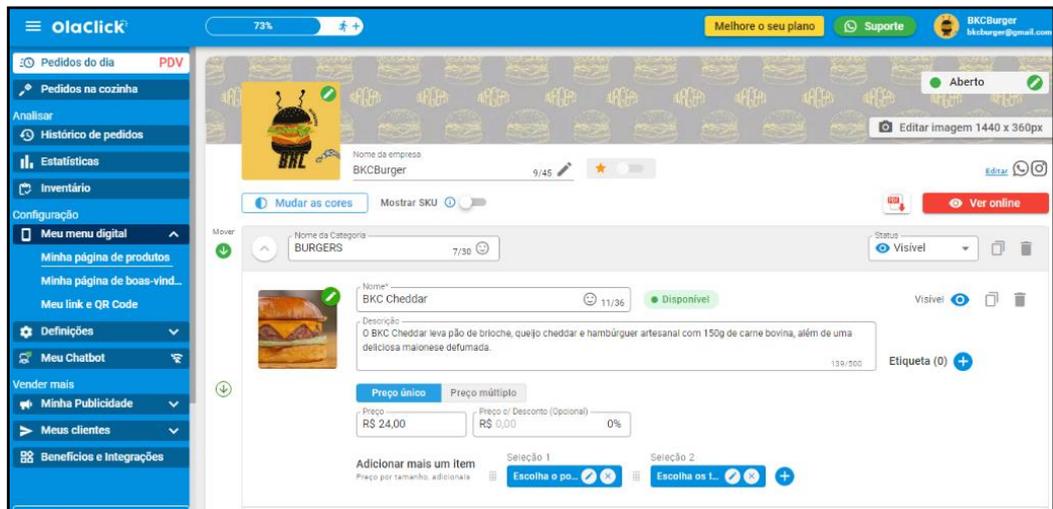
3.4 Criação do menu digital na plataforma Olaclick

Inicialmente cria-se uma conta de acesso a plataforma, cadastrando *e-mail* e número de celular. O cadastro faz-se concluído ao registrar o nome do restaurante, escolhido pelo grupo como BKCBurger.

Após a finalização do cadastro, clicando sobre o item “Seu logo aqui”, realiza-se a escolha da imagem que é utilizada como logo do restaurante. Ao lado do logo é inserido o nome da empresa. Faz-se a personalização do menu através da aba “Meu menu digital” e “Minha página de produtos”.

Para todos os itens, clica-se em “Adicionar produto”, permitindo acrescentar a descrição, foto, preço e informações adicionais, conforme mostra a Figura 3.22.

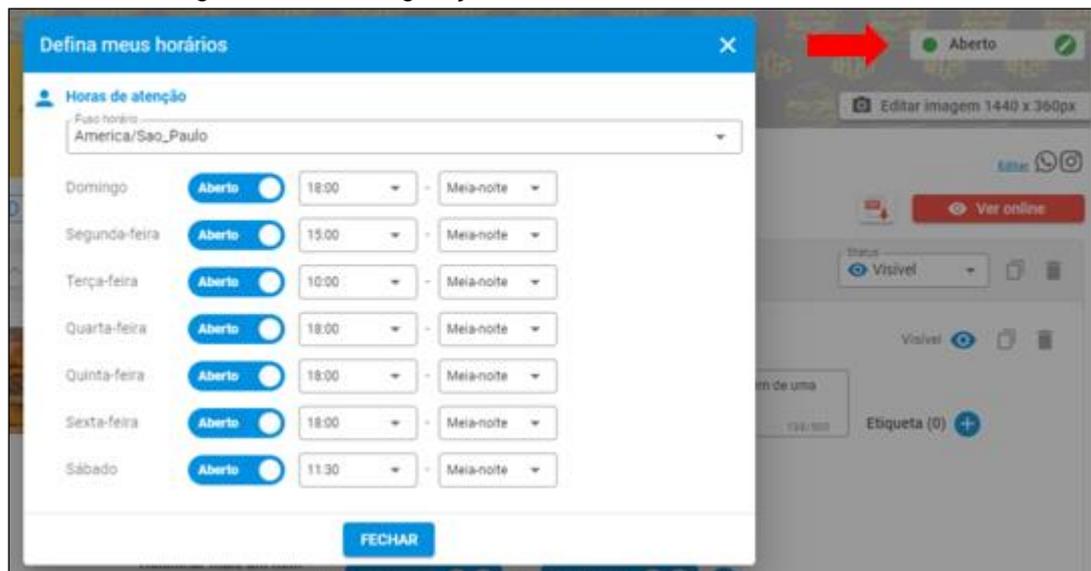
Figura 3.22 – Criação do menu editável Olaclick



Fonte: Autoria própria, 2023

Realiza-se a configuração do horário e dia de funcionamento do estabelecimento sobre o item nomeado como “Aberto”, encontrado no canto direito superior, conforme ilustra a Figura 3.23.

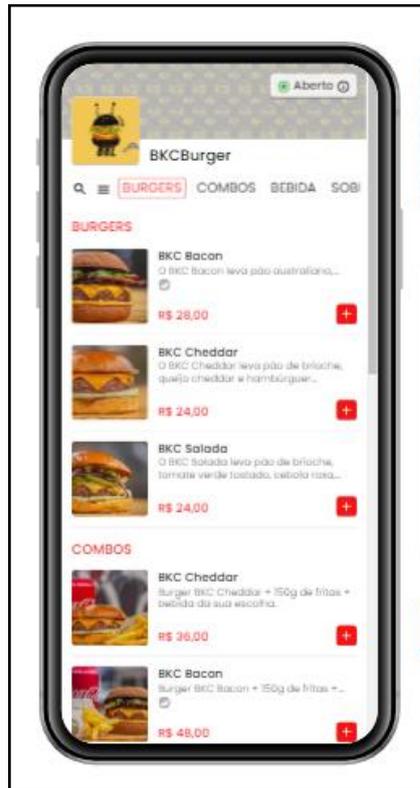
Figura 3.23 – Configuração dos horários de atendimento



Fonte: Autoria própria, 2023

Por fim, a Figura 3.24 ilustra como será visualizado o menu pelo *tablet* para a solicitação do pedido.

Figura 3.24 – Cardápio digital



Fonte: Autoria própria, 2023

3.5 Integração das partes e testes de funcionamento

Após a montagem das partes mecânica e eletrônica, os conjuntos são unidos acoplando o chassi à base principal através de espaçadores de M3 x 30 mm.

Com o conjunto finalizado, são feitos os testes de funcionamento. Para o auxílio dos testes, utiliza-se a função “monitor serial” encontrada na aba “ferramentas” do *software* Arduino IDE. Dessa forma, é possível acompanhar todos os sinais mandados e recebidos pelo microcontrolador.

Primeiramente, realiza-se de forma individual os testes dos sensores: magnético, ópticos, ultrassônico e *buzzer*. Através dos testes, consegue-se visualizar se os sinais estão sendo enviados corretamente para o Arduino. Após isso testa-se os botões, verificando se todos estão em funcionamento normal.

Em seguida, realiza-se o teste dos motores. Os motores são controlados pela Ponte H, sendo ela responsável pela alimentação e direcionar o sentido de giro dos

motores. Portanto, o teste é feito ligando os motores e verificando se ambos estão girando para a direção correta. Caso não estejam, é necessário inverter os cabos vindo dos motores. Essa ação é feita porque os motores são de corrente contínua e sua direção de giro é invertida através dessa mudança nos bornes.

Além do teste do sentido de giro dos motores, é necessário fazer a mudança da velocidade, respeitando a relação das grandezas: quanto menor velocidade, maior força.

Com o fim dos testes, carrega-se o programa do Arduino. Faz-se a ligação de todos os componentes e o ciclo é iniciado.

3.6 Obstáculos e soluções

Durante o desenvolvimento e construção do projeto, deparou-se com alguns obstáculos que foram solucionados através de conhecimentos nas pesquisas, experiência profissional das integrantes do grupo e consulta a professores.

Obstáculo 1: ao imprimir o conjunto de peças na impressora 3D, que possuem a finalidade de acomodar os alimentos, o acabamento e fixação não ficou como o desejado, não houve concordância com os raios e assim, dificultando o encaixe entre elas.

Solução: faz-se a ajustagem das peças para que elas se unam, utilizando lima chata e lixas d'água para melhor acabamento.

Obstáculo 2: ao imprimir a prateleira do conjunto mecânico, encontra-se dificuldade devido o material estar ressecado por não estar dentro da estufa, conforme indicação do fabricante.

Solução: realiza-se a troca do material.

Obstáculo 3: ao realizar o teste do funcionamento do sensor magnético, percebe-se que o contato estava quebrado, fazendo com que o sinal não alternasse quando o campo magnético era detectado através do imã.

Solução: compra-se uma nova ampola e faz-se a troca, soldando no módulo do sensor magnético.

Obstáculo 4: ao fazer as ligações dos fios nos motores, percebe-se que os fios são soltos com facilidade devido o tamanho da sua bitola.

Solução: faz-se a troca por fios com bitola maior, reforça-se a solda com uso de termo retrátil para isolação.

Obstáculo 5: percebe-se que ao realizar os testes de funcionamento dos componentes eletrônicos, ocorre com frequência a desconexão dos fios ao Arduino.

Solução: é acrescentado o *Shield* ao Arduino, adaptando os pinos para receberem fios crimpados em vez dos cabos jumper.

CONSIDERAÇÕES FINAIS

O projeto intitulado Entrega de pedidos nos restaurantes com uso de AGV tem como objetivo desenvolver um protótipo de um robô seguidor de linha que realiza a entrega de alimentos dentro dos restaurantes. O objetivo foi concluído com êxito.

O protótipo justifica-se pela redução do número de funcionários nos restaurantes, agilizando a entrega dos pedidos, melhorando a qualidade de trabalho e ergonomia do funcionário, proporciona também aos clientes uma experiência única e padronizada, evitando desconforto, além de trazer maior retorno financeiro aos proprietários.

Todo o capítulo de fundamentação teórica é de suma importância para a base do projeto. É apresentado os principais conceitos dos componentes utilizados, dando suporte para organizar e planejar o direcionamento de toda a construção, auxiliando nas escolhas dos componentes, dos *softwares* utilizados e traçando um caminho a ser seguido, detectando assim possíveis falhas que possam surgir.

Na construção, o funcionamento do AGV ocorre através do Arduino, responsável por realizar todo o controle de movimentação, monitorando sinais de entradas e saídas, afim de realizar a movimentação do modo desejado e seguir a linha de percurso proposto.

Como vantagem, destaca-se a aplicação de diversos conhecimentos vistos durante o curso, colocando teoria em prática, integrando conceitos e pesquisando novas tecnologias. Dessa forma, proporcionando novos conhecimentos e aplicando as habilidades aprendidas em situações reais.

Como desvantagem, o projeto não apresenta fácil acesso ao circuito eletrônico, aumentando assim a complexidade e o tempo de possíveis manutenções. Para uma melhoria futura, é necessário desenvolver uma arquitetura de construção que possibilite uma manutenção mais acessível.

REFERÊNCIAS

ALMEIDA, L. F. A. **Veículo autoguiado (AGV – Automated Guided Vehicle) – protótipo seguidor de linha.** 2016. 8 p. Trabalho de Conclusão de curso (Graduação) – Engenharia Elétrica, Centro Federal de Educação Tecnológica de Minas Gerais. Belo Horizonte, 2016.

AUTODESK. **Funcionamentos básicos do software Inventor.** 2022. Disponível em: <https://www.autodesk.com.br/education/home>. Acesso 26 de Setembro de 2022.

CAPELLI, A. **Automação Industrial: controle de movimento e processos contínuos.** 1. ed. São Paulo: Érica, 2006.

DELGADO, A. L. **Linguagem C++.** 2018. Disponível em: <https://www.inf.ufpr.br/ci208/NotasAula.pdf>. Acesso em: 26 de setembro de 2022.

DJP AUTOMAÇÃO. **Sensores industriais – descubra sua importância para a indústria moderna.** 2020. Disponível Em: <https://djpaotomacao.com/sensores-industriais-descubra-sua-importancia-para-a-industria-moderna/#:~:text=Sensores%20industriais%20s%C3%A3o%20dispositivos%20com,de%20detectar%20determinada%20grandeza%20f%C3%ADsica>. Acesso em: 10 de setembro de 2022.

FACULDADE DE TECNOLOGIA DE SÃO BERNARDO DO CAMPO “ADIB MOISÉS DIB”. **Manual de Normalização de Projeto de Trabalho de Graduação:** Material didático para utilização nos projetos de trabalho de graduação, dos cursos de Tecnologia em Automação Industrial e Tecnologia em Informação para Negócios. 5. Ed. Revisada e atualizada. São Bernardo do campo: Centro Paula Souza, 2017.

FERREIRA, C. A. **Sistema de monitoramento em motores de indução trifásico utilizando a plataforma Arduino.** 2019. Disponível em: https://riut.utfpr.edu.br/jspui/bitstream/1/11836/2/GP_COMIM_2019_2_03.pdf. Acesso em: 07 de outubro de 2022.

FIA BUSINESS SCHOOL. **Impressão 3D: O que é, como funciona e exemplos de aplicações.** 2019. Disponível em: <https://fia.com.br/blog/impressao>. Acesso em: 29 de outubro de 2022.

GOOMER, R. G. **Sistema de pedidos para restaurantes: conheça os 5 tipos de soluções.** 2019. Disponível em: <https://blog.goomer.com.br/sistemas-de-pedidos-para-restaurantes/> Acesso em: 30 de setembro de 2022.

G1. **Restaurante chinês usa robôs para cozinhar e servir clientes.** 2014. Disponível Em: <https://g1.globo.com/planeta-bizarro/noticia/2014/08/restaurante-chines-usa-robos-para-cozinhar-e-servir-clientes.html>. Acesso em: 10 de setembro de 2022

HONDA, F. **Motores de corrente contínua guia rápido para uma especificação precisa.** 2006. Disponível em:

http://www.marioloureiro.net/tecnica/electrif/Motores_CC_ind1.pdf. Acesso em: 07 de outubro de 2022.

INVIA ROBOTICS. **Autonomous Warehouse Robots: a Brief History**. 2019. Disponível Em: <https://inviarobotics.com/blog/autonomous-warehouse-robots-brief-history/?cn-reloaded=1>. Acesso em: 05 de novembro de 2022.

KIM C. W; TANCHOCO, J. M. A. **AGV dispatching based on workload balancing**. 1999.

LAMB, F. **Automação Industrial na prática**. AMGH; 1. Ed. São Paulo: Novatec Editora, 2015.

MATTEDE, H. **O que são sensores e quais as suas aplicações?** 2006. Disponível Em: <https://www.mundodaeletrica.com.br/o-que-sao-sensores-e-quais-as-suas-aplicacoes/>. Acesso em: 10 de setembro de 2022.

MAXCOMP. **Soluções para a sua impressão. 3D Cloner DH**. 2016. Disponível em: <https://maxcomp.com.br/index.php/modelos/dh/>. Acesso em: 29 de outubro de 2022.

MCROBERTS, M. **Arduino básico**. Tradução de Rafael Zanolli. São Paulo: Novatec Editora, 2011.

NOLETO, C. **C++: o que é, porque usar e quais as diferenças com C**. 2021. Disponível em: <https://blog.betrybe.com/linguagem-de-programacao/cpp/>. Acesso em: 26 de setembro de 2022.

OLIVEIRA, L. **OlaClick pretende dobrar número de usuários no 1º semestre de 2022**. 2022. Disponível em: <https://www.segs.com.br/info-ti/332438-olaclick-pretende-dobrar-numero-de-usuarios-no-1-semester-de-2022#:~:text=Como%20funciona%3F,para%20o%20WhatsApp%20do%20restaurant> e. Acesso em: 02 de outubro de 2022.

PORTELA, S. **4 dicas para escolha ideal do filamento para impressora 3D. CURA 3D: aprenda os primeiros passos desse fatiador 3D**. 2019. Disponível em: <https://3dlab.com.br/como-escolher-o-filamento-para-impressora-3d/>. Acesso em: 29 de outubro de 2022

PRODANOV, C. C.; FREITAS, E. C. **Metodologia do trabalho científico: Métodos e técnicas da pesquisa e do trabalho científico**. 2. ed. Rio Grande do Sul: Universidade Feevale, 2013.

RAZOR. **SketchUp: O que é e qual sua contribuição para arquitetura?** 2019. Disponível Em: <https://razor.com.br/blog/tecnologia/sketchup/>. Acesso em: 27 de outubro de 2022.

SANTOS, M. A. de A. **Sistema para controle de pedidos em restaurantes**. Orientador: MsC. Prof. Francisco Javier de Obaldia Diaz. 2014. 59. Certificado de Conclusão de Curso, Engenharia de Computação, Centro universitário de Brasília

(UniCEUB). 2014. Disponível em: <https://core.ac.uk/download/pdf/187130867.pdf>. Acesso em: 21 de agosto de 2022.

SEVERINO, A. J. **Metodologia do trabalho científico**. 23. ed. São Paulo: Cortez, 2013.

SILVEIRA, C. B. **Sensor ultrassônico: 10 aplicações para a indústria**. 2016. Disponível Em: <https://www.citisystems.com.br/sensor-ultrassonico/>. Acesso em: 10 de setembro de 2022.

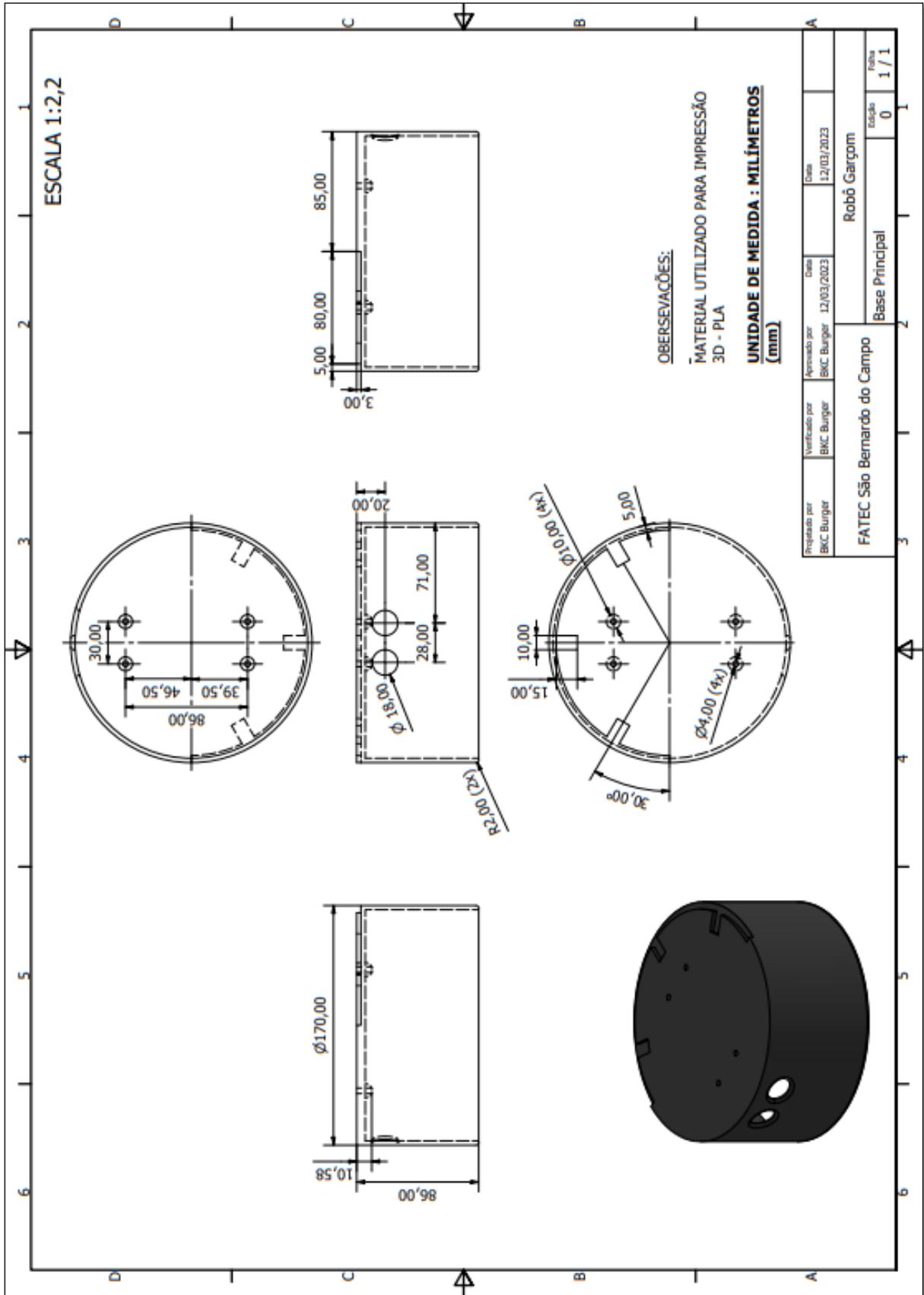
SINOVA. **Entenda como funciona o sistema de implantação do AGV**. 2022. Disponível Em: <https://www.sinova.com.br/2020/09/11/entenda-como-funciona-o-sistema-de-implantacao-do-agv/>. Acesso em: 10 de setembro de 2022.

SSI SHAEFER. **Veículos automaticamente guiados na intralogística**. 2015. Disponível Em: <https://www.ssi-schaefer.com/pt-br/produtos/transporte-interno-de-materiais/agvs>. Acesso em: 10 de setembro de 2022.

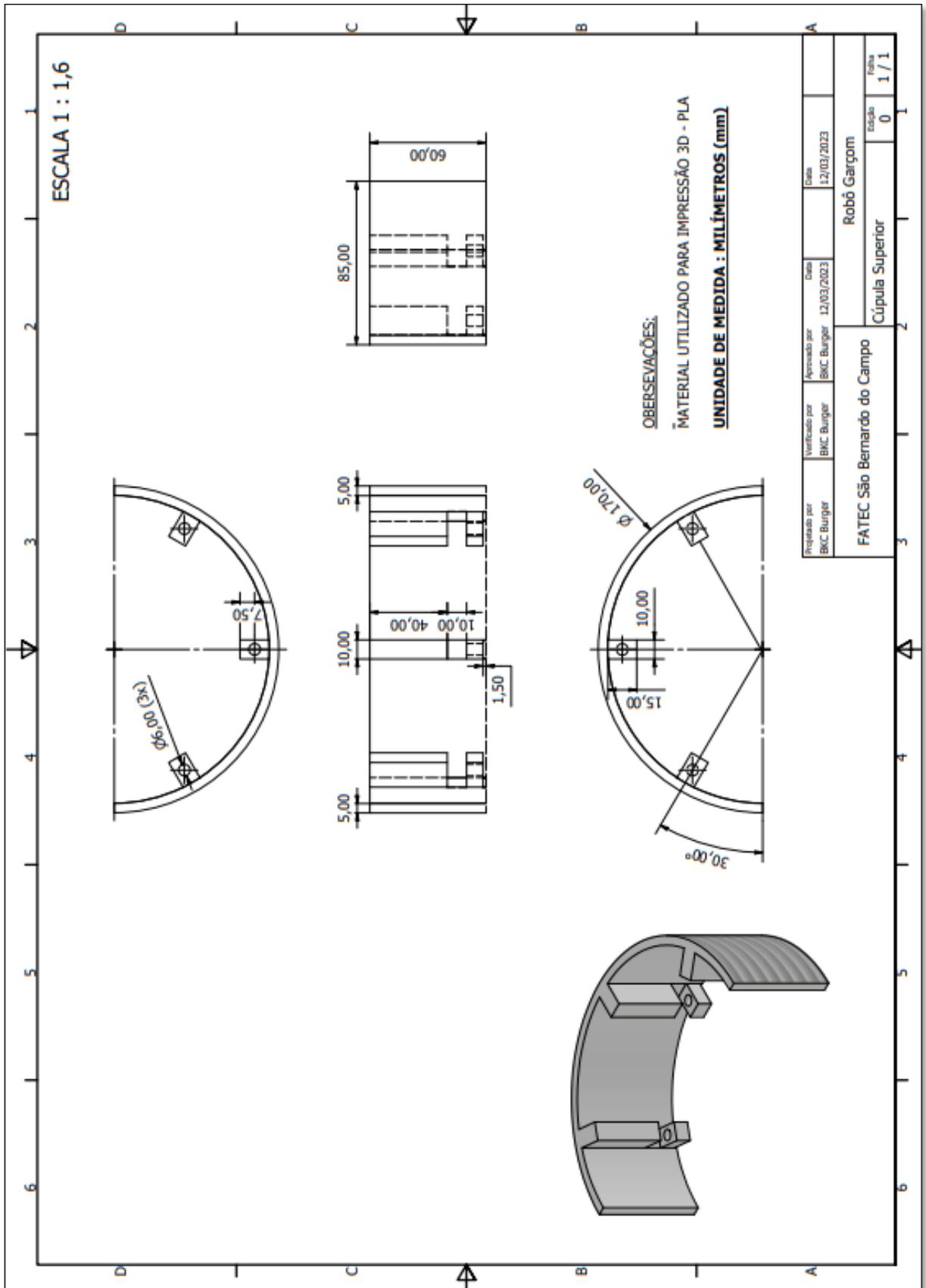
THOMAZINI, D.; ALBUQUERQUE, B. P. **Sensores industriais: fundamentos e aplicações**. 9. ed. São Paulo: Érica, 2020.

VIANNA, O. de A. **Metodologia do trabalho científico: um enfoque didático da produção científica**. São Paulo: EPU, 2001.

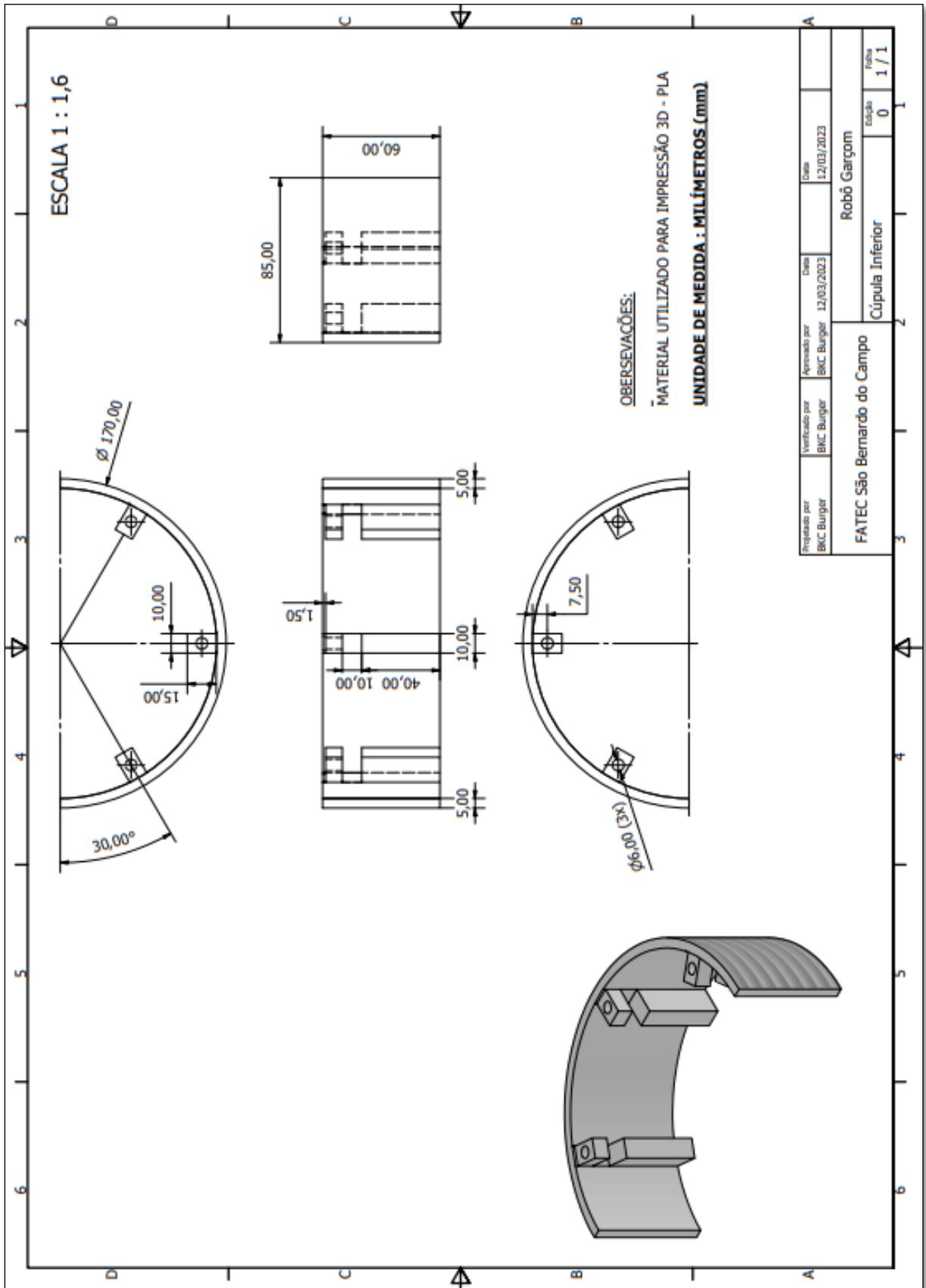
APÊNDICE A – DESENHO BASE PRINCIPAL



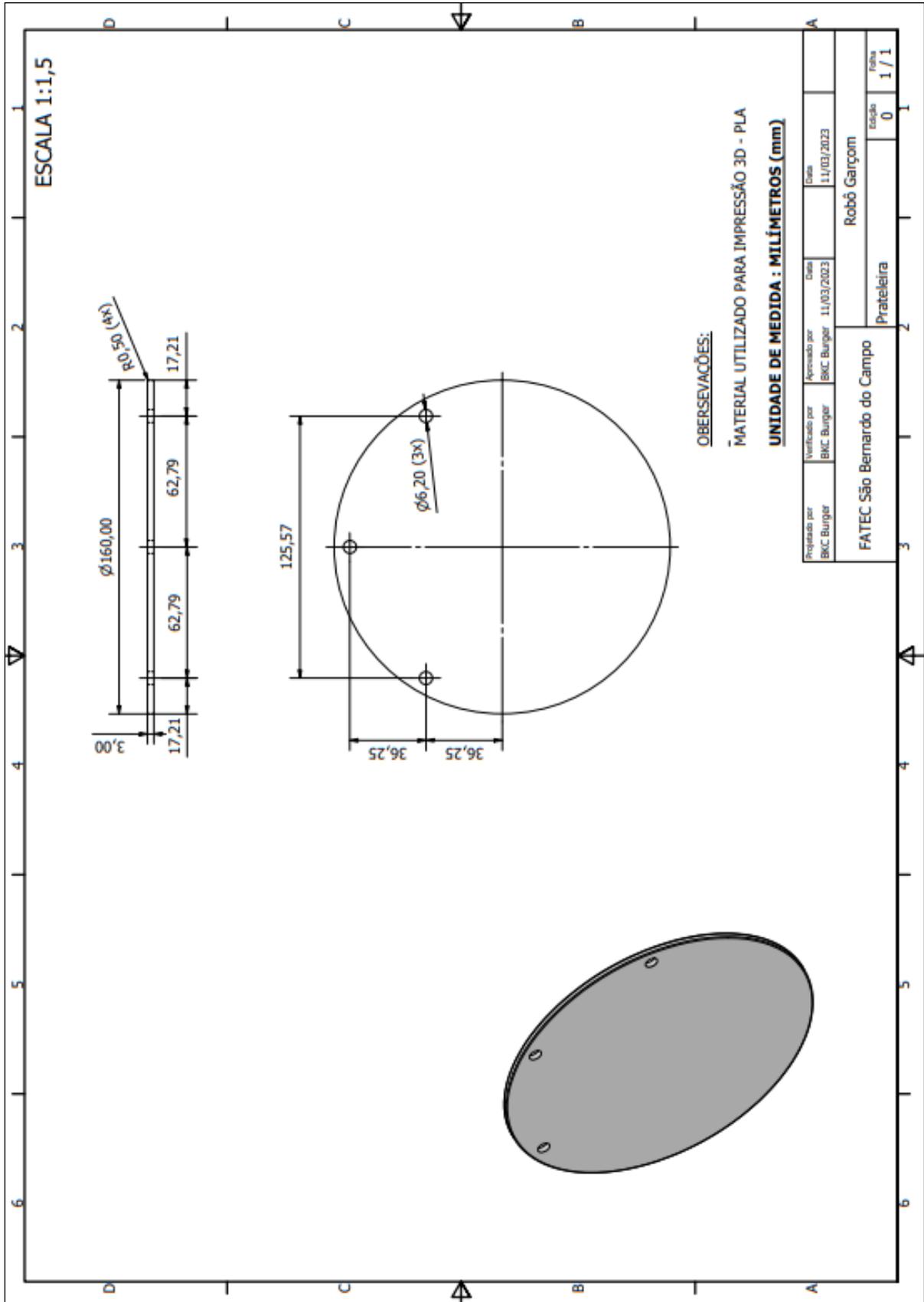
APÊNDICE B – DESENHO CÚPULA INFERIOR



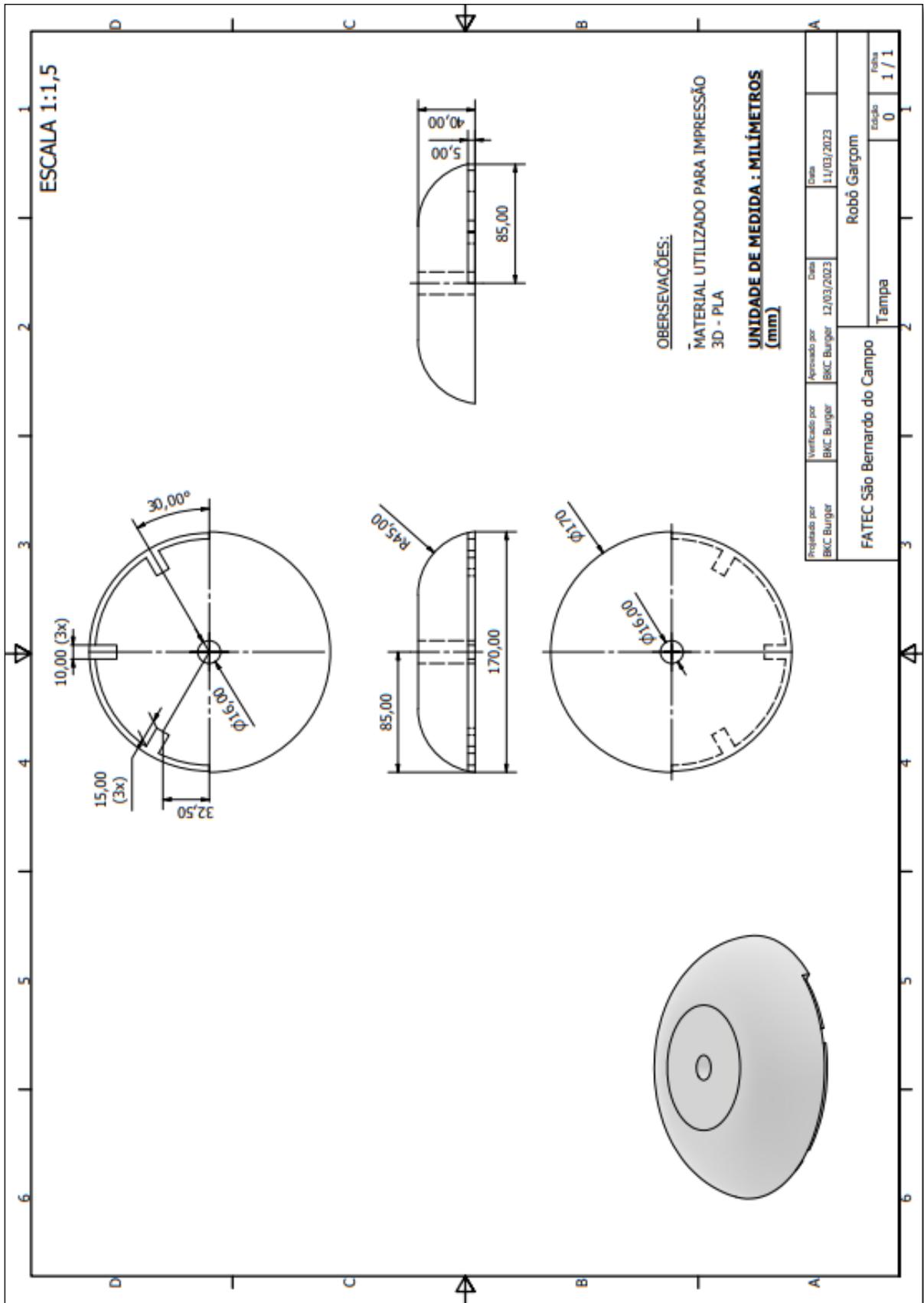
APÊNDICE C – DESENHO CÚPULA SUPERIOR



APÊNDICE D – DESENHO PRATELEIRA



APÊNDICE E – DESENHO TAMPA



APÊNDICE F – PROGRAMAÇÃO ARDUINO

```

#include <SoftwareSerial.h> // Biblioteca módulo bluetooth.
#include <NewPing.h>        // Biblioteca sensor ultrassônico.

//DECLARAÇÃO DAS VARIÁVEIS
int DIS = 0;                // Variável usada para comparar distância medida do sensor
ultrassônico.
int CONT = 0;              // Variável para contabilizar o numero de paradas.
int A = 0;                 // Variável que fará o robo andar somente quando soltar o
botão 1 para iniciar.
int E = 0;                 // Variável que fará a seleção da mesa 1 somente quando
soltar o botão.
int F = 0;                 // Variável que fará a seleção da mesa 2 somente quando
soltar o botão.
int G = 0;                 // Variável que fará a seleção da mesa 3 somente quando
soltar o botão.
int M = 0;                 // Variável que fará a contagem apenas no pulso positivo
do sensor magnético.
int B = 0;                 // Variável que fará o robo voltar para cozinha somente
quando soltar o botão 1 para confirmar.
int C = 0;                 // Variável de seleção para botão 1. Caso seja 0 - botão
como iniciar, caso seja 1 - botão como confirmar.
int V = 1;                 // Variável auxiliar tempo de saída do magneto
int PARADA1 = 0;          // Variável usada para loop de espera para iniciar ciclo.
int PARADA2 = 0;          // Variável usada para loop de espera para seleção da
mesa.
int PARADA3 = 0;          // Variável usada para loop de espera para confirmar
ciclo.
int MESA = 0;             // Variável auxiliar de seleção de mesa.
int estadobotini;         // Variável para salvar status do botão iniciar.
int estadobotmesa1;       // Variável para salvar status do botão mesa1.
int estadobotmesa2;       // Variável para salvar status do botão mesa2.
int estadobotmesa3;       // Variável para salvar status do botão mesa3.
int estadosensormag;     // Variável para salvar status do sensor magnético.
unsigned long now=0;      // Auxiliar do temporizador
unsigned long agr=0;      // Auxiliar do temporizador 2

// DEFINIÇÕES DE PINOS DO SENSOR SEGUIDOR DE LINHA
#define SensorDir 5 // Define pino 4 para o sensor seguidor de linha lado
direito.
#define SensorEsq 4 // Define pino 5 para o sensor seguidor de linha lado
esquerdo.

// DEFINIÇÕES DE PINOS DO SENSOR ULTRASSÔNICO
#define TRIGGER_PIN A4 // Define pino A4 para sensor ultrasônico.
#define ECHO_PIN A5 // Define pino A5 para sensor ultrasônico.
#define MAX_DISTANCE 200 // Define distância de detecção.

// DEFINIÇÃO DE PINO DO SENSOR MAGNÉTICO
#define SensorMag 8 // Define pino 8 para o sensor magnético.

// DEFINIÇÃO DE PINO DO BUZZER
#define Buzzer 9 // Define pino 9 para o buzzer (alarme).

```

```

// DEFINIÇÕES DE PINOS DOS MOTORES
#define dirFrente 12 // Define pino 13 para controle do motor direito para
sentido horário. (N4)
#define dirTras 13 // Define pino 12 para controle do motor direito para
sentido antihorário. (N3)
#define esqFrente 6 // Define pino 7 para controle do motor esquerdo para
sentido horário. (N1)
#define esqTras 7 // Define pino 6 para controle do motor esquerdo para
sentido antihorário. (N2)

// DEFINIÇÕES DE PINOS DOS BOTÕES
#define botaoIni A3 // Define pino A3 para botão iniciar.
#define mesa1 A0 // Define pino A0 para botão de seleção da mesa 1.
#define mesa2 A1 // Define pino A1 para botão de seleção da mesa 2.
#define mesa3 A2 // Define pino A2 para botão de seleção da mesa 3.

// DEFINIÇÕES DE PINOS CONTROLE DE VELOCIDADE PWM
#define velEsquerdo 10 //Define pino 10 para controle pwm de velocidade.
(ENB)
#define velDireito 11 //Define pino 11 para controle pwm de velocidade.
(ENA)

#define LINHA HIGH //Define linha como valor 1.

// DEFINIÇÃO BLUETOOTH
#define pinoRx 2 // Define pino 2 para receptor.
#define pinoTx 3 // Define pino 3 para transmissor.

SoftwareSerial bluetooth (pinoRx, pinoTx); // Definição módulo
bluetooth.

NewPing sonar(TRIGGER_PIN,ECHO_PIN,MAX_DISTANCE); // Definição sensor
ultrassônico.

// DECLARAÇÃO PARA CHAMADA DE FUNÇÕES
void girarDireita(); // Declara subprograma.
void girarEsquerda(); // Declara subprograma.
void motoresParados(); // Declara subprograma.
void andarFrente(); // Declara subprograma.

// DECLARAÇÃO DE VARIÁVEIS PARA CONTROLE DO SENSOR SEGUIDOR DE LINHA
bool leituraEsquerda; // Declara variável para salvar valor do sensor seguidor
de linha esquerdo.
bool leituraDireita; // Declara variável para salvar valor do sensor seguidor
de linha direito.
char DADOS=0; // Declara variável para salvar dados do bluetooth.

// DECLARAÇÕES DE ENTRADAS E SAÍDAS
void setup()

```

```

{
// DECLARAÇÃO DE PINOS COMO ENTRADA OU SAÍDA
pinMode (SensorDir,INPUT);      // Define SensorDir como entrada.
pinMode (SensorEsq, INPUT);     // Define SensorEsq como entrada.
pinMode (SensorMag, INPUT);     // Define SensorEsq como entrada.
pinMode (botaoIni, INPUT_PULLUP); // Define Botaoini como entrada PULL UP.
pinMode (mesa1, INPUT_PULLUP);  // Define Mesa1 como entrada PULL UP.
pinMode (mesa2, INPUT_PULLUP);  // Define Mesa1 como entrada PULL UP.
pinMode (mesa3, INPUT_PULLUP);  // Define Mesa1 como entrada PULL UP.
pinMode (Buzzer, OUTPUT);       // Define Buzzer como saída.
pinMode (velEsquerdo, OUTPUT);  // Define velEsquerdo como saída.
pinMode (velDireito, OUTPUT);   // Define velDireito como saída.
pinMode (dirFrente, OUTPUT);    // Define dirFrente como saída (motor do
lado direito para frente).
pinMode (dirTras, OUTPUT);      // Define dirTras como saída (motor do lado
direito para trás).
pinMode (esqFrente, OUTPUT);    // Define esqFrente como saída (motor do
lado esquerdo para frente).
pinMode (esqTras, OUTPUT);     // Define esqTras como saída (motor do lado
esquerdo para trás).
pinMode (pinoRx, INPUT);       // Define pinoRx como entrada.
pinMode (pinoTx, OUTPUT);      // Define pinoTx como saída.

// DESLIGA SAÍDAS
digitalWrite (dirFrente, LOW);  // Desliga motor direito.
digitalWrite (dirTras, LOW);    // Desliga motor direito.
digitalWrite (esqFrente, LOW);  // Desliga motor esquerdo.
digitalWrite (esqTras, LOW);    // Desliga motor esquerdo.

digitalWrite (Buzzer, 1);      // Desliga Buzzer (lógica invertida)

// DEFINIÇÃO DE VELOCIDADE MONITOR SERIAL
Serial.begin (9600);           // Define velocidade como 9600.
bluetooth.begin (9600);       // Define velocidade do serial bluetooth.
}

// INÍCIO DO PROGRAMA
void loop() {

    bool valE = digitalRead (SensorEsq);    // Guarda na variável ValE valor
do SensorEsq.
    bool valD = digitalRead (SensorDir);    // Guarda na variável valD valor
do SensorDir.

// ESPERA CONDICIONAL PARA A ESCOLHA DA MESA
while (PARADA2==0) // SELEÇÃO MESA 1
{
    Serial.println ("Seleciona mesa ");
    estadobotmesa1 = digitalRead (mesa1); // Guarda na variável
estadobotmesa1 valor do mesa1.
    estadobotmesa2 = digitalRead (mesa2); // Guarda na variável
estadobotmesa1 valor do mesa2.
    estadobotmesa3 = digitalRead (mesa3); // Guarda na variável
estadobotmesa1 valor do mesa3.
}
}

```

```

// LEITURA DO MÓDULO BLUETOOTH
  if (bluetooth.available())
  {
    DADOS=bluetooth.read(); // Guarda na variável dados o valor recebido
do bluetooth.
    Serial.println(DADOS); // Mostra o valor recebido pelo bluetooth.
    DADOS=DADOS-48;        // Converte o valor recebido.
    delay(200);
  }

// SELEÇÃO DA MESA 1
// SUBROTINA PARA SINAL DO BOTÃO COM BORDA DE SUBIDA
  if (estadobotmesa1==0 && E==0)
  {
    E=1;
  }
  if ((estadobotmesa1==1 && E==1) || DADOS==1)
  {
    Serial.println ("Mesa 1 selecionada");
    MESA=1;
    E=0;
    PARADA2=1;
  }

// SELEÇÃO DA MESA 2
// SUBROTINA PARA SINAL DO BOTÃO COM BORDA DE SUBIDA
  if (estadobotmesa2==0 && F==0)
  {
    F=1;
  }
  if ((estadobotmesa2==1 && F==1) || DADOS==2)
  {
    Serial.println ("Mesa 2 selecionada");
    MESA=2;
    F=0;
    PARADA2=1;
  }

// SELEÇÃO DA MESA 3
// SUBROTINA PARA SINAL DO BOTÃO COM BORDA DE SUBIDA
  if (estadobotmesa3==0 && G==0)
  {
    G=1;
  }
  if ((estadobotmesa3==1 && G==1) || DADOS==3)
  {
    Serial.println ("Mesa 3 selecionada");
    MESA=3;
    G=0;
    PARADA2=1;
  }
}

```

```

// ESPERA CONDICIONAL PARA O BOTÃO INICIAL
while (PARADA1==0)
{
  estadobotini = digitalRead (botaoIni);
  Serial.println ("Seleciona botão iniciar ");
  if (estadobotini == 0 && A==0)
  {
    A=1;
  }
  if (estadobotini == 1 && A==1)
  {
    C=1;
    Serial.println ("Botão iniciar acionado");
    A=0;
    PARADA1=1;
  }
}

// MANTER ROBÔ NA FAIXA
if (vale == LINHA && valD == LINHA)
{
  motoresParados();
}
else if (valD == LINHA)
{
  girarDireita();
}
else if (vale == LINHA)
{
  girarEsquerda();
}
else
{
  andarFrente();
}

// SENSOR MAGNÉTICO
estadosensormag = digitalRead (SensorMag);
if (estadosensormag==1 && M==0)
{
  M=1;
}
if (estadosensormag==0 && M==1 && V==1)
{
  CONT=CONT+1;
  Serial.print ("Contagem do sensor magnético = ");
  Serial.println (CONT);
  M=0;
}

// CHEGOU NA MESA DESEJADA

```

```

if (MESA == CONT && PARADA3==0)
{
  motoresParados();
  V = 0;
  digitalWrite (Buzzer, 0);
  agr=millis();
  while(millis()<agr+500){}
  digitalWrite (Buzzer, 1);
  agr=millis();
  while(millis()<agr+300){}
  digitalWrite (Buzzer, 0);
  agr=millis();
  while(millis()<agr+500){}
  digitalWrite (Buzzer, 1);

// ESPERA CONDICIONAL PARA BOTÃO CONFIRMAR
while (PARADA3==0)
{
  Serial.println ("Espera botão para confirmar");
  estadobotini = digitalRead (botaoIni);
  if (estadobotini == 0 && B==0 && C==1)
  {
    B=1;
  }
  if (estadobotini == 1 && B==1 && C==1)
  {
    Serial.println ("Pedido confirmado");
    now=millis();
    PARADA3=1;
    B=0;
    C=0;
    while(millis()<now+1000){}
    V=1;
  }
}
}
if (PARADA3==1)
{
  //andarFrente();
  if (valeE == LINHA && valD == LINHA)
  {
    motoresParados();
  }
  else if (valD == LINHA)
  {
    girarDireita();
  }
  else if (valeE == LINHA)
  {
    girarEsquerda();
  }
  else
  {
    andarFrente();
  }
}

```

```

    }
}

// RETORNOU À COZINHA
if (CONT==4)
{
  motoresParados();
  Serial.println("Retornou a cozinha");
  CONT=0;
  MESA=0;
  PARADA1=0;
  PARADA2=0;
  PARADA3=0;
  DADOS=0;
}

// PASSAGEM OBSTRUIDA
Serial.print ("Ping: ");
Serial.print (sonar.ping_cm());
Serial.println ("cm");
int DIS = sonar.ping_cm(); // Salva valor lido na variável DIS.

if (DIS <= 4 && DIS!=0)
{
  motoresParados();
  digitalWrite (Buzzer, 0);
  delay(2000);
  digitalWrite (Buzzer, 1);
  delay(500);
}
}

void motoresParados()
{
  digitalWrite (dirFrente, LOW);
  digitalWrite (dirTras, LOW);
  digitalWrite (esqFrente, LOW);
  digitalWrite (esqTras, LOW);
}

void andarFrente()
{
  digitalWrite (dirFrente, HIGH);
  digitalWrite (dirTras, LOW);
  analogWrite (velDireito,60);
  digitalWrite (esqFrente, HIGH);
  digitalWrite (esqTras, LOW);
  analogWrite (velEsquerdo,70);
}

```

```
void girarDireita()  
{  
    digitalWrite (dirFrente, LOW);  
    digitalWrite (dirTras, HIGH);  
    analogWrite (velDireito, 60);  
    digitalWrite (esqFrente, HIGH);  
    digitalWrite (esqTras, LOW);  
    analogWrite (velEsquerdo,70);  
}
```

```
void girarEsquerda()  
{  
    digitalWrite (dirFrente, HIGH);  
    digitalWrite (dirTras, LOW);  
    analogWrite (velDireito, 60);  
    digitalWrite (esqFrente, LOW);  
    digitalWrite (esqTras, HIGH);  
    analogWrite (velEsquerdo,70);  
}
```