

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE MATO GROSSO DO SUL**

JOÃO VITOR ALVES GONÇALVES

**DISPOSITIVO AUTÔNOMO PARA MAPEAMENTO E SOLUÇÃO DE
LABIRINTOS UTILIZANDO ESP8266 E ALGORITMO DE BUSCA EM
LARGURA**

**JARDIM - MS
2023**

JOÃO VITOR ALVES GONÇALVES

**DISPOSITIVO AUTÔNOMO PARA MAPEAMENTO E SOLUÇÃO DE
LABIRINTOS UTILIZANDO ESP8266 E ALGORITMO DE BUSCA EM
LARGURA**

Trabalho de conclusão de curso apresentado
como requisito parcial à obtenção do título
de Técnico em Informática do Instituto
Federal de Mato Grosso do Sul.

Orientador: Me. Marcelo Christiano da
França Júnior

**JARDIM - MS
2023**



MINISTÉRIO DA EDUCAÇÃO
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso do Sul



ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO (TCC)

Aos 13 dias do mês de dezembro de 2023, no anfiteatro do Instituto Federal de Educação, Ciências e Tecnologia do Mato Grosso do Sul - IFMS, Câmpus Jardim, realizou-se a sessão pública de defesa de TCC do Curso técnico Integrado em Informática, do acadêmico João Vitor Alves Gonçalves sob orientação do professor Marcelo Christiano da França Júnior, intitulada Dispositivo Autônomo para Mapeamento e Solução de Labirintos Utilizando ESP8266 e Algoritmo de Busca em Largura.

Compuseram a Banca Examinadora os professores:

Orientador: Marcelo Christiano da França Júnior

Membro 2: Ewerton da Silva Schroeder

Membro 3: Patrik Olá Bressan

Após a exposição oral, o(s) (a)(s) candidato (a)(s) foi(ram) arguido(a)(s) pelos membros da banca, os quais reuniram-se reservadamente, e decidiram, Aprovação com a média final 9,0. Para constar, redigi a presente ata, que aprovada por todos os presentes, vai ser assinada por mim, Orientador do TCC, e pelos demais membros da banca.

Obs: _____

Professor Orientador: Nota: 9,0 Assinatura: Marcelo /

Professor Membro 2: Nota: 9,0 Assinatura: Ewerton Schroeder

Professor Membro 3: Nota: 9,0 Assinatura: Patrik Olá Bressan

Agradecimentos

Agradeço aos meus pais Teodison Gonçalves e Susana Aparecida Dos Santos Alves e avó Ivone Dos Santos Alves, por tudo que simboliza em minha vida, por todo apoio que sempre me proporcionaram, e por nunca deixarem desistir.

Ao meu orientador, Marcelo Christiano da França Júnior, Agradeço por toda paciência durante minha orientação e pelos grandes ensinamentos passados durante esta jornada e também pela confiança em mim de realizar este trabalho. Além de meu orientador ter sido uma grande inspiração para mim dentro do campus tanto quanto fora, sendo uma amizade que posso contar.

Agradeço também ao professor Ewerton Da Silva Schroeder que sempre deu boas ideias para realização do trabalho e se fez presente sempre que necessário, agradeço também pela ajuda fornecida na criação de um dos principais componentes do trabalho sem a mesma o trabalho não poderia ser realizado. Além do mais, expresso minha gratidão pelos sábios conselhos passados a mim durante este trabalho.

Aos meus parceiros André, Caio e Renan por serem grandes amigos, que sempre buscaram somar durante a realização do meu trabalho, foram amigos excelentes que me ajudaram quando tive alguma dificuldade, vale destacar que os mesmo acompanharam o trabalho desde o início por também serem orientados pelo professor Marcelo Christiano da França Júnior.

Ao IFMS campus Jardim por oferecer acesso a um ensino de qualidade em uma região de interior onde a educação não detém o seu devido valor.

Ao IF MAKER campus jardim, pelo fornecimento de materiais para a elaboração do protótipo, também expresso minha gratidão pelo espaço cedido para o desenvolvimento e testes do meu projeto.

*Código nosso que está em C
Santificado seja vós, console
Venha nos o vosso array[10] E seja feita, sim, {vossa chave}
Assim no if{} } Como no Else{} }
O for(nosso;de cada dia;nos daí hoje++)
Debugai as nossas sentenças
Assim como nós colocamos o ponto e vírgula esquecido;
E não nos Deixeis errar Indentação
Mas livrai-nos das funções recursivas
A main ().*

- *Oração Do Programador*

RESUMO

O trabalho de realizar o mapeamento de um labirinto e produzir o melhor caminho para o mesmo é um dever árduo para os seres humanos, sendo capaz de levar horas ou dias. Desta maneira, a utilização da robótica é essencial, tendo em vista o objetivo de automatizar tarefas complexas e repetitivas. O objetivo geral deste trabalho é produzir um robô autônomo que seja apto a mapear e solucionar um labirinto utilizando algoritmo de busca em largura, além de microcontrolador ESP8266, motores de corrente contínua e sensores infravermelhos.

Palavras-chave: Robótica autônoma. Labirinto. ESP8266. Algoritmo de busca em largura .

ABSTRACT

The task of mapping a maze and generating the best path through it is a challenging duty for humans, capable of taking hours or even days. Therefore, the use of robotics is essential, given the goal of automating complex and repetitive tasks. The overall objective of this work is to create an autonomous robot capable of mapping and solving a maze using a breadth-first search algorithm, in addition to an ESP8266 microcontroller, direct current motors, and infrared sensors.

Keywords: Autonomous robotics. Labyrinth. ESP8266. Breadth-first search algorithm.

LISTA DE ILUSTRAÇÕES

Figura 1: ESP-12E.....	4
Figura 2 – Sinal Analógico.....	5
Figura 3 - Sinal digital.....	6
Figura 4: Formula duty cycle.....	6
Figura 5: Duty cycle com porcentagem de 30%.....	7
Figura 6: ESP8266 12-E Chip Pinagem.....	7
Figura 6: NodeMCU.....	8
Figura 7: pinagem NodeMCU.....	9
Figura 8: IDE Arduino.....	10
Figura 9: Ferramentas IDE Arduino.....	11
Figura 10: Preferências IDE Arduino.....	12
Figura 11: Sensor HW-201.....	13
Figura 12: Ligação sensor HW-201 com Arduino.....	14
Figura 13: Eixos de medição do acelerômetro e giroscópio do MPU-6050.....	15
Figura 14: MPU-6050.....	15
Figura 15: Componentes motores elétricos.....	16
Figura 16: Motor de corrente alternada.....	17
Figura 17: Motor de corrente contínua.....	17
Figura 18: Algoritmo de busca em largura.....	18
Figura 19: Motor shield.....	21
Figura 20: Teste sensor com NodeMCU.....	21
Figura 21: Circuito Teste sensor com NodeMCU.....	22
Figura 23: Circuito teste de ponte H com moteres.....	23
Figura 24: Teste de bancada com motores.....	23
Figura 25: Código teste de bancada com motores.....	24
Figura 26: Estrutura.....	25
Figura 27:Labirinto moldavel.....	26
Figura 28 :Erro de medida protótipo.....	27
Figura 29:Segundo prototipo.....	28
Figura 30:Circuito segundo prototipo.....	28
Figura 31: Terceiro protótipo.....	30
Figura 32:Circuito Terceiro Protótipo.....	30
Figura 34:Caracteristica labirinto.....	32
Figura 35:Busca em largura aplicado.....	33
Figura 36: Quarto prototipo.....	34
Figura 38: Circuito quarto prototipo.....	35
Figura 39: Cenário de validação.....	36

Figura 40: Matrizes resultantes validação.....	36
Figura 41: Cenário.....	37
Figura 42: Matrizes resultantes.....	37
Figura 43: Cenário e matriz referente ao mapeamento.....	38
Figura 44: Caminho gerado.....	38

LISTA DE ABREVIASÕES

1. MHz: Milhão de Hertz
2. V : *Volts.*
3. GPIO : *General Purpose Input/Output.*
4. DC: Corrente continua.
5. PWM : *Pulse Width Modulation.*
6. IoT: *Internet of Things*
7. IDE : Ambiente de desenvolvimento integrado
8. GND: Filtro de densidade neutra
9. RST : Resetar
10. 3D : Três dimensões
11. CM: Centímetro
12. RPM : Rotação por minuto
13. I2c: Circuito Inter-integrado
14. IMU: Unidade de medida inercial
15. MPU: Unidade de Processamento de Movimento
16. LED: Diodo Emissor de Luz
17. SCL: Linha de Relógio Serial
18. SDA: Linha de Dados Serial
19. g: Força da gravidade
20. XDA: Linha Externa de Dados Serial
21. XCL: Linha Externa de Relógio Serial

SUMÁRIO

1. INTRODUÇÃO-----	1
1.1. JUSTIFICATIVA-----	2
1.2. OBJETIVOS-----	2
1.2.1. Objetivo Geral-----	2
1.2.2. Objetivos Específicos-----	2
1.3. ESTRUTURA DO TRABALHO-----	3
2. REVISÃO DA LITERATURA-----	4
2.1. INTRODUÇÃO-----	4
2.2. MICROCONTROLADORES-----	4
2.2.1. ESP8266-----	4
2.2.1.1. NodeMCU-----	8
2.2.1.2. PROGRAMAÇÃO-----	9
2.2.1.2.1. Software de codificação-----	10
2.3. SENSORES-----	12
2.3.1. Sensor reflexivo Infravermelho HW-201-----	13
2.3.1.1. Funcionamento-----	13
2.3.2. Sensor MPU-6050-----	14
2.4. ATUADORES-----	16
2.4.1. Motores elétricos-----	16
2.5. ALGORITMO DE BUSCA EM LARGURA-----	18
3. Metodologia-----	20
3.1. Componentes-----	20
3.1.1. Estrutura-----	24
3.1.2. Labirinto-----	25
3.2. Desenvolvimento-----	27
4. Resultados-----	36
4.1. Documentação-----	38
5. CONCLUSÃO-----	40
5.1. Trabalhos futuros-----	40
REFERÊNCIAS BIBLIOGRÁFICAS-----	41
Apêndice A - Código terceiro protótipo-----	44
Apêndice B - Código para página web-----	64
Apêndice C - Código Final-----	67

1. INTRODUÇÃO

Os labirintos fazem parte da história desde muito cedo, o mais conhecido labirinto é o labirinto de Creta que, segundo a mitologia grega, foi construído por Dédalo a pedido do Rei Minos para aprisionar o Minotauro. [1] Personagem mitológico com corpo humano e cabeça de touro. A construção se deu devido a um presente que Poseidon deu ao Rei, um belíssimo touro branco para que fosse sacrificado em sua homenagem, mas a beleza do touro fascinou o rei, que sacrificou outro touro no lugar do belo animal que Poseidon havia lhe dado. Em revolta a Minos que não sacrificou o animal correto, Poseidon fez com que Pasífae, esposa de Minos, se apaixonasse pelo touro branco e dessa paixão surge o Minotauro, após isso o labirinto de Creta foi construído. [2]

Entretanto, não há apenas labirintos como o de Creta, existem cerca de duas variações de labirintos, os chamados Meandros, cujos caminhos são únicos e geralmente usados para meditação; e os que desafiam o viajante a encontrar a saída, chamados Maze, o interesse em desenvolver algoritmos que pudessem descrever a solução para se encontrar a saída de qualquer labirinto surgiram com o desafio que esse segundo tipo de labirinto oferece aqueles dispostos a tentar resolver este, A partir do século XX, com o avanço da robótica principalmente autônoma este desafio se tornou mais relevante.[1]

A palavra robô foi popularizada pelo dramaturgo tcheco Karel Capek em 1921, com sua peça Robôs universais de Rossum ,a palavra "robô" resulta da combinação das palavras tchecas *rabota*, que significa "trabalho obrigatório", e *robotnik*, que significa "servo". [1] Entretanto a robótica não fica apenas em trabalho obrigatório, como dito anteriormente com o passar do tempo a robótica evolui para autônoma, um robô autônomo atua com base nas suas próprias decisões e não é controlado por um ser humano, porém, ainda existem robôs que não são autônomos, sendo controlados de alguma forma por seres humanos para realizar determinada atividade.[3] O robô autônomo tem a capacidade de identificar o ambiente em sua volta utilizando sensores, com estes sensores é possível ouvir, tocar e enxergar assim obtendo informações, portanto com estas informações obtidas o dispositivo executa determinada função que o auxilia na resolução da tarefa sem haver interferência humana.[3]

1.1. JUSTIFICATIVA

O ato de resolver um labirinto, para muitas pessoas, pode parecer algo muito complexo e trabalhoso, pois requer muito tempo e esforço. Desse modo, com auxílio da tecnologia e de um ESP8266, é possível desenvolver soluções para este problema dos seres humanos, resolvendo com precisão e rapidez um labirinto, possibilitando a chegada de um ponto A para um ponto B sem possuir conhecimento do caminho, através de algoritmos que são capazes de realizar tais atividades, como algoritmos de busca em largura, busca em profundidade entre outros, neste trabalho de conclusão de curso será usado o algoritmo de busca em largura.

1.2. OBJETIVOS

1.2.1. Objetivo Geral

Este trabalho de conclusão de curso tem como objetivo geral desenvolver um protótipo de um dispositivo autônomo para mapeamento e solução de labirintos utilizando ESP8266 e algoritmos de busca em largura, a fim de encontrar uma solução para qualquer labirinto, mais prática e mais rápida, buscando mostrar que a tecnologia ESP8266 é capaz de desenvolver um trabalho mesmo este possuindo uma realização complexa.

1.2.2. Objetivos Específicos

Para realização do objetivo geral, é necessário alguns objetivos específicos, os quais estão detalhados a seguir:

- Entender sobre o funcionamento do ESP8266 e seus componentes;
- Revisar conhecimento sobre o funcionamento de motores de corrente contínua;
- Compreender sobre o funcionamento do motor shield para NodeMCU ESP8266 que possui Ponte-H L293D incluso na placa;
- Entender sobre o funcionamento do sensor de Obstáculo Reflexivo Infravermelho HW-201;
- Criar um protótipo virtual para validação do dispositivo;
- Imprimir as peças necessárias para montagem utilizando impressão 3D;

- Desenvolver um protótipo ;
- Criar um algoritmo de busca em largura para mapeamento de labirintos.
- Analisar os resultados obtidos;

1.3. ESTRUTURA DO TRABALHO

Esse trabalho está estruturado em 5 capítulos:

O capítulo 1 apresenta a introdução do trabalho proposto, logo após a justificativa que mostra o porquê do trabalho ser feito, e seguido dos objetivos.

O capítulo 2, aborda a revisão de literatura discutida no trabalho, é feita uma descrição dos modelos de microcontroladores, posteriormente é apresentado o funcionamento dos sensores e atuadores usados neste trabalho, por fim abordamos sobre algoritmos de busca em largura.

O capítulo 3 relata sobre metodologia aplicada, onde é detalhado como foi construído o protótipo, quais os métodos e materiais utilizados.

O capítulo 4 descreve os resultados obtidos através do protótipo criado baseados nos métodos e materiais utilizados e nos testes realizados com o protótipo.

No capítulo 5, estão presentes as considerações finais do trabalho desenvolvido assim como as propostas de continuidade do projeto.

2. REVISÃO DA LITERATURA

2.1. INTRODUÇÃO

Para iniciar o estudo sobre o tema proposto, será necessário revisarmos os conhecimentos relevantes sobre diversos temas, a fim de conceber um protótipo capaz de resolver labirintos. Nesta revisão de literatura abordaremos os principais tópicos relacionados.

Iniciaremos falando dos microcontroladores e suas características, posteriormente passaremos a falar sobre os principais sensores para obtenção de dados, continuaremos descrevendo sobre motores identificando os tipos e características de cada um e por fim abordaremos os principais conceitos abordados no estudo de algoritmos de busca em largura.

2.2. MICROCONTROLADORES

2.2.1. ESP8266

O microcontrolador ESP8266 produzido pela fabricante chinesa Espressif, é um microcontrolador de 32 bits que inclui um núcleo microprocessado Tensilica L106, que funciona na frequência-padrão de 80 MHz, podendo chegar a 160 MHz, este componente possui tensão de funcionamento de 3.3V. É possível encontrar diversos módulos deste microcontrolador ESP-01, ESP-12E, ESP-07, ESP32, entre outros [4]. Neste trabalho será utilizado o módulo ESP-12E, este microcontrolador é composto por um sistema de comunicação *WiFi*, sendo este o seu grande diferencial no mercado. Na figura 1 temos a exibição do modelo ESP-12E.

Figura 1: ESP-12E



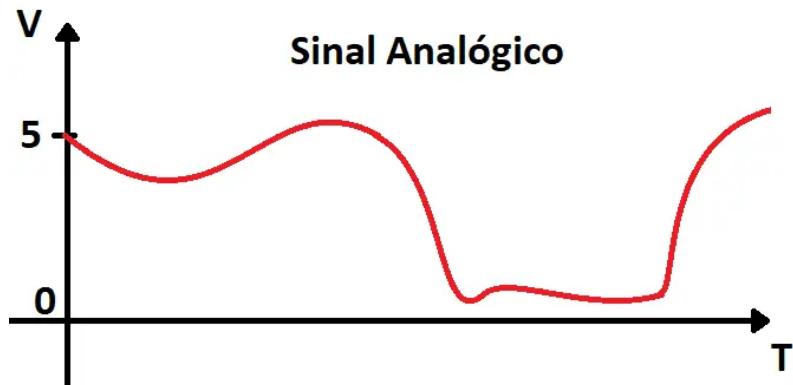
Fonte: makerhero - ESP8266

Em relação às interfaces de entrada e saída, o ESP8266 apresenta 17 interfaces GPIO

(General Purpose Input/Output) que são entradas e saídas que podem ser usadas de diversos modos como por exemplo podemos trabalhar com sinais do tipo analógico , digital e PWM (Pulse Width Modulation).[5]

Para podermos entender sobre as entradas e saídas, precisamos revisar certos conceitos, dentre eles sinais analógicos e digitais. A informação analógica corresponde a uma onda eletromagnética, que pode assumir infinitos valores no tempo, portanto é um tipo de sinal contínuo, como representação numérica deste sinal podemos ter todos os valores intermediários de uma determinada faixa de números, abaixo segue uma imagem de exemplo de variação de um sinal analógico. [6]

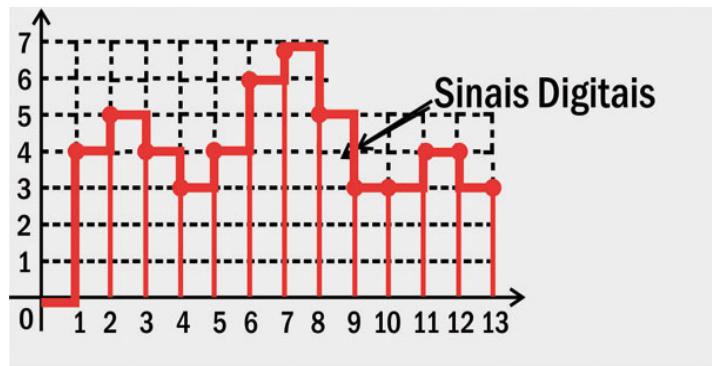
Figura 2 – Sinal Analógico



Fonte : embarcados.com.br - Sinal Analógico x Sinal Digital

O sinal digital, ao contrário do analógico, esse tipo possui apenas um conjunto limitado de valores que podem ser assumidos, dentro da escala de tempo ou amplitude adotada. Como esse tipo de sinal apresenta apenas valores discretos, ele também se caracteriza por arredondar para cima ou para baixo as frequências com valor quebrado.[6]

Figura 3 - Sinal digital



Fonte: oficina Brasil - Sinais Analógicos e Digitais de um Sistema Embarcado

Dentre as GPIO presentes no microcontrolador, quatro podem ser configuradas como saídas PWM. Que é uma técnica para controlar circuitos analógicos a partir de circuitos digitais, utilizando pulsos para gerar um nível de tensão média tendo como referência o *duty cycle* que é a proporção em que o sinal fica em nível alto em relação ao tempo total da onda, com isso gerando a tensão que será emitida pelo PWM, na figura 4 apresenta-se uma ilustração da fórmula do *duty cycle*, onde "Tp" é igual ao tempo em que a onda fica em estado de maior voltagem, posteriormente mostra-se "P" sendo o período total de duração de uma onda a outra.[7]

Figura 4: Formula *duty cycle*

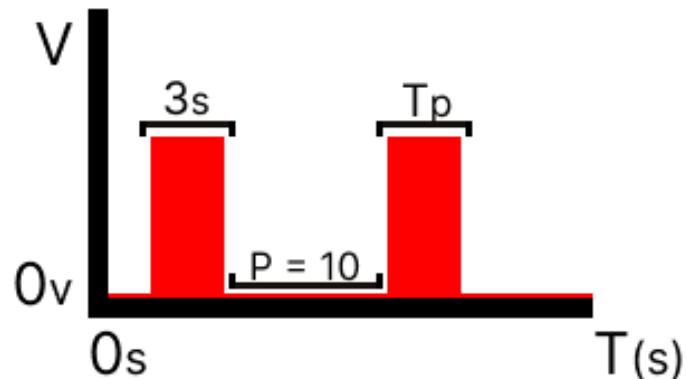
$$\text{Duty cycle} = \frac{\text{Tp}}{\text{P}}$$

Fonte: Elaborado pelo autor

Com o conhecimento desta fórmula podemos analisar a porcentagem de *duty cycle* e transformarmos para a voltagem que será mandada para o componente. [7]

Na figura 5 é exemplificado uma onda digital de período de 10 segundos, e tempo de pulso de 3 segundos, para podermos ter conhecimento sobre a porcentagem de *duty cycle* presente iremos utilizar a fórmula da figura 4, portanto podemos concluir que o *duty cycle* desta onda é de 30%, com este valor é esperado que o valor aplicado na carga que será usado no componente, seja de 30% conforme o valor da tensão máxima do sinal.[8]

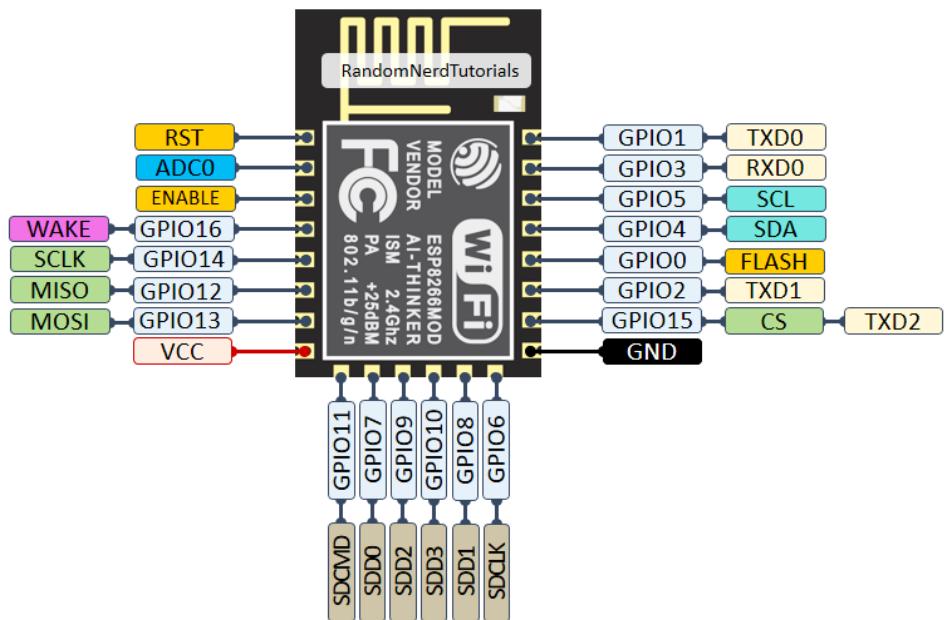
Figura 5: *Duty cycle* com porcentagem de 30%.



Fonte: Elaborado pelo autor

A seguir, apresenta-se uma imagem dos pinos presente no ESP8266 com as respectivas portas GPIO dentre outros pinos de alimentação presentes no ESP8266.

Figura 6: ESP8266 12-E *Chip Pinagem*



Fonte: random nerd tutorials - ESP8266 Pinout Reference

2.2.1.1. NodeMCU

O NodeMCU é uma plataforma open source da família ESP8266 criada para ser utilizada no desenvolvimento de projetos IoT (*Internet of Things*) e também surgiu para facilitar a construção de projetos utilizando o ESP8266, [9]

É o módulo mais popular do ESP8266, possui como vantagem a agregação de uma plataforma de desenvolvimento em linguagem Lua, entretanto, continua-se possível o desenvolvimento na IDE arduino (linguagem C++). Outra grande vantagem é que este modelo possui por uma interface usb-serial, onde através da mesma, é possível tanto criar um código no computador e enviar a placa como facilita a alimentação do microcontrolador pois pode ser realizada por meio de um carregador de celular.[10]

Este módulo apresenta diversas características, dentre essas pôde-se ressaltar o baixo custo, o tamanho reduzido, a facilidade de uso, regulador de tensão de 3,3 V (tensão de operação do ESP8266) e a quantidade de funcionalidades que possui, como a integração de sensores e atuadores já existentes no mercado.[4] Na figura 6 a seguir temos a exibição do NodeMCU.

Figura 6: NodeMCU



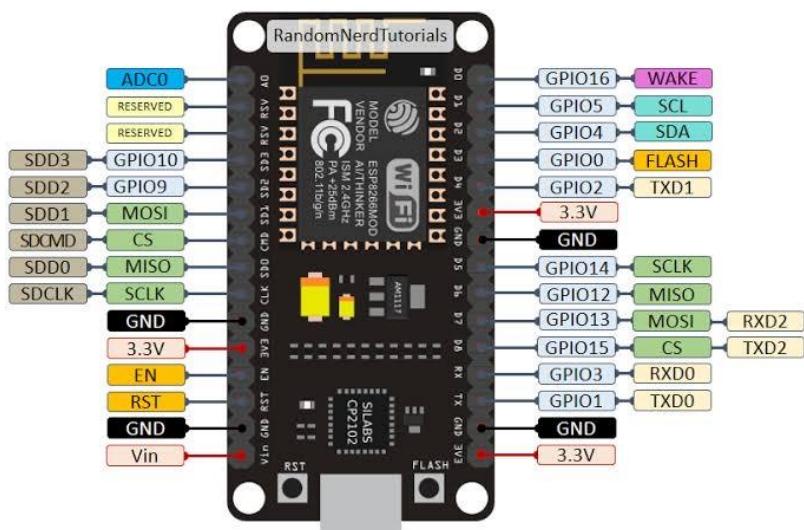
Fonte: Casa da robótica - NodeMCU Amica CP2102

A pinagem que este módulo possui facilita a prototipação e a montagem de circuitos. Este componente abrange áreas residenciais, monitoramentos, rede de sensores, localização,

entre outros, como este módulo possui grande compatibilidade e funcionamento semelhante com a placa arduino, consegue-se usufruir deste módulo no lugar das placas da família arduino e outras. [4]

Na figura 7, apresenta-se a ilustração da pinagem do NodeMCU. Na qual a exibição das portas GPIO que estão interligadas com as respectivas interfaces de entradas e saídas do ESP8266 e estão presentes também algumas interfaces, como três GND (*Ground* - Terra) que funcionam como 0v sendo também associados a negativo de uma bateria, e três saídas 3.3V que são de extrema importância para alimentação de atuadores.[11] Na parte inferior temos 2 botões presentes na placa, um deles sendo o botão RST que basicamente seria o botão de resetar, o qual usamos para reiniciar o código já presente na placa, vale ressaltar que esse botão está interligado com o pino RST do ESP. Esta placa também conta com um botão denominado FLASH que está ligado a porta GPIO0 que por meio do mesmo podemos usar de diversas maneiras pois o mesmo é uma entrada digital, contudo podemos usá-lo para diversas aplicações.[11]

Figura 7: pinagem NodeMCU



Fonte: random nerd tutorials - ESP8266 12-E NodeMCU

2.2.1.2. PROGRAMAÇÃO

O NodeMCU pode ser programado através de programas denominados de *sketch*, para que o microcontrolador seja programado e assim se torne autônomo é necessário o uso de um *software*, no qual basicamente sua função será de escrever, salvar, compilar e gravar o código na placa, após a gravação o código gravado será executado automaticamente quando a placa

for alimentada.[12]

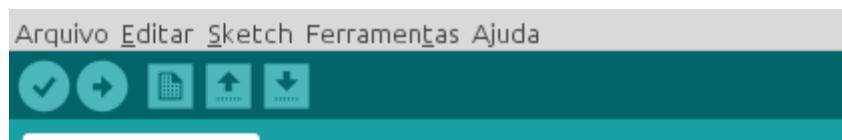
O NodeMCU pode ser programado através de scripts escritos na linguagem Lua, ou através IDE Arduino utilizando a linguagem C++ .[4] A linguagem C++ é uma das mais usadas em trabalhos acadêmicos, por diversos fatores, entre eles: por ser uma linguagem de programação mais consolidada na literatura, sendo ensinada na maioria das universidades e escolas técnicas, o que facilita a busca por materiais de consulta como artigos e trabalhos acadêmicos.[10]

Por outro lado, temos a linguagem Lua na qual não temos uma grande quantidade de material para consulta tanto na internet como em projetos usando a mesma integrada com NodeMCU. [10]

2.2.1.2.1. Software de codificação

Uma das maneiras de programar o NodeMCU é através da utilização da IDE do Arduino, esta IDE é *open source*, através desse ambiente podemos usar bibliotecas que utilizam a linguagem C++ entre outras ferramentas fornecidas pelo ambiente.[13] Na figura 8, a uma demonstração da tela principal do *software* IDE Arduino

Figura 8: IDE Arduino



```
Arquivo Editar Sketch Ferramentas Ajuda
sketch_jun26a
void setup() {
  // put your setup code here, to run once:
}

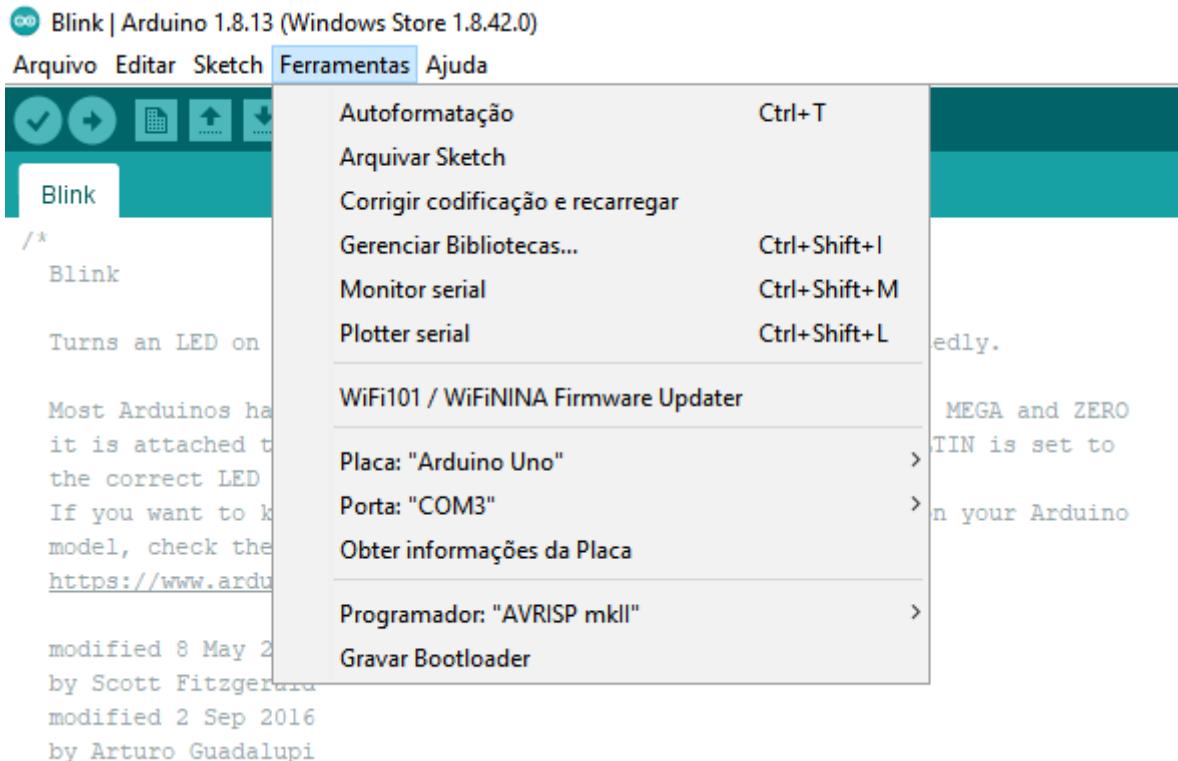
void loop() {
  // put your main code here, to run repeatedly:
}
```

Fonte:Elaborado pelo autor

Usar este ambiente de desenvolvimento traz diversas vantagens para os usuários entre elas: compilador integrado, ferramentas para que gerenciamento do código seja otimizado

como por exemplo: monitor serial, o qual permite o monitoramento de dados obtidos pelo microcontrolador em tempo real.[10] Na figura 9 apresenta-se as ferramentas contidas no software descrito.

Figura 9: Ferramentas IDE Arduino

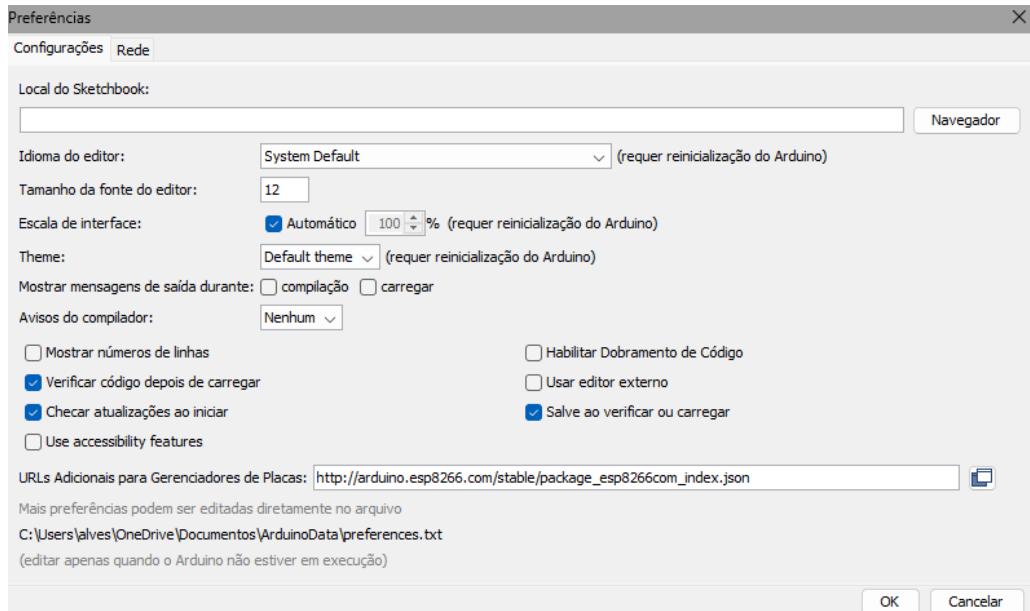


Fonte: Lobo da Robótica - IDE do Arduino: O Guia Definitivo!

Entretanto esse ambiente de desenvolvimento não é nativo para ESP8266 para a utilização do mesmo é necessário realizar algumas modificações de preferências e gerenciamento de placas.[4]

Inicialmente é necessário em “Arquivos” posteriormente “Preferências” e adicionar ao campo “URLs Adicionais de Gerenciadores de Placas” a URL http://arduino.esp8266.com/stable/package_esp8266com_index.json. Na figura 10 ilustra-se a tela de Preferências da IDE arduino.[4]

Figura 10: Preferências IDE Arduino.



Fonte: Internet das Coisas com ESP8266, Arduino e Raspberry Pi. 1^a edição.

Logo após, em “Ferramentas”, “Placa” e em “Gerenciador de Placas” é necessário instalar a versão mais recente disponível do suporte à placa ESP8266. Em seguida, é possível utilizar o ambiente normalmente, executando todas as ferramentas presentes com o módulo NodeMCU ESP8266 e demais variantes.[4]

2.3. SENsores

Sensores são dispositivos eletrônicos utilizados para obtenção de dados, efetuando a função de medir algum estímulo recebido pelo ambiente externo, relacionando esses dados com alguma grandeza física que precisa ser mensurada, sendo assim a entrada obtida.[12]

Esse dispositivo não possui capacidade de converter a entrada conquistada em dados comprehensíveis, portanto é necessário usar um componente chamado transdutor, que tem a função de converter a grandeza física recebida em uma grandeza elétrica comprehensível e manipulável.[14]

Em relação a saída pode se obter um sinal do tipo analógico ou digital.[12] Analógicos são aqueles que podem assumir infinitos valores, portanto é um tipo de sinal contínuo, como representação numérica deste sinal podemos ter todos os valores intermediários de uma determinada faixa de números.[6] Digitais são aqueles que estão limitados a uma saída

binária sendo esta saída 0 ou 1.[14]

2.3.1. Sensor reflexivo Infravermelho HW-201

Este sensor reflexivo Infravermelho é um módulo eletrônico capaz de detectar objetos que estejam em sua frente.[15] Esta detecção se dá por meio de um LED emissor infravermelho sendo este o responsável por emitir a luz infravermelha e um LED receptor infravermelho (fotodiodo) sendo o responsável por detectar o sinal infravermelho, posicionados na parte frontal da placa.[16] A figura 11 apresenta o sensor Sensor reflexivo Infravermelho HW-201.

Figura 11: Sensor HW-201



Fonte: Usinainfo - Sensor de Obstáculo Reflexivo Infravermelho HW-201 com Ajuste de Sensibilidade

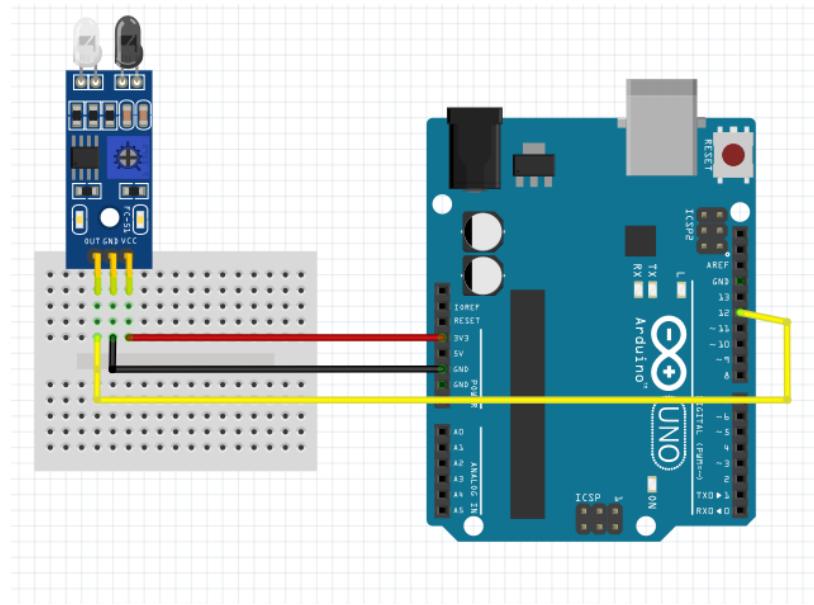
2.3.1.1. Funcionamento

O funcionamento desse sensor consiste em a todo momento que estiver ligado, o LED emissor emite diferentes luzes infravermelhas, as quais rebatem no obstáculo e são captadas pelo receptor. [16]

O sinal de saída deste sensor sempre está em estado *HIGH* mudando apenas para *LOW* quando um obstáculo for detectado em seu campo de monitoramento, o valor de retorno dado pelo sensor é 0 ou 1 sendo assim uma entrada digital.[14] O sensor HW-201 possui a capacidade de operar em 3.3V ou 5V, sendo assim compatível com Arduino, ESP e outros microcontroladores[15]. Este Módulo possui uma grande característica em sua composição sendo ela um *trimpot* no qual tem a funcionalidade de fazer o ajuste de sensibilidade de detecção do sensor, variando de 2 cm a 30cm [16]. Abaixo segue uma ilustração de como

realizar a ligação do sensor com um Arduino.

Figura 12: Ligação sensor HW-201 com Arduino

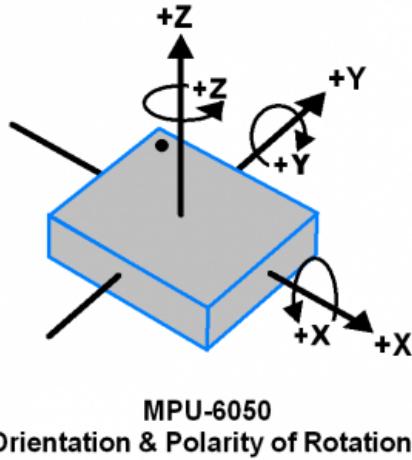


Fonte: Elaborado pelo autor

2.3.2. Sensor MPU-6050

O sensor MPU-6050 é um modelo que combina um giroscópio e um acelerômetro, ambos possuindo 3 eixos, também detém 3 conversores Analogicos para digital sendo de 16 bits para realizar a digitalização dos resultados do giroscópio, também possui mais 3 conversores idênticos para realizar a mesma função com os dados obtidos pelo acelerômetro. Na figura 13 é possível observar os eixos de medição do acelerômetro e giroscópio do MPU-6050. [17]

Figura 13: Eixos de medição do acelerômetro e giroscópio do MPU-6050



Fonte: Makergram

Giroscópio é um dispositivo que mede a variação do ângulo, realizando a conversão de quanto rápido o sensor está girando em graus por segundo.[18]

O acelerômetro exerce a função de medir acelerações, tendo como referência forças externas que agem sobre o mesmo.[18]

A comunicação entre o ESP8266 com o MPU-6050, ocorre através do tipo de comunicação I2c, por meio dos pinos denominados SCL e SDA, os pinos AD0 presente no módulo é usado para modificar o endereço de acesso ao sensor, possibilitando a utilização de uma maior quantidade de sensor, os pinos XDA, SCI e INT são utilizados para o controle de dispositivos externos, da mesma forma , dispõe de pinos VCC e GND para a alimentação do mesmo. Na figura 14, apresenta-se o sensor MPU-6050 e seus 8 pinos.[17]

Figura 14: MPU-6050



Fonte: MakerHero - Acelerômetro e Giroscópio 3 Eixos 6 DOF MPU-6050

2.4. ATUADORES

Atuadores são dispositivos que exercem a função de converter energia elétrica, hidráulica ou pneumática em potência mecânica. Por meio de um sistema de transmissão de potência, essa força gerada é enviada a um eixo para se realizar a locomoção.[19]

Neste trabalho estão empregados motores elétricos nos quais estão presentes na família dos atuadores elétricos, geralmente são utilizados em robôs de pequeno e médio porte, por possuírem maior precisão comparados com outros atuadores.[20]

2.4.1. Motores elétricos

Motores elétricos possuem componentes indispensáveis um deles chamado estator que é uma parte fixa do motor, que pode conter um ou mais enrolamento por polo, para receber corrente e produzir o campo magnético fixo, outro componente presente é o rotor no qual é uma parte móvel do motor ligado ao eixo de transmissão de movimento.[21] Para o funcionamento do motor uma corrente é aplicada no estator e uma outra tensão de corrente é aplicada no rotor. [21] Na figura 15 apresenta-se os componentes citados que estão presentes nos motores elétricos entre outros.

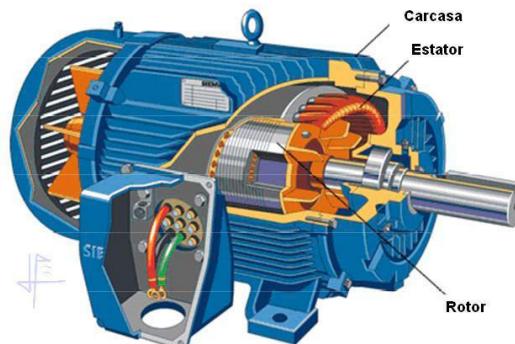
Figura 15: Componentes motores elétricos



Fonte: revistamanutenção.com.br - Motores elétricos: componentes e suas aplicações

Esses tipos de motores possuem duas variações sendo elas motores de corrente alternada e corrente contínua.[22] Os motores de corrente alternada são motores cuja a fonte alimentação apresenta uma corrente que possui alterações, ou seja, variam com uma determinada frequência. [22] São os motores mais utilizados, por terem um custo mais baixo e uma vez que a distribuição de energia elétrica é feita principalmente em corrente alternada, e também por funcionarem com uma velocidade fixa proporcionando grandes potências.[22] Na figura 16 ilustra-se um motor de corrente alternada com ênfase em seus principais componentes.

Figura 16: Motor de corrente alternada



Fonte: eletrovalmotores.com.br - Motor de corrente alternada

Por outro lado, os motores de corrente contínua são de custo mais elevado e precisam de uma fonte de corrente contínua.[23] Estes motores precisam de uma manutenção frequente e devido ao baixo rendimento não são tão utilizados atualmente[22]. Entretanto ao contrário dos motores de corrente alternada este pode funcionar com velocidades ajustáveis também possui fácil controle pois seu controle se dá pela corrente aplicada.[23] Na figura 17 temos exemplos de motores de corrente contínua.

Figura 17: Motor de corrente contínua

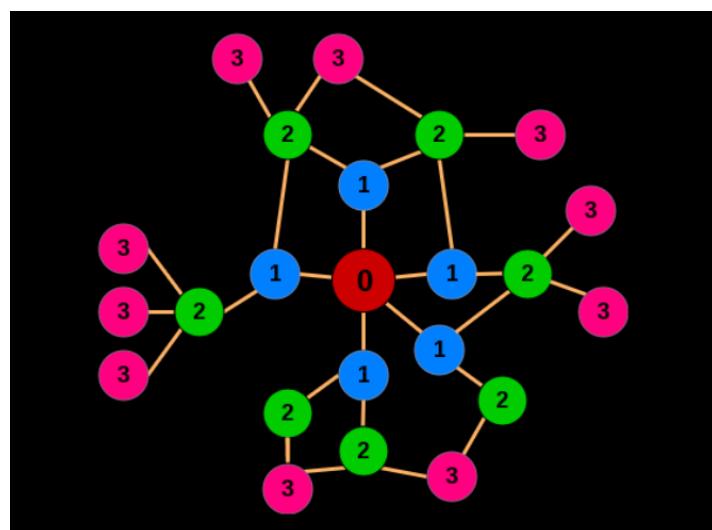


Fonte: mundodaeletrica- Motor de corrente contínua, características e aplicações

2.5. ALGORITMO DE BUSCA EM LARGURA

O algoritmo de busca em largura é utilizado para determinar o menor caminho geral de um ponto a outro, o funcionamento deste algoritmo segue a ideia de: fazer uma busca a partir de um ponto inicial posteriormente é feita uma busca em todos os nós (caminhos) adjacentes/vizinhos e para cada um desses a busca se repete para todos os nós adjacentes e assim repetindo até encontrar o ponto final.[24] Na figura 18 temos a representação deste algoritmo fazendo a busca em todos os vizinhos adjacentes de um nó.

Figura 18: Algoritmo de busca em largura



Fonte: blog pantuza - BUSCA EM LARGURA

Esse algoritmo se caracteriza por encontrar o menor caminho manualmente, pois faz a busca em todos os caminhos possíveis, e o escolhido é o que encontrar o ponto final em um menor número de caminhos percorridos.[24] Para encontrar esse caminho é necessário o uso de uma fila que funciona exatamente como uma fila de vida real, que utiliza-se para fazer o gerenciamento de prioridades de caminhos assim ao final do processo de busca basta seguir um caminho contrário de acordo com a prioridade de cada caminho dada pela fila auxiliar, assim descobrindo qual caminho da matriz foi resultante do menor caminho.[25]

3. Metodologia

Durante a elaboração do protótipo foi utilizada uma metodologia de pesquisa. O trabalho se caracteriza por empregar uma abordagem qualquantitativa por se realizar uma análise quantitativa dos dados, posteriormente, foi realizada uma análise subjetiva. Possui uma natureza aplicada por ter como objetivo gerar conhecimentos novos para serem aplicados logo após. Conforme os objetivos, caracteriza por ser exploratória, pois busca maior compreensão de um fato pouco estudado na literatura. Quanto aos procedimentos técnicos empregados neste trabalho são de pesquisa bibliográfica por serem utilizadas outros trabalhos como fontes.

3.1. Componentes

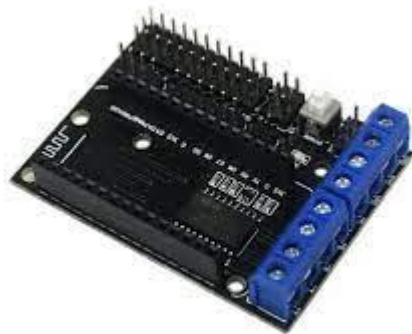
Para a realização deste projeto ser possível, foram utilizados determinados componentes além de conhecimento necessário sobre cada um separadamente.

Dentre os componentes está o principal sendo o microcontrolador NodeMCU ESP8266. Através da IDE do Arduino sua programação foi realizada.

Também são importantes sensores HW-201 que farão captação de sinais, que logo após irão ser interpretadas pelo protótipo. Esse sensor trata-se de um modelo reflexivo Infravermelho que é capaz de detectar objetos que estejam em sua frente, através de um LED emissor infravermelho o responsável por emitir uma luz infravermelha e outro LED sendo o receptor infravermelho sendo o responsável por detectar o sinal infravermelho.

Além da essencial utilização de motores de corrente contínua para o robô se locomover no ambiente inserido, estão sendo usados motores de 30 RPM de 6v, controlados por uma motor *shield* que possui em sua composição uma ponte H L293D, assim possibilita fazer o controlamento de até dois motores, através do acoplamento de um NodeMCU ESP8266 nesta placa. Na figura 19, verifica-se a apresentação deste considerável componente.

Figura 19: Motor shield

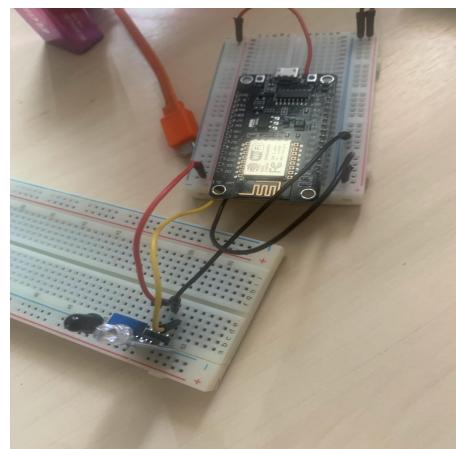


Fonte: mundoeletronica - Shield Driver L293d

Para a execução deste trabalho foi necessário a realização de testes de bancada de cada um dos componentes, sendo assim adquirindo conhecimento sobre o funcionamento separadamente para logo após aplicarmos todos em conjunto.

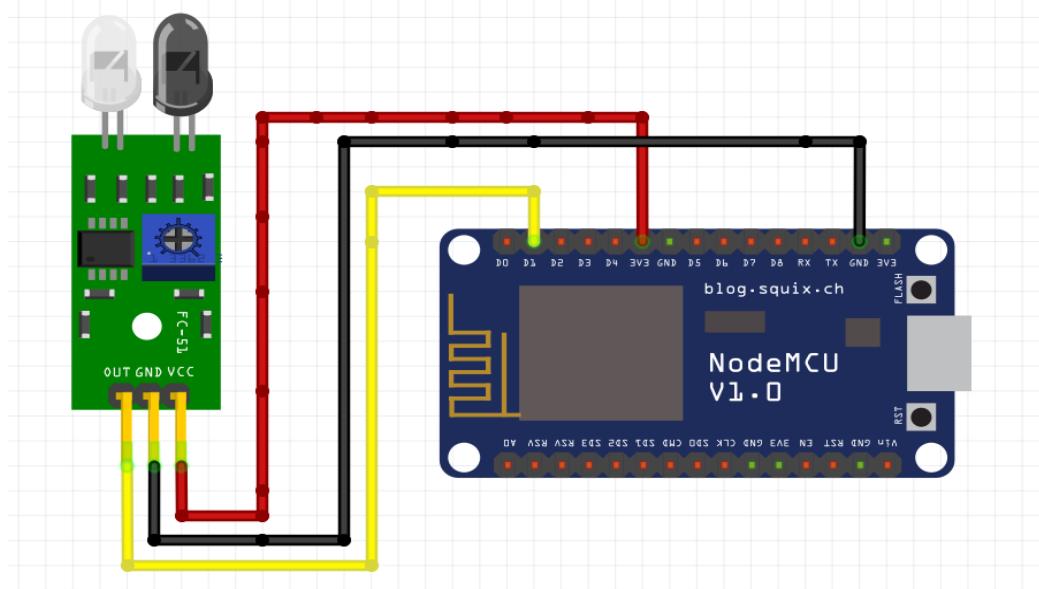
Primeiramente, foi realizado o teste de bancada com o sensor HW-201 utilizando um NodeMCU ESP8266. Esse teste tinha como objetivo obter os valores do sensor com este microcontrolador, além da calibração do potenciômetro presente no sensor para ajustar a distância que o sinal infravermelho percorrerá. Na figura 20, verifica-se a ilustração de como o teste ocorreu.

Figura 20: Teste sensor com NodeMCU



Fonte: Elaborado pelo autor

Figura 21: Circuito Teste sensor com NodeMCU



Fonte: Elaborado pelo autor

Na figura 21, há a apresentação do circuito utilizado para produzir as conexões com o micro controlador.

A fim de realizar o teste há necessidade da produção de um código conforme ilustrado na figura 22. O qual realiza a leitura do sensor de forma digital, armazena em uma variável e mostramos o valor da variável no monitor serial.

Figura 22: Código teste de bancada com sensores

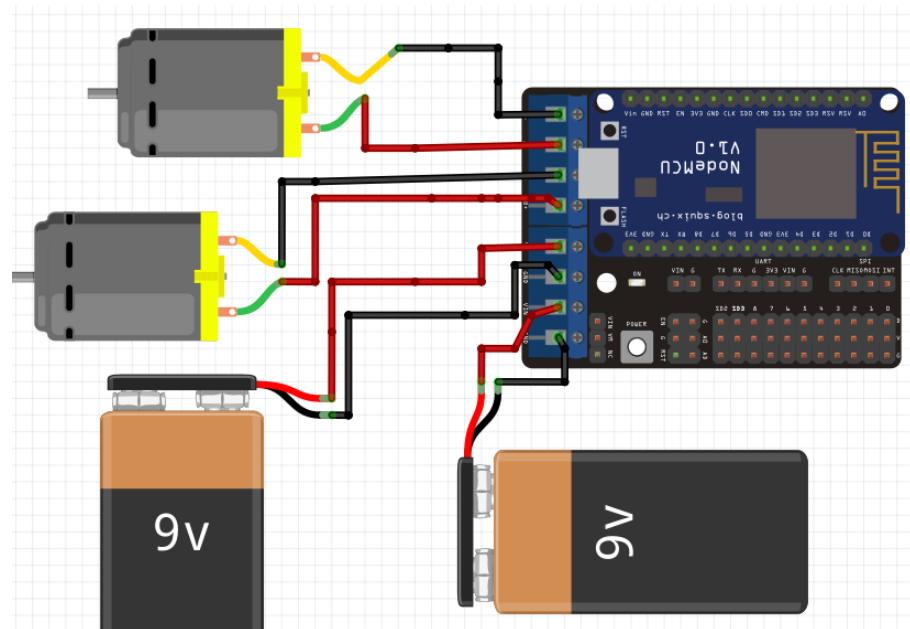
```
int sensor = 5; //Definindo porta do sensor
int valor = 0;// variavel que recebera valor do sensor

void setup()
{
    Serial.begin(115200); // Iniciando a comunicacao serial na frequencia 115200
    pinMode(sensor, INPUT); //Setando o Pino 5 como entrada
}
void loop()
{
    val = digitalRead(sensor); //Lendo o valor de entrada do sensor e armazenando
    Serial.println(valor); //Mostrando n monitor serial o valor do sensor
    delay(100);
}
```

Fonte: Elaborado pelo autor

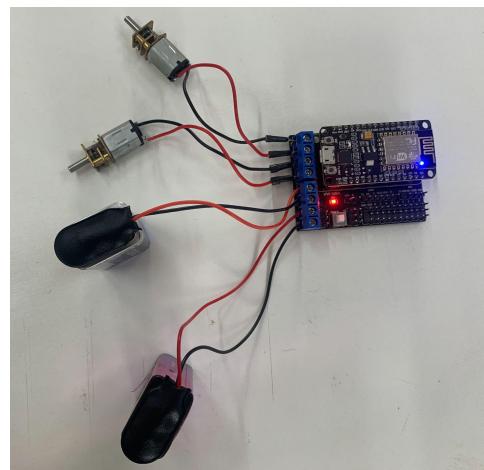
O segundo teste realizado decorreu sobre o uso do motor *Shield* para NodeMCU acoplado com uma ponte H com dois motores conectados, para o teste ser possível realizou-se a criação de um circuito como ilustrado na figura 23. Na figura 24, apresenta-se o teste de bancada realizado.

Figura 23: Circuito teste de ponte H com motores



Fonte: Elaborado pelo autor

Figura 24: Teste de bancada com motores



Fonte: Elaborado pelo autor

O código empregado neste teste apresenta-se na figura 25. Onde primeiramente declaramos as portas utilizadas pela *Shield* para controle de motores nas quais são D1 e D2 portas PWM que são empregadas para definir a velocidade dos motores sendo de 0 a 1023, D3 e D4 usadas para definir a direção dos motores sendo *HIGH* ou *LOW*. abaixo na função setup foi definido o estado do pino no qual por serem motores são definidos OUTPUT, na função loop sucedeu o desenvolvimento de um código para realizar o movimento dos motores em qualquer sentido durante um determinado tempo e logo após mudar a direção para o sentido oposto, como este código foi desenvolvido na função loop ele será executado sucessivamente.

Figura 25: Código teste de bancada com motores

```
const int pwmMotorA = D1;
const int pwmMotorB = D2;
const int dirMotorA = D3;
const int dirMotorB = D4;

int motorVelocidade = 500;

void setup()
{
    pinMode(pwmMotorA, OUTPUT);
    pinMode(pwmMotorB, OUTPUT);
    pinMode(dirMotorA, OUTPUT);
    pinMode(dirMotorB, OUTPUT);
}

void loop()
{
    analogWrite(pwmMotorA, motorVelocidade);
    digitalWrite(dirMotorA, LOW);
    analogWrite(pwmMotorB, motorVelocidade);
    digitalWrite(dirMotorB, LOW);
    delay(2500);

    analogWrite(pwmMotorA, motorVelocidade);
    digitalWrite(dirMotorA, HIGH);
    analogWrite(pwmMotorB, motorVelocidade);
    digitalWrite(dirMotorB, HIGH);
    delay(2500);
}
```

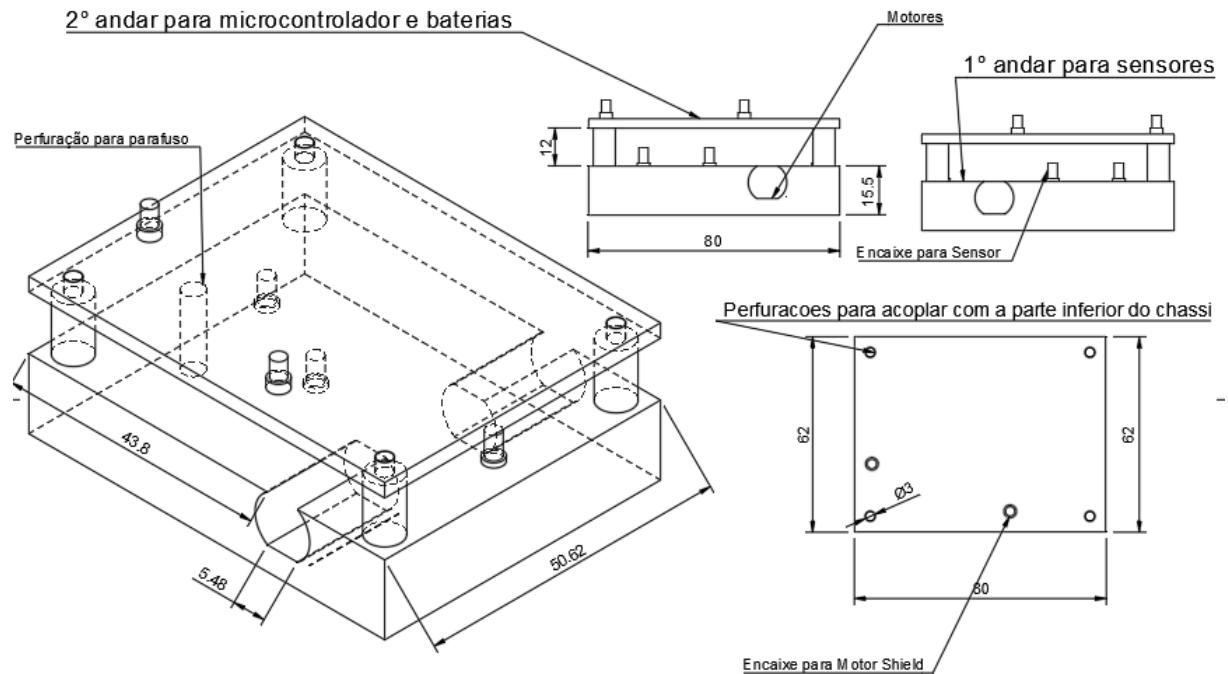
Fonte: Elaborado pelo autor

3.1.1. Estrutura

Após concluir os testes com os componentes separadamente, foi necessário projetar a

estrutura do robô em 3D para garantir que todos os componentes sejam bem posicionados assim garantindo a melhor eficiência dos mesmos. Na figura 26, temos a demonstração da estrutura feita através do software Fusion 360 da Autodesk.

Figura 26: Estrutura



Fonte: Elaborado pelo autor

No primeiro andar, serão aplicados os sensores e motores que contam com uma perfuração para cada motor, possui também três suportes pertencendo um para cada sensor assim sendo possível posicioná-los para as suas direções respectivas, na retaguarda desta parte possui uma cavidade para a entrada de um parafuso utilizado como terceira roda de apoio.

O segundo andar detém dois suporte para o acoplamento da motor shield juntamente com o NodeMCU e também possui espaço suficiente para alojar duas baterias

3.1.2. Labirinto

Posteriormente, a determinação dos componentes e desenvolvimento da estrutura, ocorreu a necessidade de criação de um labirinto flexível para realização de testes de bancadas. Esse significativo componente do trabalho foi planejado para possuir 6 linhas e 6

colunas de caminhos e 5 e 5 respectivamente de possíveis paredes, além disso, dispor de paredes em suas laterais. Na figura 27, apresenta-se a ilustração do labirinto moldável desenvolvido.

Figura 27:Labirinto moldavel



Fonte: Elaborado pelo autor

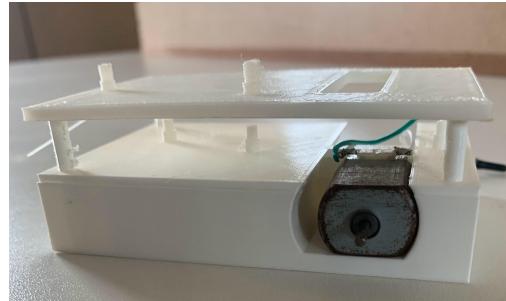
Deve-se ressaltar, que este elemento possui paredes de diferentes tamanhos sendo 15 cm e 12 cm para ser tornar mais fácil o encaixe, da mesma forma facilitar a locomoção do protótipo tendo em vista a largura da estrutura, destaca-se também por possuir medida total de 90x90 cm, havendo em seu piso perfurações a cada 15 cm para demarcar local de acoplamento das paredes

3.2. Desenvolvimento

Após os testes de bancadas realizados separadamente, os mesmos proporcionaram um maior conhecimento sobre todos os componentes. Assim, deu-se início à construção do protótipo.

Inicialmente, o desenvolvimento do protótipo começou com a criação de um modelo 3D, a primeira versão contou com dois motores DC sem caixa de redução, 3 sensores HW-201, motor *shield* com NodeMCU, suporte para duas baterias de 4.2 V e por fim perfuração esférica para acoplamento de uma bola de vidro, entretanto, após a impressão deste modelo notou-se que certas medidas como a de encaixe dos motores estavam inadequadas para uso. Na figura 28, apresenta-se o erro desta etapa.

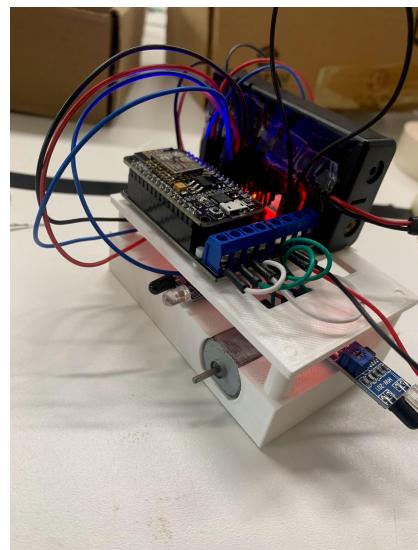
Figura 28 :Erro de medida protótipo



Fonte: Elaborado pelo autor

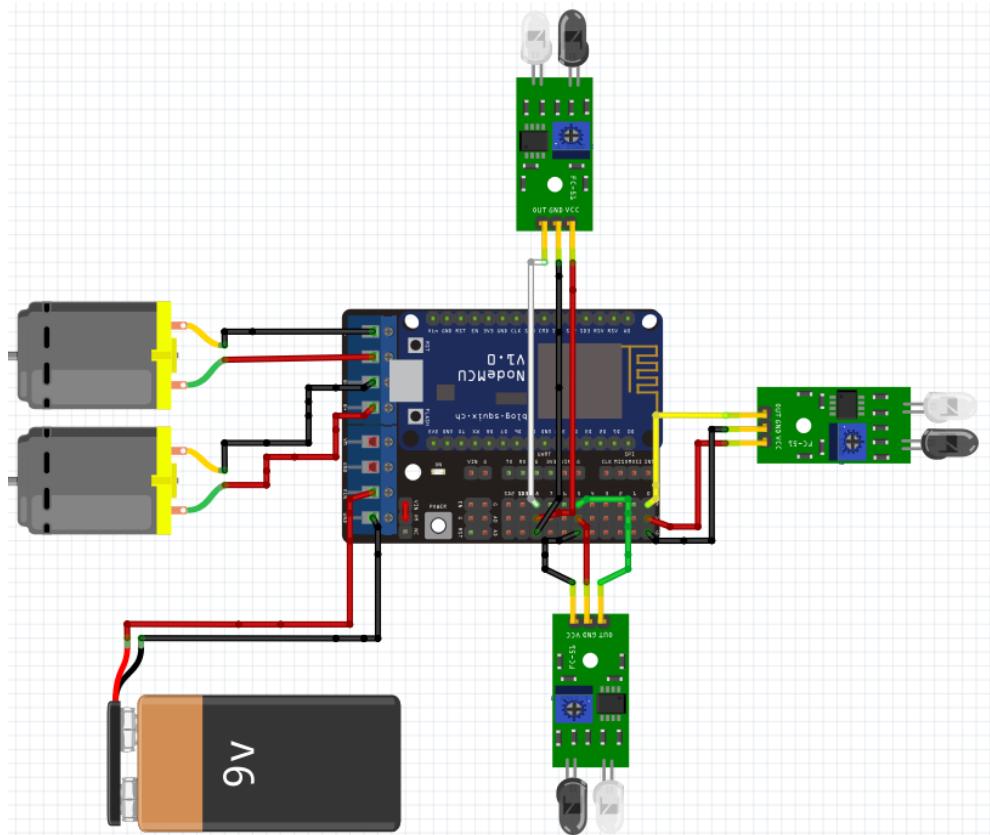
Após este problema ser descoberto, foi realizada a correção para ser impresso novamente. Entretanto, ao decorrer do trabalho ocorreram cerca de 9 erros referentes a alguma medida, sendo assim necessário a impressão de 9 vezes da estrutura para serem alcançadas as medidas exatas para o segundo protótipo, assim sendo possível haver testes com todos os componentes interligados. Na figura 29 ilustra-se o segundo protótipo já implementado e na figura 30 apresenta-se o circuito utilizado no segundo protótipo.

Figura 29:Segundo prototipo



Fonte: Elaborado pelo autor

Figura 30:Circuito segundo prototipo



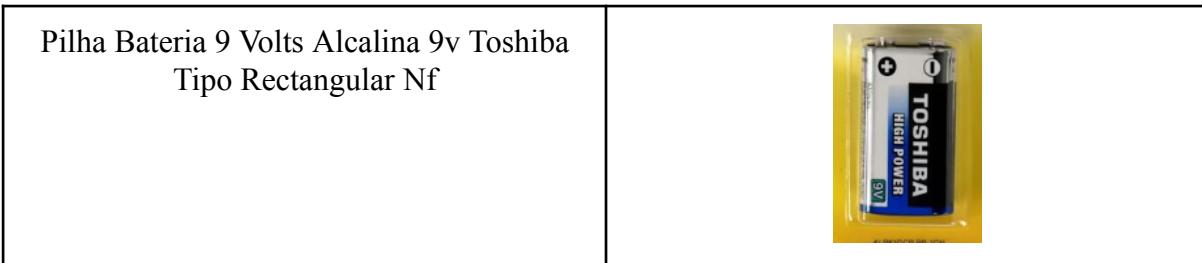
Fonte: Elaborado pelo autor

Desse modo, testes foram realizados com este protótipo a fim de confirmar as capacidades desse modelo, sem demora, notou-se que os motores utilizados não seriam eficazes para tal trabalho por possuírem pouco torque e baixa compatibilidade com rodas disponíveis, assim sendo necessário trocá-los por outra opção que fornecesse maior desempenho e maior uniformidade, também percebeu-se que a bola de vidro utilizada como terceira roda não era eficaz tendo em vista o acúmulo de resíduos assim deixando o protótipo mais lento, além disso, constatou que as baterias usadas não possuíam a capacidade de alimentar todos os componentes.

Dessa maneira, foram adquiridos mini motores DC com caixa de redução de 300 RPM e rodas compatíveis com os mesmos, e com o propósito de extinguir o uso da bola de vidro realizou-se uma perfuração na parte posterior para o alojamento de um parafuso, desta forma, melhorando a locomoção do robô, ao observar que as baterias não estavam conseguindo entregar a melhor eficiência as mesmas passaram-se a serem substituídas por duas de 9V alcalinas, sendo uma para alimentação do microcontrolador e sensores logo a outra somente para os motores. deve ressaltar que não houve necessidade de alteração da estrutura base para adicionar os novos componentes apenas ocorreu a instalação de suportes para alojamento das baterias. A tabela 1 indica os novos componentes utilizados e na figura 31 o terceiro protótipo implementado.

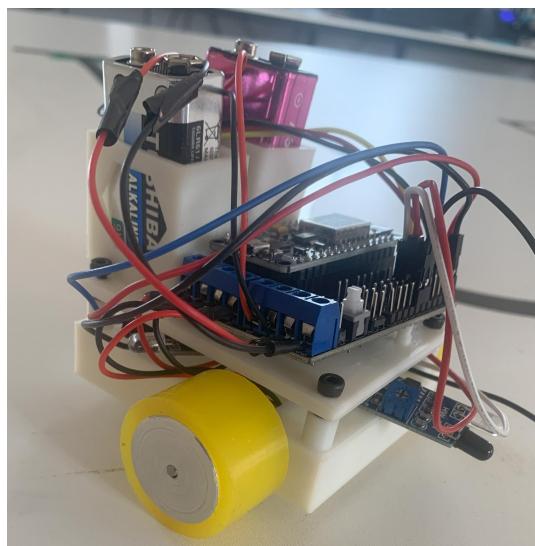
Tabela 1: Motores e Rodas obtidos

Motor 300 RPM, 6V	
Roda StickyMAX S15 - Macia - 32mm	



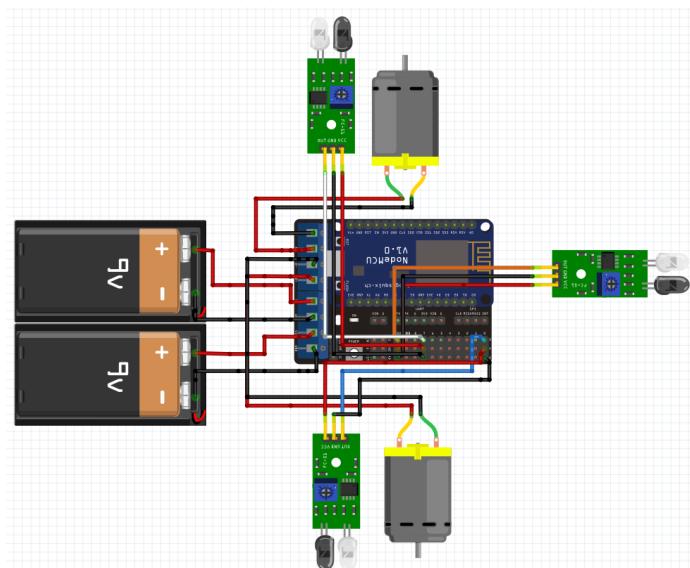
Fonte: Elaborado pelo autor

Figura 31: Terceiro protótipo



Fonte: Elaborado pelo autor

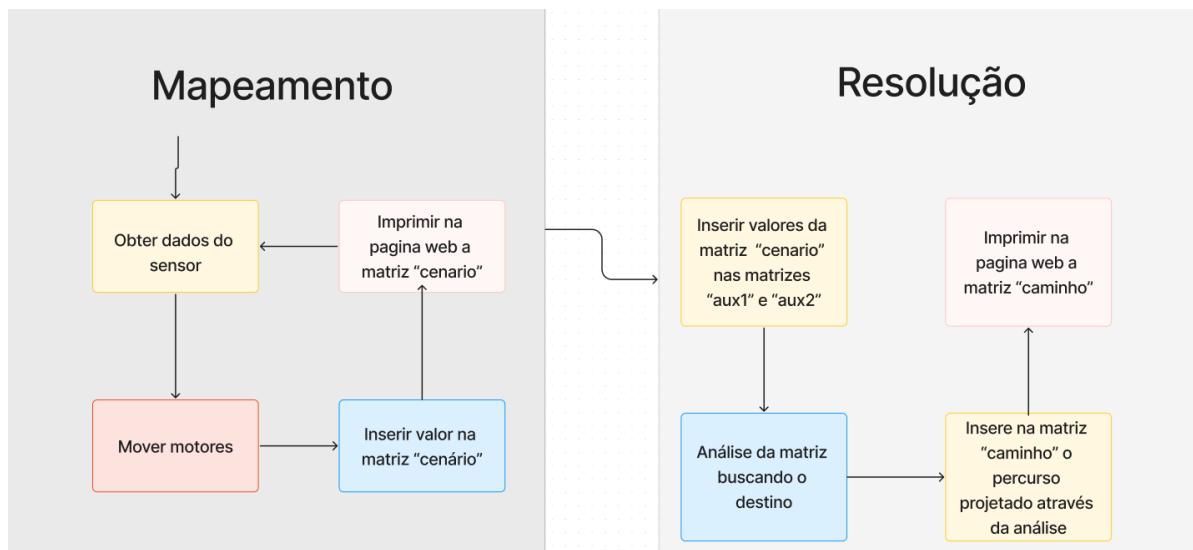
Figura 32:Circuito Terceiro Protótipo



Fonte: Elaborado pelo autor

Com o terceiro protótipo montado, iniciou a fase da programação, para assim se realizar testes de bancada e se necessário buscar melhorias. A princípio o código principal foi dividido em duas vertentes sendo o mapeamento do labirinto e logo após a resolução do mesmo. O código desenvolvido nesta etapa do trabalho cujo objetivo busca validação do protótipo apresenta-se no apêndice A. Na figura 33 ilustra o fluxograma do código principal implementado.

Figura 33: Fluxograma



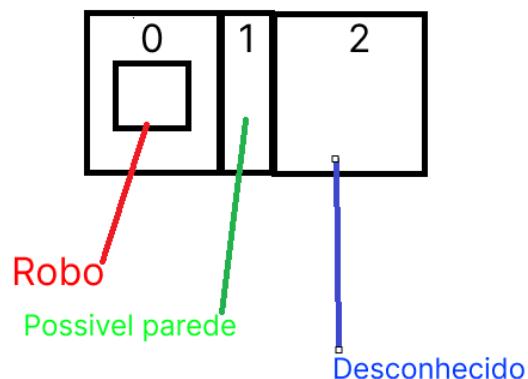
Fonte: Elaborado pelo autor

Conforme a Figura 33, a qual apresenta o fluxograma do código desenvolvido, inicialmente é feita a obtenção dos valores dos sensores, sendo 0 ou 1 por serem digitais, como o protótipo apresenta 3 sensores a 8 possibilidades paredes, dessa maneira após análise dos valores será constatado qual tipo de movimento o robô fará sendo para esquerda e direita em 90 graus, para frente e retorno de 180 graus se necessário.

Desse modo, a próxima etapa consiste na inserção de valores em uma matriz que constará todos os caminhos mapeados, no primeiro momento essa matriz contém apenas 0 em suas linhas e colunas representando um espaço desconhecido, logo após cada movimentação será inserido na mesma matriz o valor 1 duas vezes, representando um caminho livre, a inclusão de dois valores acontece pois todos os números ímpares da matriz são possíveis paredes, esta característica ocorre em razão de que é feita o mapeamento através das paredes próximas, tendo em vista que não há conhecimento de que na retaguarda haverá um percurso

livre ou fechado, deve se destacar que essa matriz possui o valor 0 como representante de caminho desconhecido e parede, 1 como caminho livre e -99 e 999 como origem e destino respectivamente. Na figura 34 ilustra-se a característica da inserção de valores.

Figura 34:Caracteristica labirinto



Fonte: Elaborado pelo autor

A seguir ocorre a impressão da matriz representando o cenário em uma página *web* criada pelo ESP8266 utilizado neste trabalho, esta função auxilia na visualização do mapeamento em tempo real. Com finalidade de visualizar o labirinto simultaneamente, para essa funcionalidade foi necessário a inserção no código de uma biblioteca denominada “ESP8266WiFi” que será responsável por toda a comunicação do microcontrolador com a rede de internet, a mesma fará a criação de um servidor na porta 80 onde ficará a página *web* que será acessada. No Apêndice B apresenta-se um exemplo de código responsável por desenvolver uma página *web* simples.

O código responsável pelo mapeamento está em um loop, desta maneira, o loop se encerra quando se encontra o final do labirinto mapeado. Por consequência inicializando a fase de busca do menor caminho possível.

Dessa maneira, inicialmente é feita a criação de 3 matrizes, no qual serão utilizadas para auxiliar a busca e expor o menor caminho posteriormente, duas matriz receberão os valores do cenário obtido anteriormente, destacando a origem e o destino, consequentemente, é realizada a busca de todos os caminhos possíveis através da análise de uma matriz auxiliar, assim gerando todos os possíveis caminhos e a contagem de passos necessários para chegar ao

destino final. Na figura 18, ilustra-se o que esta fase da busca realizará, no apêndice A, a função “busca” apresenta o código desenvolvido para realizar esta tarefa descrita.

Consequentemente, nesta etapa detém todos os caminhos possíveis para alcançar o destino, desta forma, é produzida uma análise de uma matriz auxiliar contendo esses valores, essa análise busca encontrar o menor caminho possível priorizando linhas retas, enquanto o percurso é gerado, o mesmo é atribuído a uma matriz que conterá apenas o caminho final. Como resultado é elaborada a impressão da matriz resultante no monitor serial e na página *web* para visualização. Na figura 35, exibe-se o resultado da busca em um cenário mapeado utilizando o código desenvolvido onde os números em sequências representam o caminho gerado com a quantidade de passos necessários para percorrer da origem ao destino, no apêndice A apresenta-se o algoritmo desenvolvido e utilizado nesta etapa.

Figura 35:Busca em largura aplicado

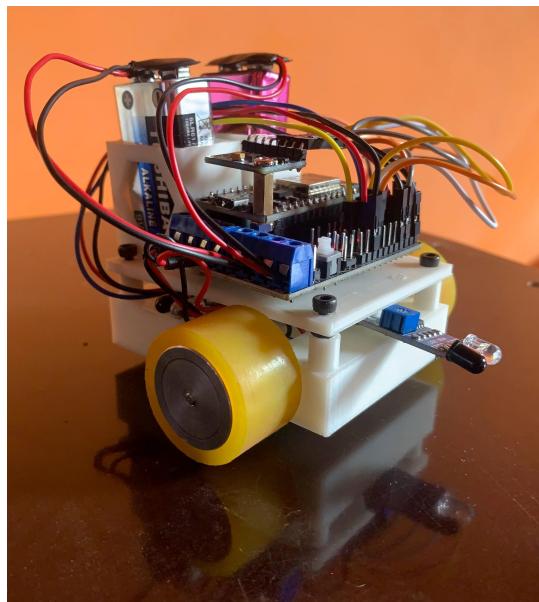
```
|0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0|
|0| |1| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0|
|0| |2| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0|
|0| |3| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0|
|0| |4| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0|
|0| |5| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0|
|0| |6| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0|
|0| |7| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0|
|0| |8| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0|
|0| |9| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0|
|0| |10| |11| |12| |13| |14| |0| |0| |0| |0| |0| |0| |0| |0| |0|
|0| |10| |0| |0| |15| |0| |0| |0| |0| |0| |0| |0| |0| |0| |0|
|0| |0| |0| |0| |0| |16| |17| |18| |19| |20| |21| |22| |1001| |0|
```

Fonte: Elaborado pelo autor

Entretanto, ao realizar testes com possíveis cenários notou-se que havia extrema dificuldade em determinar os valores para a locomoção, havendo necessidade de auxiliar o robô manualmente para completar um percurso, desta maneira, visando o melhor desempenho do protótipo a fim de concluir todos os objetivos propostos. Ocorreu a substituição do motor de 300 RPM para um de 30 RPM, outrossim, a instalação de um sensor giroscópio MPU6050, no qual tem a função de determinar o ângulo em que o protótipo está

em relação ao seu estado inicial, assim sendo possível determinar com precisão as funções de movimento, tornando o protótipo mais independente de ações externas, O quarto protótipo elaborando ilustra-se na figura 36.

Figura 36: Quarto protótipo



Fonte: Elaborado pelo autor

Para estabelecer as rotinas de movimento foi necessário desenvolver um código que obtém os valores dos ângulos, o código desenvolvido apresenta-se na figura 37, no qual utilizou-se a biblioteca “MPU6050_tockn” para que se tornasse possível a utilização do sensor, além da utilização da biblioteca “Wire” para realizar a conexão I2c entre o microcontrolador e o sensor. O circuito utilizado no quarto protótipo com a implementação do sensor MPU 6050 apresenta-se na figura 38.

Figura 37: Código para obtenção de ângulo com MPU6050

```
#include <MPU6050_tockn.h>
#include <Wire.h>

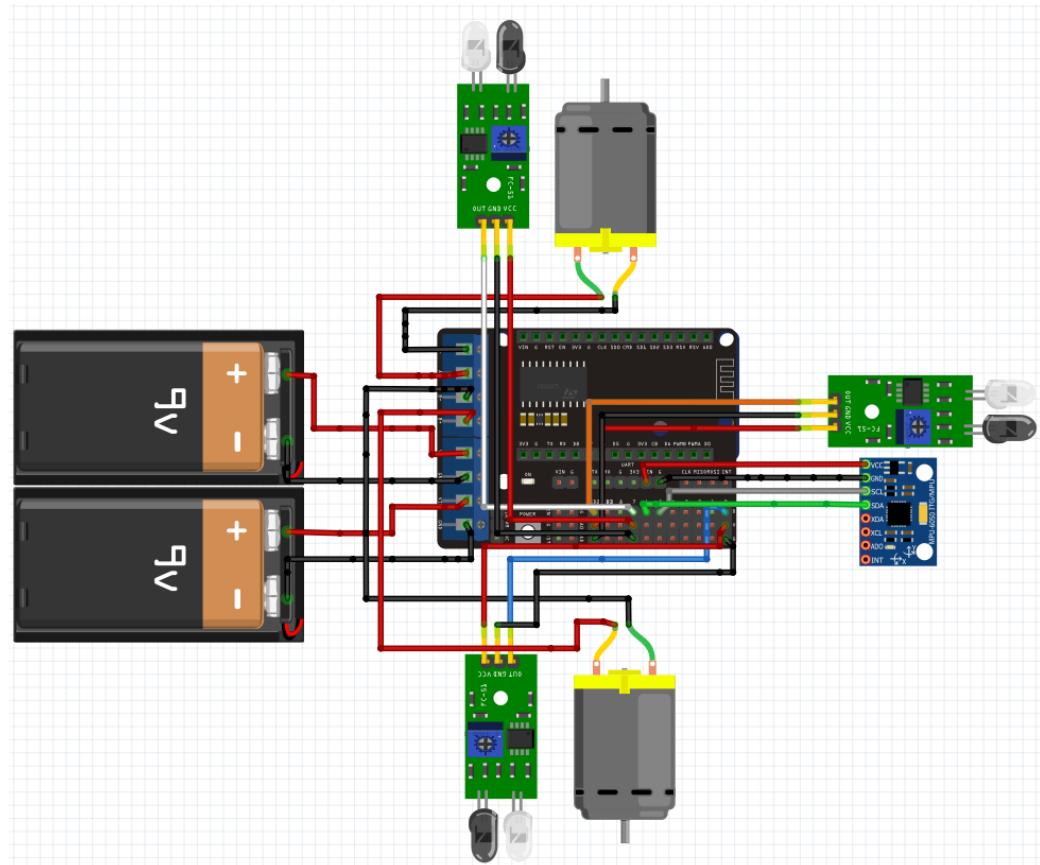
const uint8_t scl = D5;
const uint8_t sda = D6;
MPU6050 mpu6050(Wire);

void setup() {
    Serial.begin(115200);
    Wire.begin(sda, scl);
    mpu6050.begin();
    mpu6050.calcGyroOffsets(true);
}

void loop() {
    mpu6050.update();
    Serial.print("angleX : ");
    Serial.print(mpu6050.getAngleX());
    Serial.print("\tangleY : ");
    Serial.print(mpu6050.getAngleY());
    Serial.print("\tangleZ : ");
    Serial.println(mpu6050.getAngleZ());
}
```

Fonte: Elaborado pelo autor

Figura 38: Circuito quarto protótipo

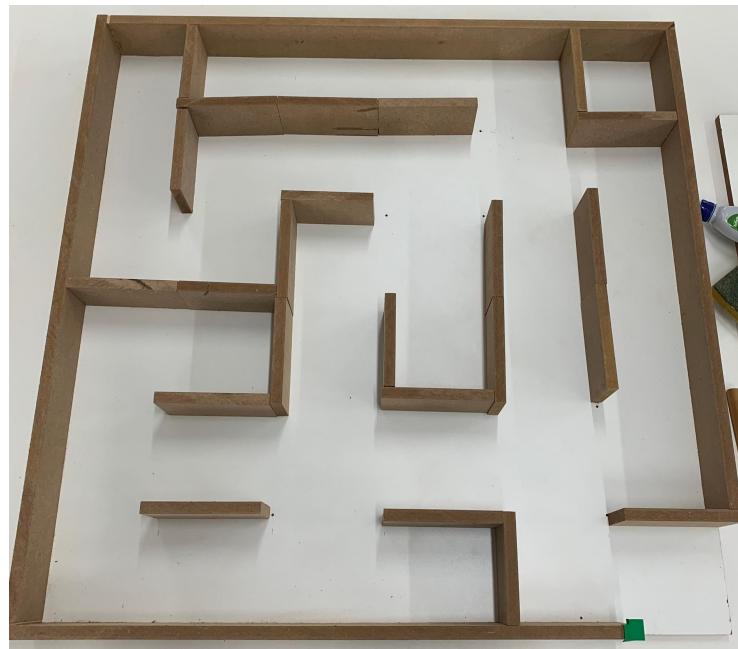


Fonte: Elaborado pelo autor

4. Resultados

Após a elaboração do protótipo final, foram realizados testes de bancadas a fim de validar o protótipo elaborado. No primeiro teste utilizou-se o cenário ilustrado na figura 39, consequentemente chegou-se a um resultado aceitável ilustrado na figura 40, no qual apresenta-se as duas matrizes fornecidas pelo robô, sendo uma o labirinto mapeado e a outra o caminho encontrado pelo algoritmo implementado.

Figura 39: Cenário de validação



Fonte: Elaborado pelo autor

Figura 40: Matrizes resultantes validação

Fonte: Elaborado pelo autor

Posteriormente, com a finalidade de ressaltar os resultados alcançados realizou-se outro teste com o cenário ilustrado na figura 41, obtendo assim as matrizes resultantes apresentadas na figura 42.

Figura 41: Cenário



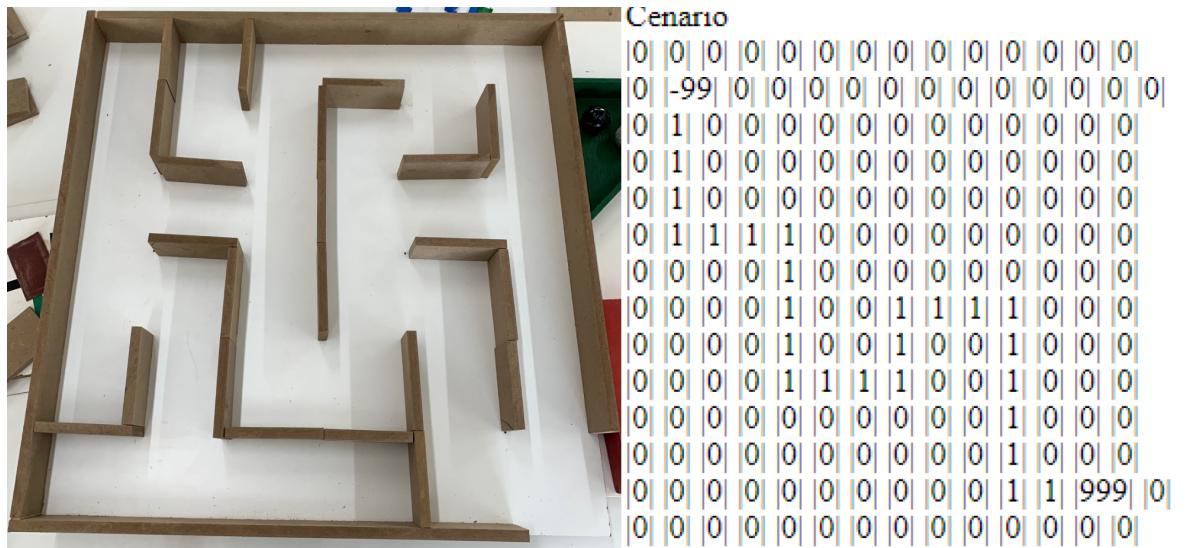
Fonte: Elaborado pelo autor

Figura 42: Matrizes resultantes

Fonte: Elaborado pelo autor

Também foi elaborado utilizando o cenário presente na figura 43, onde o protótipo gerou como resultado a matriz descrita na figura 44.

Figura 43: Cenário e matriz referente ao mapeamento



Fonte: Elaborado pelo autor

Figura 44: Caminho gerado

Caminho																									
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	5	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	10	0	0	17	18	19	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	11	0	0	0	16	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	12	13	14	15	0	0	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	26	26	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Fonte: Elaborado pelo autor

4.1. Documentação

No apêndice C, encontra-se o código fonte desenvolvido para aplicação do protótipo

final, contendo todas as informações necessárias para que a execução do programa desenvolvido seja perfeita.

5. CONCLUSÃO

Os objetivos apresentados no capítulo 1 deste trabalho foram atingidos com êxito, portanto, expondo dessa forma que é possível realizar o desenvolvimento de um protótipo autônomo capaz de mapear e solucionar labirintos utilizando tecnologia do ESP8266 e algoritmos de busca em largura.

A lógica utilizada para a obtenção dos valores dos sensores, locomoção dos motores, inserção de dados em matriz e algoritmo de busca em largura, mostrou-se eficiente, levando em consideração a baixa quantidade de erros aplicados a essa lógica e resultados satisfatórios de acordo com testes realizados.

Devido ao fato do protótipo final ser compacto e didático, pode ser usado para fins de aprendizado sobre algoritmos de busca aplicados em situações do cotidiano, tendo em vista que o protótipo realiza o mapeamento dos possíveis trajeto necessário para alcançar o destino e logo após procurar o melhor caminho.

5.1. Trabalhos futuros

O protótipo foi desenvolvido utilizando o microcontrolador ESP8266, sensor infravermelho HW-201, sensor giroscópio MPU6050, motor *shield*, motores e estrutura feita em impressoras 3D, dessa forma é possível mencionar continuações para o presente trabalho:

- Realizar o planejamento de uma estrutura visando melhoria de eficácia de determinados componentes, e aplicando diferentes elementos, objetivando uma melhor performance;
- Desenvolver um sistema para o mapeamento utilizando visão computacional;
- Realizar o desenvolvimento de um aplicativo que exibe o mapeamento do labirinto em tempo real e após apresente o melhor caminho;
- Implementar algoritmo Proporcional integral derivativo com base nos valores de ângulos obtidos para uma melhor locomoção entre as paredes;
- Implementar diferentes algoritmos de busca, visando validar a eficácia dos mesmos em um ambiente real.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] CINTRA, Thayller Vilela; ROLAND, Carlos Eduardo de França. **Problema do labirinto: estudo de caso de solução com robô Zumo Pololu 32U4.** Revista EduFatec: educação, tecnologia e gestão, Franca, v.1, n.2, p. 1-19, jul. /dez. 2018. Disponível em: <https://revistaedufatec.fatecfranca.edu.br/wp-content/uploads/2019/03/Thayller-Vilela-Cintra.pdf>. Acesso em: 22 maio
- [2] PACIEVITCH. THAIS PACIEVITCH. **Labirinto de Creta.** InfoEscola. Disponível em: <https://www.infoescola.com/mitologia-grega/labirinto-de-creta/>. Acesso em: 29 mar. 2023.
- [3] MATARIC, Maja J. **Introdução à Robótica.** São Paulo: Editora Unesp, 2014. Acesso em: 8 abril
- [4] OLIVEIRA, Sérgio de Oliveira. **Internet das Coisas com ESP8266, Arduino e Raspberry Pi.** 1^a edição. São Paulo: Novatec Editora, 2017. Acesso em: 15 abril
- [5] VALERE. VITOR VALERI. **O que é GPIO e o que ele pode fazer por você.** Disponível em <https://www.oficinadanet.com.br/hardware/40552-o-que-e-gpio>. Acesso em: 29 abril. 2023.
- [6] AQUINO. ADELMO ARTUR DE AQUINO. Sinal digital: **O modelo DRM em contraste ao sistema brasileiro de radiodifusão.** Disponível em <https://repositorio.ufersa.edu.br/handle/prefix/6033>. Acesso em: 5 maio. 2023
- [7] GONZAGA. ARTHUR SIMÕES. **Estudo de Técnica de Modulação por Largura de Pulso (PWM) Aplicado a Inversores Trifásicos.** Disponível em: https://bdm.unb.br/bitstream/10483/23636/1/2018_ArthurSimoesGonzaga_tcc.pdf. Acesso em: 14 maio . 2023.
- [8] FUTIDA, I. T., & ROMERO, R. A. F. (2008). **Desenvolvimento mecânico e de Controle PWM para sistema robótico.** Disponível em: <https://repositorio.usp.br/bitstreams/069f3045-0184-4dc0-af7b-888324b072ab>. Acesso em: 16 maio. 2023.
- [9] OLIVEIRA. GREICE OLIVEIRA. **NodeMCU – Uma plataforma com características singulares para o seu projeto IoT.** blogmasterwalkershop. Disponível em: <https://blogmasterwalkershop.com.br/embarcados/nodemcu/nodemcu-uma-plataforma-com-caracteristicas-singulares-para-o-seu-projeto-iot>. Acesso em: 20 maio. 2023.
- [10] GRAÇA, Pedro Cannavale. **Sistema de aquisição de dados utilizando o módulo ESP8266 NodeMCU.** 2017. 42 f. Trabalho de conclusão de curso (Bacharelado - Engenharia Elétrica) - Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2017. Disponível em: <<http://hdl.handle.net/11449/156909>>. Acesso em: 24 maio. 2023.
- [11] Handson Technology. **ESP8266 NodeMCU WIFI Development Board User Manual V1.3**. Online, Disponivel em: <https://handsontec.com/datasheets/module/esp8266-V13.pdf> Acesso em: 31 maio. 2023.

- [12] GALÍCIO, Micaela Sanches. **ANALISADOR DE CONSUMO DE CORRENTE ELÉTRICA.** 2019. 36 f. TCC (Nível técnico) - Curso de Técnico em Informática, Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso do Sul, Jardim, 2019. Cap. 2. Acesso em: 02 junho. 2023.
- [13] ARDUINO, **Arduino Software.** Disponível em: <https://www.arduino.cc/en/software>. Acesso em 25 junho. 2023.
- [14] STEVAN; S. L; **Internet das Coisas: Fundamentos e aplicações em Arduíno e NodeMCU;** São Paulo: Saraiva Editora, 2018. Acesso em: 28 junho . 2023.
- [15] ARDUINO NITRO. **Detecting Objects Using the Infrared(IR) Obstacle Sensor [Online].** Disponível em: <https://arduinointro.com/articles/projects/detecting-objects-using-the-infrared-ir-obstacle-sensor>. Acesso em: 11 Agosto 2023.
- [16] OLIVEIRA, Euler Oliveira, **Como usar com Arduino – Sensor Infravermelho Reflexivo de Obstáculo,** Disponível em: <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-sensor-infravermelho-reflexivo-de-obstaculo> . Acesso em 14 Agosto 2023.
- [17] LONGARETTI, Dionatas. **Implementação de uma luva microcontrolada para a captura de gestos.** Orientador: Alessandro Gonçalves Girardi. 2015. 63 p. Trabalho de Conclusão de Curso (Bacharel em Engenharia Elétrica) - Universidade Federal do Pampa, Curso de Engenharia Elétrica, Alegrete, 2015. Acesso em 18 Agosto 2023.
- [18] PISCIOTTA, Alex. **Controle de um robô equilibrista de duas rodas utilizando lógica Fuzzy.** 2020. Acesso em 22 Agosto 2023.
- [19] INÁCIO, M. J. **Sensores e Atuadores.** Disponível em: https://d1wqtxts1xzle7.cloudfront.net/51886225/Sensores_e_Atuadores_1-libre.pdf?1487697885=&response-content-disposition=inline%3B+filename%3DSensores_e_Atuadores_Sensores_e_Atuadores.pdf&Expires=1689613044&Signature=SpEFs8-jG4gPupPHYAUX72kPiAHv u4FFCqh9U0AVFQfWwkqGTU3G1td3I0XXHAC1oxGefdZUhwufu-dLAdCEaVGbUYgnd GKzEQUITlhK7ItH91sQuEs3TxAK8l9ox8298wL2OPg2~ci-Gdk2a7h9ie50j1cDEuzWbkwp Va4mJaKBVnP3g6DnWzaAPLL6Pfe1nbyf5GW9YiWkoxFY9vpEvZ~jDnWUIoCYzOtounr 24Lr7EYrtp08LpOZCS7xohxtxFUYd8oFwd9y8MT2tJb3o~WbyCdSsNIpH2DMTj6HCk35 8hq3--eP8OKnEsbypDCS~YqWWLsBcu47OJVWzRg01A &Key-Pair-Id=APKAJLOHF5 GGSLRBV4ZA. Acesso em 27 Agosto. 2023.
- [20].CARRARA, Valdemir. Introdução à Robótica Industrial. INPE. Disponível em: <http://mtc-m21b.sid.inpe.br/col/sid.inpe.br/mtc-m21b/2015/08.25.14.16/doc/publicacao.pdf>. Acesso em 30 Agosto 2023.
- [21] Santos, Gabriel Almeida, and Leonardo Freire Pacheco. "**MOTORES ELÉTRICOS DE CORRENTE CONTÍNUA.**" Acesso em 01 Setembro 2023.
- [22] MATOS, N. M. R. **Análise do funcionamento de um servomotor de corrente**

alternada com ímãs permanentes. Disponível em:

<https://repositorio-aberto.up.pt/bitstream/10216/72684/1/000155343.pdf>. Acesso em: 05 Setembro. 2023.

[23] ANDRADE, D. O. & SOUZA, E. C. & ECHEGARAY, G. V. F. & SOUZA, I. C. M. & FILHO, M. C. L. M. **Análise da viabilidade econômica de motores elétricos em uma indústria de cobre.** Disponível em:

http://repositoriosenaiba.fieb.org.br/bitstream/fieb/1752/1/Theoprax_DANIEL%20OLIVEIRA%20DE%20ANDRADE%20et%20al.pdf. Acesso em: 05 Setembro. 2023.

[24] LEÃO, Felipe César Pinheiro. **Controle de um robô em um ambiente monitorado para resolução de labirintos.** 2017. Acesso em: 10 Setembro. 2023.

[25] Bhargava, Aditya. **Entendendo Algoritmos.** São Paulo. Novatec Editora ,2017 Acesso em: 15 Setembro. 2023.

Apéndice A - Código terceiro protótipo

Arquivo CódigoPrincipal.ino

```
#define D0 16 //
#define D1 5 //
#define D2 4 //
#define D3 0 //
#define D4 2 //
#define D5 14 //
#define D6 12 //
#define D7 13 //
#define D8 15
#define D9 3
#define D10 1
#define D12 10

//Wifi

#include <ESP8266WiFi.h>

// Configuração do WiFi
const char* ssid = "Nome da rede"; //VARIÁVEL QUE ARMAZENA O NOME DA REDE
SEM FIO EM QUE VAI CONECTAR
const char* password = "Senha"; //VARIÁVEL QUE ARMAZENA A SENHA DA REDE
SEM FIO EM QUE VAI CONECTAR

WiFiServer server(80);

int funcao = 0;

//sensores
int sensor1 = D7;//Esquerda azul/
int sensor2 = D12;//Frente/
int sensor3 = D0;//Direita branco
```

```
int valor1;
int valor2;
int valor3;

//MOTORES
const int pwmMotorA = D1;
const int pwmMotorB = D2;
const int dirMotorA = D3;
const int dirMotorB = D4;

int motorSpeed = 100;
int motorSpeedNegativo = -100;
int parar = 0;
```

```
//Matriz
int x = 1;
int y = 2;
#define tamanho 14
int a, b, indice=0,cont=0;
int origem[2],destino[2];

int escolher = 0;

//O = 0 N = 1 L=2 S=3
int ref = 3;
int caminho [tamanho][tamanho];
int aux [tamanho][tamanho];
int matriz [tamanho][tamanho];

int cenario [tamanho][tamanho] =
```

{


```

        caminho[x][y] = 0;
    }
}

Serial.println(""); //PULA UMA LINHA NA JANELA SERIAL
Serial.println(""); //PULA UMA LINHA NA JANELA SERIAL
Serial.print("Conectando a "); //ESCREVE O TEXTO NA SERIAL
Serial.print(ssid); //ESCREVE O NOME DA REDE NA SERIAL

WiFi.begin(ssid, password); //PASSA OS PARÂMETROS PARA A FUNÇÃO QUE VAI
FAZER A CONEXÃO COM A REDE SEM FIO

while (WiFi.status() != WL_CONNECTED) { //ENQUANTO STATUS FOR DIFERENTE
DE CONECTADO
    delay(500); //INTERVALO DE 500 MILISEGUNDOS
    Serial.print(".");
}

Serial.println(""); //PULA UMA LINHA NA JANELA SERIAL
Serial.print("Conectado a rede sem fio "); //ESCREVE O TEXTO NA SERIAL
Serial.println(ssid); //ESCREVE O NOME DA REDE NA SERIAL
server.begin(); //INICIA O SERVIDOR PARA RECEBER DADOS NA PORTA DEFINIDA
EM "WiFiServer server(porta);"
Serial.println("Servidor iniciado"); //ESCREVE O TEXTO NA SERIAL

Serial.print("IP para se conectar ao NodeMCU: "); //ESCREVE O TEXTO NA SERIAL
Serial.print("http://"); //ESCREVE O TEXTO NA SERIAL
Serial.println(WiFi.localIP()); //ESCREVE NA SERIAL O IP RECEBIDO DENTRO DA
REDE SEM FIO (O IP NESSA PRÁTICA É RECEBIDO DE FORMA AUTOMÁTICA)

yield();
}

void web()

```

```

{
    yield();
    WiFiClient client = server.available(); //VERIFICA SE ALGUM CLIENTE ESTÁ
    CONECTADO NO SERVIDOR

    String request = client.readStringUntil('\r'); //FAZ A LEITURA DA PRIMEIRA LINHA DA
    REQUISIÇÃO
    client.flush(); //AGUARDA ATÉ QUE TODOS OS DADOS DE SAÍDA SEJAM
    ENVIADOS AO CLIENTE

    client.println("HTTP/1.1 200 OK"); //ESCREVE PARA O CLIENTE A VERSÃO DO
    HTTP
    client.println("Content-Type: text/html"); //ESCREVE PARA O CLIENTE O TIPO DE
    CONTEÚDO(texto/html)
    client.println("");
    client.println("<!DOCTYPE HTML>"); //INFORMA AO NAVEGADOR A
    ESPECIFICAÇÃO DO HTML
    client.println("<html>"); //ABRE A TAG "html"

    client.println("Funcao:");
    client.print(funcao);
    client.println("<br>");
    client.println("sensor1:");
    client.print(valor1);
    client.println("<br>");
    client.println("sensor2:");
    client.print(valor2);
    client.println("<br>");
    client.println("sensor3:");
    client.print(valor3);
    client.println("<br>");
    client.println("referencia:");
    //O = 0 N = 1 L=2 S=3
    if(ref == 0 ){

```

```
client.print("Oeste");
}
else if(ref == 1 ){
    client.print("Norte");
}
else if(ref == 2 ){
    client.print("Leste");
}
else if(ref == 3 ){
    client.print("Sul");
}
client.println("<br>");
client.println("<br>");
client.print("cenario");
client.println("<br>");

for (int y = 0; y<tamanho;y++){
    for (int x = 0; x<tamanho;x++){
        client.print(" |");
        client.print(cenario[x][y]);
        client.print(" | ");
    }
    client.println("<br>");
}

client.println("<br>");
client.print("Caminho");
client.println("<br>");

for (int y = 0; y<tamanho;y++){
    for (int x = 0; x<tamanho;x++){
        client.print(" |");
        client.print(caminho[x][y]);
        client.print(" | ");
    }
}
```

```

        }

        client.println("<br>");

    }

client.println("<br>");
client.println("X:");
client.print(x);
client.println("Y:");
client.println(y);
client.println("<br>");

client.println("</center></h1>"); //ESCREVE "Ola cliente!" NA PÁGINA
client.println("<center><font size='5'></center>"); //ESCREVE "Seja bem vindo!" NA
PÁGINA
client.println("</html>"); //FECHA A TAG "html"
delay(1); //INTERVALO DE 1 MILISEGUNDO
Serial.println("Cliente desconectado"); //ESCREVE O TEXTO NA SERIAL
Serial.println(""); //PULA UMA LINHA NA JANELA SERIAL
yield();
}

void verificarXY(){

    if(cenario[x][y] == -99)
    {
        Serial.print("Nao adiciono valor");
    }
    else if(cenario[x][y] == 999)
    {
        Serial.print("Nao adiciono valor");
    }

    else if(cenario[x][y] == 0)
    {

```

```
    cenario[x][y]= 1;
}
else if(cenario[x][y] == 1)
{
    Serial.print("Nao adiciono valor");
}

}

void adicionarValorMatriz(){
//O = 0 N = 1 L=2 S=3
if(ref == 0){
    verificarXY();
    x--;
    verificarXY();
    x--;
    verificarXY();
}

else if(ref == 1){
    verificarXY();
    y--;
    verificarXY();
    y--;
    verificarXY();
}

else if(ref == 2){
    verificarXY();
    x++;
    verificarXY();
    x++;
    verificarXY();
}

else if(ref == 3){
    verificarXY();
    y++;
    verificarXY();
}
```

```
y++;  
verificarXY();  
}  
Serial.print("ref =");  
Serial.print(ref);  
}  
  
void frente(){  
  
Serial.println("Frente");  
analogWrite(pwmMotorA, motorSpeed);  
digitalWrite(dirMotorA, LOW);  
analogWrite(pwmMotorB, motorSpeed+27);  
digitalWrite(dirMotorB, HIGH);  
delay(800);  
  
  
Serial.print("ref =");  
Serial.print(ref);  
funcao = 1;  
  
}  
  
void direita(){  
  
Serial.println("Direita");  
  
  
analogWrite(pwmMotorA, motorSpeed);  
digitalWrite(dirMotorA,LOW);  
analogWrite(pwmMotorB, motorSpeed);  
digitalWrite(dirMotorB, LOW);  
delay(1000);
```

```
analogWrite(pwmMotorA, parar);
digitalWrite(dirMotorA, LOW);
analogWrite(pwmMotorB, parar);
digitalWrite(dirMotorB, LOW);

analogWrite(pwmMotorA, motorSpeed );
digitalWrite(dirMotorA, LOW);
analogWrite(pwmMotorB, motorSpeed);
digitalWrite(dirMotorB, HIGH);
delay(200);

if(ref == 0)
    ref = 1;
else if(ref == 1)
    ref = 2;
else if(ref == 2)
    ref = 3;
else if(ref == 3)
    ref = 0;

funcao = 2;

}

void esquerda(){
Serial.println("Esquerda");

analogWrite(pwmMotorA, motorSpeed);
digitalWrite(dirMotorA, HIGH);
analogWrite(pwmMotorB, motorSpeed);
digitalWrite(dirMotorB, HIGH);
```

```
delay(1000);

analogWrite(pwmMotorA, parar);
digitalWrite(dirMotorA, LOW);
analogWrite(pwmMotorB, parar);
digitalWrite(dirMotorB, LOW);

analogWrite(pwmMotorA, motorSpeed );
digitalWrite(dirMotorA, LOW);
analogWrite(pwmMotorB, motorSpeed +27);
digitalWrite(dirMotorB, HIGH);
delay(200);

if(ref == 0)
    ref = 3;
else if(ref == 3)
    ref = 2;
else if(ref == 2)
    ref = 1;
else if(ref == 1)
    ref = 0;

funcao = 3;

}

void retorno(){
    analogWrite(pwmMotorA, motorSpeed);
    digitalWrite(dirMotorA, HIGH);
    analogWrite(pwmMotorB, motorSpeed);
    digitalWrite(dirMotorB, HIGH);
```

```

delay(2000);

Serial.print("sair");
analogWrite(pwmMotorA, parar);
digitalWrite(dirMotorA, LOW);
analogWrite(pwmMotorB, parar);
digitalWrite(dirMotorB, LOW);

analogWrite(pwmMotorA, motorSpeed);
digitalWrite(dirMotorA, LOW);
analogWrite(pwmMotorB, motorSpeed+27);
digitalWrite(dirMotorB, HIGH);
delay(1000);

//O = 0 N = 1 L=2 S=3

if(ref == 0)
    ref = 2;
else if(ref == 2)
    ref = 0;
else if(ref == 1)
    ref = 3;
else if(ref == 3)
    ref = 1;

}

void mapeamneto(){
yield();
int final_labirinto = 0;
Serial.println("dentro da map wile");
web();
while (final_labirinto == 0){

    analogWrite(pwmMotorA, parar );
    digitalWrite(dirMotorA, LOW);
    analogWrite(pwmMotorB,parar);
}

```

```

digitalWrite(dirMotorB, LOW);
delay(100);

Serial.println("");
Serial.print("X:");
Serial.println(x);
Serial.print("Y:");
Serial.print(y);
Serial.println("");



valor1 = digitalRead(sensor1);
Serial.print("sensor esquerda");
Serial.println(valor1);//Diminuir energia*/
valor2 = digitalRead(sensor2);
Serial.print("sensor frente");
Serial.println(valor2); //valor invertido;
valor3 = digitalRead(sensor3);
Serial.print("sensor direita");
Serial.println(valor3);

for (int y = 0; y<tamanho;y++){
    for (int x = 0; x<tamanho;x++){
        Serial.print(" |");
        Serial.print(cenario[x][y]);
        Serial.print("| ");
    }
    Serial.println(" ");
}
Serial.println(" ");

if(valor1 == 0 and valor2 == 1 and valor3 ==0){
    frente();
    adicionarValorMatriz();
}

```

```
}

else if(valor1 == 0 and valor2 == 0 and valor3 ==1){
    direita();
    adicionarValorMatriz();
}

else if(valor1 == 1 and valor2 == 0 and valor3 ==0){
    esquerda();
    adicionarValorMatriz();
}

else if(valor1 == 0 and valor2 == 0 and valor3 ==0){
    retorno();
}

else if(valor1 == 0 and valor2 == 1 and valor3 ==1){
    direita();
    adicionarValorMatriz();
}

else if(valor1 == 1 and valor2 == 0 and valor3 ==1){
    direita();
    adicionarValorMatriz();
}

else if(valor1 == 1 and valor2 == 1 and valor3 ==0){
    esquerda();
    adicionarValorMatriz();
}

else if(valor1 == 1 and valor2 == 1 and valor3 ==1){
    delay(1000);
}

if(x >= 12 and y >= 12){
    final_labirinto = 1;

}
```

```
web();
delay(500);

}

}

void busca()
{

Serial.print("O cenario pra busca");
for (int y = 0; y<tamanho;y++){
    for (int x = 0; x<tamanho;x++){
        Serial.print(" |");
        Serial.print(cenario[x][y]);
        Serial.print("| ");
    }
    Serial.println(" ");
}
Serial.println(" ");

web();
yield();

for(int x=0;x<tamanho;x++)
{
    yield();
    for(int y=0;y<tamanho;y++)
    {
        yield();
        if(cenario[x][y] == -99)
        {
            origem[0] = x;
            origem[1] = y;
            matriz[x][y]=0;
        }
    }
}
```

```

aux[x][y]=0;

}

if(cenario[x][y] == 999)
{
    destino[0] = x;
    destino[1] = y;
    matriz[x][y]=999;
    aux[x][y]=999;

}

}

if(cenario[x][y] == 1)
{
    matriz[x][y]=-60;//caminhos livres
    aux[x][y]=-60;
}

if(cenario[x][y] == 0)
{
    matriz[x][y]=-30;//paredes ou desconhecidos
    aux[x][y]=-30;

}

}

}

}

//verificando valores inseridos
for (int y = 0; y<tamanho;y++){
    for (int x = 0; x<tamanho;x++){
        Serial.print(" |");
        Serial.print(matriz[x][y]);
        Serial.print("| ");
    }
    Serial.println(" ");
}

```

```
Serial.println(" ");

int achou=0;
while (achou==0){
    yield();

    for (int x=0;x<tamanho;x++)
    {
        for(int y=0;y<tamanho;y++)
        {

            if(aux[x][y]==indice)
            {
                cont++;
                Serial.println("CONT ++");
            }
        }
    }

    if (cont>=1)
    {
        for (y=0;y<tamanho;y++)
        {
            for(x=0;x<tamanho;x++)
            {
                if(aux[x][y]==indice)
                {

                    a=x;
                    b=y;
                    Serial.println("");
                    Serial.print("A:");
                    Serial.println(a);
                    Serial.print("B:");
                }
            }
        }
    }
}
```

```

Serial.print(b);
Serial.println("");
aux[x][y]=-99; // -2 na tabela auxiliar significa que foi lido o campo

if (aux[a+1][b]==-60) { matriz[a+1][b]=indice+1; aux[a+1][b]=indice+1; }
if (aux[a][b+1]==-60) { matriz[a][b+1]=indice+1; aux[a][b+1]=indice+1; }
if (aux[a-1][b]==-60) { matriz[a-1][b]=indice+1; aux[a-1][b]=indice+1; }
if (aux[a][b-1]==-60) { matriz[a][b-1]=indice+1; aux[a][b-1]=indice+1; }

else if(matriz[a+1][b]==999 || matriz[a][b+1]==999 || matriz[a-1][b]==999
|| matriz[a][b-1]==999 ) { Serial.println("achou = 1 ");achou=1;}
cont--;
}

}

}

if(cont==0)
{
indice++;
}
}

//Visualização da matriz gerada
for (int y = 0; y<tamanho;y++){
for (int x = 0; x<tamanho;x++){
Serial.print("|");
Serial.print(matriz[x][y]);
Serial.print("|");
}
Serial.println(" ");
}
Serial.println(" ");
delay(500);

```

```

caminho[origem[0]][origem[1]]=-10;
caminho[destino[0]][destino[1]]=indice;

a=origem[0];
b=origem[1];

int busca = 0;
while(busca<indice)
{ busca++;

    if (matriz[a-1][b]==busca) { caminho[a-1][b]=busca+1; a=a-1; b=b; } //1
    else if (matriz[a+1][b]==busca) { caminho[a+1][b]=busca+1; a=a+1; b=b; } //2
    else if (matriz[a][b+1]==busca) { caminho[a][b+1]=busca+1; a=a; b=b+1; } //3
    else if (matriz[a][b-1]==busca) { caminho[a][b-1]=busca+1; a=a; b=b-1; } //4
    //muldar 3 e 4 pra 2 e 3 altera a busca de baixo para lateral

}

caminho[origem[0]][origem[1]]=01;
//Visualização do caminho
for (int y = 0; y<tamanho;y++){
    for (int x = 0; x<tamanho;x++){
        Serial.print(" |");
        Serial.print(caminho[x][y]);
        Serial.print("| ");
    }
    Serial.println(" ");
}
Serial.println(" ");
delay(500);

web();
}

```

```
void loop()
{
    delay(100);
    web();
    mapeamneto();
    Serial.println("Começar busca");
    delay(1000);
    busca();
    web();

}
```

Apêndice B - Código para pagina web

Arquivo WIFI_ESP8266.ino

```
//Wifi
#include <ESP8266WiFi.h>

// Configuração do WiFi
const char* ssid = "Nome da rede";
const char* password = "Senha";

WiFiServer server(80);

void setup()
{
Serial.begin(115200);

Serial.println(""); //PULA UMA LINHA NA JANELA SERIAL
Serial.print("Conectando a "); //ESCREVE O TEXTO NA SERIAL
Serial.print(ssid); //ESCREVE O NOME DA REDE NA SERIAL

WiFi.begin(ssid, password); //PASSA OS PARÂMETROS PARA A FUNÇÃO QUE VAI FAZER A
CONEXÃO COM A REDE SEM FIO

while (WiFi.status() != WL_CONNECTED) { //ENQUANTO STATUS FOR DIFERENTE DE
CONECTADO
    delay(500); //INTERVALO DE 500 MILISEGUNDOS
    Serial.print(".");
}

Serial.println(""); //PULA UMA LINHA NA JANELA SERIAL
Serial.print("Conectado a rede sem fio "); //ESCREVE O TEXTO NA SERIAL
Serial.println(ssid); //ESCREVE O NOME DA REDE NA SERIAL
server.begin(); //INICIA O SERVIDOR PARA RECEBER DADOS NA PORTA DEFINIDA EM
"WiFiServer server(porta);"
yield();
Serial.println("Servidor iniciado"); //ESCREVE O TEXTO NA SERIAL

Serial.print("IP para se conectar ao NodeMCU: "); //ESCREVE O TEXTO NA SERIAL
Serial.print("http://"); //ESCREVE O TEXTO NA SERIAL
```

```

Serial.println(WiFi.localIP()); //ESCREVE NA SERIAL O IP RECEBIDO DENTRO DA REDE
SEM FIO (O IP NESSA PRÁTICA É RECEBIDO DE FORMA AUTOMÁTICA)

}

void loop(){

    WiFiClient client = server.available(); //VERIFICA SE ALGUM CLIENTE ESTÁ CONECTADO
    NO SERVIDOR
    if (!client) { //SE NÃO EXISTIR CLIENTE CONECTADO, FAZ
        return; //EXECUTA O PROCESSO ATÉ QUE ALGUM CLIENTE SE CONECTE AO
        SERVIDOR
    }
    Serial.println("Novo cliente se conectou!"); //ESCREVE O TEXTO NA SERIAL

    String request = client.readStringUntil('\r'); //FAZ A LEITURA DA PRIMEIRA LINHA DA
    REQUISIÇÃO
    Serial.println(request); //ESCREVE A REQUISIÇÃO NA SERIAL
    client.flush(); //AGUARDA ATÉ QUE TODOS OS DADOS DE SAÍDA SEJAM ENVIADOS AO
    CLIENTE

    client.println("HTTP/1.1 200 OK"); //ESCREVE PARA O CLIENTE A VERSÃO DO HTTP
    client.println("Content-Type: text/html"); //ESCREVE PARA O CLIENTE O TIPO DE
    CONTEÚDO(texto/html)
    client.println("");
    client.println("<!DOCTYPE HTML>"); //INFORMA AO NAVEGADOR A ESPECIFICAÇÃO
    DO HTML
    client.println("<html>"); //ABRE A TAG "html"

    client.println("Hello Word:");//ESCREVE NA TELA Hello Word

    client.println("</center></h1>"); //ESCREVE "Ola cliente!" NA PÁGINA
    client.println("<center><font size='5'></center>"); //ESCREVE "Seja bem vindo!" NA PÁGINA
    client.println("</html>"); //FECHA A TAG "html"

    yield();
}

```



Apéndice C - Código Final.

Arquivo CódigoPrincipal.ino

```
#define D0 16 //
#define D1 5 //
#define D2 4 //
#define D3 0 //
#define D4 2 //
#define D5 14 //
#define D6 12 //
#define D7 13 //
#define D8 15
#define D9 3
#define D10 1
#define D12 10

//Wifi

#include <ESP8266WiFi.h>

// Configuração do WiFi
const char* ssid = "Nome da rede"; //VARIÁVEL QUE ARMAZENA O NOME DA REDE SEM
FIO EM QUE VAI CONECTAR
const char* password = "Senha"; //VARIÁVEL QUE ARMAZENA A SENHA DA REDE SEM FIO
EM QUE VAI CONECTAR

WiFiServer server(80);

int funcao = 0;

//sensores
int sensor1 = D7;//Esquerda azul/
int sensor2 = D12;//Frente/
int sensor3 = D0;//Direita branco

int valor1;
```

```

int valor2;
int valor3;

//MOTORES
const int pwmMotorA = D1;
const int pwmMotorB = D2;
const int dirMotorA = D3;
const int dirMotorB = D4;

int motorSpeed = 100;
int motorSpeedNegativo = -100;
int parar = 0;

#include <MPU6050_tockn.h>
#include <Wire.h>
const uint8_t scl = D5;
const uint8_t sda = D6;
MPU6050 mpu6050(Wire);

//Matriz
int x = 1;
int y = 2;
#define tamanho 14
int a, b, indice=0,cont=0;
int origem[2],destino[2];

int escolher = 0;

//O = 0 N = 1 L=2 S=3
int ref = 3;
int caminho [tamanho][tamanho];
int aux [tamanho][tamanho];
int matriz [tamanho][tamanho];

int cenario [tamanho][tamanho] =
{
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
}

```

```
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0,0,0,0,0}
```

```
};
```

```
void setup()  
{  
    Serial.begin(115200);  
  
    scenario[1][1] = -99;  
    scenario[13][12] = 999;  
  
    pinMode(pwmMotorA , OUTPUT);  
    pinMode(pwmMotorB, OUTPUT);  
    pinMode(dirMotorA, OUTPUT);  
    pinMode(dirMotorB, OUTPUT);  
    pinMode(sensor1, INPUT);  
    pinMode(sensor2, INPUT);  
    pinMode(sensor3, INPUT);  
  
    for (int x = 0; x < tamanho;x++) {  
        for (int y = 0; y < tamanho;y++) {  
            aux[x][y] = -20;  
            caminho[x][y] = 00;  
        }  
    }
```

```

}

Wire.begin(sda, scl);
mpu6050.begin();
mpu6050.calcGyroOffsets(true);

mpu6050.update();
Serial.print("angleX : ");
Serial.print(mpu6050.getAngleX());
Serial.print("\tangleY : ");
Serial.print(mpu6050.getAngleY());
Serial.print("\tangleZ : ");
Serial.println(mpu6050.getAngleZ());

Serial.println(""); //PULA UMA LINHA NA JANELA SERIAL
Serial.println(""); //PULA UMA LINHA NA JANELA SERIAL
Serial.print("Conectando a "); //ESCREVE O TEXTO NA SERIAL
Serial.print(ssid); //ESCREVE O NOME DA REDE NA SERIAL

WiFi.begin(ssid, password); //PASSA OS PARÂMETROS PARA A FUNÇÃO QUE VAI FAZER A CONEXÃO COM A REDE SEM FIO

while (WiFi.status() != WL_CONNECTED) { //ENQUANTO STATUS FOR DIFERENTE DE CONECTADO
    delay(500); //INTERVALO DE 500 MILISEGUNDOS
    Serial.print(".");
    Serial.println("");
    Serial.print("Conectado a rede sem fio ");
    Serial.println(ssid);
    server.begin(); //INICIA O SERVIDOR PARA RECEBER DADOS NA PORTA DEFINIDA EM "WiFiServer server(porta);"
    yield();
    Serial.println("Servidor iniciado");
    Serial.print("IP para se conectar ao NodeMCU: ");
    Serial.print("http://");
    Serial.println(WiFi.localIP()); //ESCREVE NA SERIAL O IP RECEBIDO DENTRO DA REDE SEM FIO (O IP NESSA PRÁTICA É RECEBIDO DE FORMA AUTOMÁTICA)
}

```

```

    yield();
}

void web0
{
    yield();
    WiFiClient client = server.available(); //VERIFICA SE ALGUM CLIENTE ESTÁ CONECTADO
NO SERVIDOR

String request = client.readStringUntil('\r'); //FAZ A LEITURA DA PRIMEIRA LINHA DA
REQUISIÇÃO
client.flush(); //AGUARDA ATÉ QUE TODOS OS DADOS DE SAÍDA SEJAM ENVIADOS AO
CLIENTE

client.println("HTTP/1.1 200 OK"); //ESCREVE PARA O CLIENTE A VERSÃO DO HTTP
client.println("Content-Type: text/html"); //ESCREVE PARA O CLIENTE O TIPO DE
CONTEÚDO(texto/html)
client.println("");
client.println("<!DOCTYPE HTML>"); //INFORMA AO NAVEGADOR A ESPECIFICAÇÃO
DO HTML
client.println("<html>"); //ABRE A TAG "html"

client.println("Funcao:");
client.print(funcao);
client.println("<br>");
client.println("sensor1:");
client.print(valor1);
client.println("<br>");
client.println("sensor2:");
client.print(valor2);
client.println("<br>");
client.println("sensor3:");
client.print(valor3);
client.println("<br>");
client.println("referencia:");
//O = 0 N = 1 L=2 S=3
if(ref == 0 ){
    client.print("Oeste");
}

```

```
}

else if(ref == 1 ){

    client.print("Norte");

}

else if(ref == 2 ){

    client.print("Leste");

}

else if(ref == 3 ){

    client.print("Sul");

}

client.println("<br>");

client.println("<br>");

client.print("cenario");

client.println("<br>");

for (int y = 0; y<tamanho;y++){

    for (int x = 0; x<tamanho;x++){

        client.print(" | ");

        client.print(cenario[x][y]);



        client.print("| ");

    }

    client.println("<br>");

    client.print("Caminho");

    client.println("<br>");



    for (int y = 0; y<tamanho;y++){

        for (int x = 0; x<tamanho;x++){

            client.print(" | ");

            client.print(caminho[x][y]);



            client.print("| ");

        }

        client.println("<br>");

    }

    client.println("<br>");

    client.println("X:");

}
```

```

client.print(x);
client.println("Y:");
client.println(y);
client.println("<br>");

client.println("</center></h1>"); //ESCREVE "Ola cliente!" NA PÁGINA
client.println("<center><font size='5'></center>"); //ESCREVE "Seja bem vindo!" NA PÁGINA
client.println("</html>"); //FECHA A TAG "html"
delay(1); //INTERVALO DE 1 MILISEGUNDO
Serial.println("Cliente desconectado"); //ESCREVE O TEXTO NA SERIAL
Serial.println(""); //PULA UMA LINHA NA JANELA SERIAL
yield();
}

void verificarXY(){
    if(cenario[x][y] == -99)
    {
        Serial.print("Nao adiciono valor");
    }
    else if(cenario[x][y] == 999)
    {
        Serial.print("Nao adiciono valor");
    }

    else if(cenario[x][y] == 0)
    {
        cenario[x][y]= 1;
    }
    else if(cenario[x][y] == 1)
    {
        Serial.print("Nao adiciono valor");
    }
}

void adicionarValorMatriz(){
    //O = 0 N = 1 L=2 S=3
    if(ref == 0){
        verificarXY();
        x--;
    }
}

```

```

    verificarXY();
    x--;
    verificarXY();
}
else if(ref == 1){
    verificarXY();
    y--;
    verificarXY();
    y--;
    verificarXY();
}
else if(ref == 2){
    verificarXY();
    x++;
    verificarXY();
    x++;
    verificarXY();
}
else if(ref == 3){
    verificarXY();
    y++;
    verificarXY();
    y++;
    verificarXY();
}
Serial.print("ref =");
Serial.print(ref);
}

void frente(){
    Serial.println("Frente");
    analogWrite(pwmMotorA, motorSpeed);
    digitalWrite(dirMotorA, LOW);
    analogWrite(pwmMotorB, motorSpeed+27);
    digitalWrite(dirMotorB, HIGH);
    delay(800);

    Serial.print("ref =");
}

```

```
Serial.print(ref);
funcao = 1;

}

void direita(){

Serial.println("Direita");
mpu6050.update();

int direita = mpu6050.getAngleZ() - 90;

while (mpu6050.getAngleZ()>= direita){
mpu6050.update();

Serial.print("angleX : ");
Serial.print(mpu6050.getAngleX());
Serial.print("\tangleY : ");
Serial.print(mpu6050.getAngleY());
Serial.print("\tangleZ : ");
Serial.println(mpu6050.getAngleZ());

analogWrite(pwmMotorA, motorSpeed);
digitalWrite(dirMotorA,LOW);
analogWrite(pwmMotorB, motorSpeed);
digitalWrite(dirMotorB, LOW);

}

Serial.print("sair");
analogWrite(pwmMotorA, parar);
digitalWrite(dirMotorA, LOW);
analogWrite(pwmMotorB, parar);
digitalWrite(dirMotorB, LOW);

analogWrite(pwmMotorA, motorSpeed );
digitalWrite(dirMotorA, LOW);
analogWrite(pwmMotorB, motorSpeed);
```

```
digitalWrite(dirMotorB, HIGH);
delay(200);

if(ref == 0)
    ref = 1;
else if(ref == 1)
    ref = 2;
else if(ref == 2)
    ref = 3;
else if(ref == 3)
    ref = 0;

funcao = 2;

}

void esquerda(){

Serial.println("Esquerda");

mpu6050.update();

int esquerda = mpu6050.getAngleZ() + 90;

while (mpu6050.getAngleZ() <= esquerda){

mpu6050.update();
Serial.println(esquerda);
Serial.println(mpu6050.getAngleZ());

analogWrite(pwmMotorA, motorSpeed);
digitalWrite(dirMotorA, HIGH);
analogWrite(pwmMotorB, motorSpeed);
digitalWrite(dirMotorB, HIGH);
}
}
```

```
Serial.print("sair");
analogWrite(pwmMotorA, parar);
digitalWrite(dirMotorA, LOW);
analogWrite(pwmMotorB, parar);
digitalWrite(dirMotorB, LOW);

analogWrite(pwmMotorA, motorSpeed );
digitalWrite(dirMotorA, LOW);
analogWrite(pwmMotorB, motorSpeed +27);
digitalWrite(dirMotorB, HIGH);
delay(200);

if(ref == 0)
    ref = 3;
else if(ref == 3)
    ref = 2;
else if(ref == 2)
    ref = 1;
else if(ref == 1)
    ref = 0;

funcao = 3;

}

void retorno0{

mpu6050.update();
int retorno = mpu6050.getAngleZ() + 180;

while (mpu6050.getAngleZ()<= retorno){

mpu6050.update();
Serial.print("angleX : ");
Serial.print(mpu6050.getAngleX());
Serial.print("\tangleY : ");
Serial.print(mpu6050.getAngleY());
Serial.print("\tangleZ : ");
Serial.println(mpu6050.getAngleZ());
```

```

analogWrite(pwmMotorA, motorSpeed);
digitalWrite(dirMotorA, HIGH);
analogWrite(pwmMotorB, motorSpeed);
digitalWrite(dirMotorB, HIGH);

}

Serial.print("sair");
analogWrite(pwmMotorA, parar);
digitalWrite(dirMotorA, LOW);
analogWrite(pwmMotorB, parar);
digitalWrite(dirMotorB, LOW);

analogWrite(pwmMotorA, motorSpeed);
digitalWrite(dirMotorA, LOW);
analogWrite(pwmMotorB, motorSpeed+27);
digitalWrite(dirMotorB, HIGH);
delay(1000);

//O = 0 N = 1 L=2 S=3

if(ref == 0)
    ref = 2;
else if(ref == 2)
    ref = 0;
else if(ref == 1)
    ref = 3;
else if(ref == 3)
    ref = 1;

}

void mapeamneto(){
yield();
int final_labirinto = 0;
web();
while (final_labirinto == 0){

//O = 0 N = 1 L=2 S=3

```

```

Serial.println("");
analogWrite(pwmMotorA, 0 );
digitalWrite(dirMotorA, LOW);
analogWrite(pwmMotorB,0);
digitalWrite(dirMotorB, LOW);
delay(1000);

Serial.println("");
Serial.print("X:");
Serial.println(x);
Serial.print("Y:");
Serial.print(y);
Serial.println("");

valor1 = digitalRead(sensor1);
Serial.print("sensor esquerda");
Serial.println(valor1);//Diminuir energia*/
valor2 = digitalRead(sensor2);
Serial.print("sensor frente");
Serial.println(valor2); //valor invertido;
valor3 = digitalRead(sensor3);
Serial.print("sensor direita");
Serial.println(valor3);
for (int y = 0; y<tamanho;y++){
for (int x = 0; x<tamanho;x++){
Serial.print(" ");
Serial.print(cenario[x][y]);
Serial.print(" | ");
}
Serial.println(" ");
}
Serial.println(" ");

if(valor1 == 0 and valor2 == 1 and valor3 ==0){
// paredes(false,true,false);
frente();
adicionarValorMatriz();
}

```

```
}

else if(valor1 == 0 and valor2 == 0 and valor3 ==1){
// paredes(false,false,true);
direita();
adicionarValorMatriz();
}

else if(valor1 == 1 and valor2 == 0 and valor3 ==0){
// paredes(true,false,false);
esquerda();
adicionarValorMatriz();
}

else if(valor1 == 0 and valor2 == 0 and valor3 ==0){
retorno();
}

else if(valor1 == 0 and valor2 == 1 and valor3 ==1){
direita();
adicionarValorMatriz();
}

else if(valor1 == 1 and valor2 == 0 and valor3 ==1){
direita();
adicionarValorMatriz();
}

else if(valor1 == 1 and valor2 == 1 and valor3 ==0){

esquerda();
adicionarValorMatriz();

}

else if(valor1 == 1 and valor2 == 1 and valor3 ==1){
delay(1000);
}

if (x >= 12 and y >= 12){
final_labirinto = 1;
```

```

        }
        web();
        delay(1000);

    }

void busca()
{
    Serial.print("O cenario pra busca");
    for (int y = 0; y<tamanho;y++){
        for (int x = 0; x<tamanho;x++){
            Serial.print(" |");
            Serial.print(cenário[x][y]);
            Serial.print("| ");
        }
        Serial.println(" ");
    }
    Serial.println(" ");

    web();
    yield();
    for(int x=0;x<tamanho;x++)
    {
        yield();
        for(int y=0;y<tamanho;y++)
        {
            yield();
            if(cenário[x][y] == -99)
            {
                origem[0] = x;
                origem[1] = y;
                matriz[x][y]=0;
                aux[x][y]=0;
            }
            if(cenário[x][y] == 999)
            {

```

```

destino[0] = x;
destino[1] = y;
matriz[x][y]=999;
aux[x][y]=999;

}

if(cenario[x][y] == 1)
{
    matriz[x][y]=-60;
    aux[x][y]=-60;
}

if(cenario[x][y] == 0)
{
    matriz[x][y]=-30;
    aux[x][y]=-30;
}

}

}

}

Serial.print("A matriz pra busca");
for (int y = 0; y<tamanho;y++){
    for (int x = 0; x<tamanho;x++){
        Serial.print("|");
        Serial.print(matriz[x][y]);
        Serial.print("| ");
    }
    Serial.println(" ");
}
Serial.println(" ");

int achou=0;
while (achou==0){
    yield();
}

for (int x=0;x<tamanho;x++)
{
    for(int y=0;y<tamanho;y++)

```

```

{
    if(aux[x][y]==indice)
    {
        cont++;
        Serial.println("CONT ++");
    }
}

if (cont>=1)
{
    for (y=0;y<tamanho;y++)
    {
        for(x=0;x<tamanho;x++)
        {
            if(aux[x][y]==indice)
            {

                a=x;
                b=y;
                Serial.println("");
                Serial.print("A:");
                Serial.println(a);
                Serial.print("B:");
                Serial.print(b);
                Serial.println("");
                aux[x][y]=-99; // -2 na tabela auxiliar significa que foi lido o campo

                if (aux[a+1][b]==-60) { matriz[a+1][b]=indice+1; aux[a+1][b]=indice+1; }
                if (aux[a][b+1]==-60) { matriz[a][b+1]=indice+1; aux[a][b+1]=indice+1; }
                if (aux[a-1][b]==-60) { matriz[a-1][b]=indice+1; aux[a-1][b]=indice+1; }
                if (aux[a][b-1]==-60) { matriz[a][b-1]=indice+1; aux[a][b-1]=indice+1; }

                else if(matriz[a+1][b]==999 || matriz[a][b+1]==999 ||matriz[a-1][b]==999
                ||matriz[a][b-1]==999 ) { Serial.println("achou = 1 ");achou=1; }

                cont--;
            }
        }
    }
}

```

```

        }

    }

if(cont==0)
{
    indice++;
}

}

for (int y = 0; y<tamanho;y++){
    for (int x = 0; x<tamanho;x++){
        Serial.print("|");
        Serial.print(matriz[x][y]);
        Serial.print("|");
    }
    Serial.println(" ");
}

Serial.println(" ");
delay(500);

caminho[origem[0]][origem[1]]=-10;
caminho[destino[0]][destino[1]]=indice;

a=origem[0];
b=origem[1];

int busca = 0;
while(busca<indice)
{ busca++;

    if (matriz[a-1][b]==busca) { caminho[a-1][b]=busca+1; a=a-1; b=b; } //1
    else if (matriz[a+1][b]==busca) { caminho[a+1][b]=busca+1; a=a+1; b=b; } //2
    else if (matriz[a][b+1]==busca) { caminho[a][b+1]=busca+1; a=a; b=b+1; } //3
    else if (matriz[a][b-1]==busca) { caminho[a][b-1]=busca+1; a=a; b=b-1; } //4
    //muldar 3 e 4 pra 2 e 3 altera a busca de baixo para lateral
}

```

```
    }
    caminho[origem[0]][origem[1]]=01;
    for (int y = 0; y<tamanho;y++){
        for (int x = 0; x<tamanho;x++){
            Serial.print(" |");
            Serial.print(caminho[x][y]);
            Serial.print("| ");
        }
        Serial.println(" ");
        delay(500);
        web();
    }
    void loop()
{
    delay(100);
    mapeamneto();
    Serial.println(" Comecar busca");
    delay(1000);
    busca();
    web();
}
```