



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

RUAN FELIPE DE ALMEIDA SILVA

**DESENVOLVIMENTO DE UM PROTÓTIPO DE VEÍCULO AUTÔNOMO COM
LOCALIZAÇÃO EM UM AMBIENTE INDOOR**

QUIXADÁ

2022

RUAN FELIPE DE ALMEIDA SILVA

DESENVOLVIMENTO DE UM PROTÓTIPO DE VEÍCULO AUTÔNOMO COM
LOCALIZAÇÃO EM UM AMBIENTE INDOOR

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. André Ribeiro Braga

QUIXADÁ

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S583d Silva, Ruan Felipe de Almeida.
Desenvolvimento de um protótipo de veículo autônomo com localização em um ambiente indoor / Ruan Felipe de Almeida Silva. – 2022.
93 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Computação, Quixadá, 2022.
Orientação: Prof. Dr. André Ribeiro Braga.

1. Veículo Autônomo. 2. Filtro de Kalman. 3. Localização Espacial. 4. Robótica. I. Título.

CDD 621.39

RUAN FELIPE DE ALMEIDA SILVA

DESENVOLVIMENTO DE UM PROTÓTIPO DE VEÍCULO AUTÔNOMO COM
LOCALIZAÇÃO EM UM AMBIENTE INDOOR

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em: ___/___/___

BANCA EXAMINADORA

Prof. Dr. André Ribeiro Braga (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Antônio Joel Ramiro de Castro
Universidade Federal do Ceará (UFC)

Prof. Dr. Thiago Werlley Bandeira da Silva
Universidade Federal do Ceará (UFC)

À minha família, por sempre acreditar em mim e me apoiar em todos os momentos nessa caminhada. Aos meus amigos e colegas que participaram e me apoiaram nessa trajetória.

AGRADECIMENTOS

À Deus, pelo dom da vida e por me proporcionar a graça de chegar a este momento.

Aos meus pais, por todos os esforços e dedicações feitos por mim e por me apoiarem em todas as etapas da minha graduação.

Ao meu irmão, por toda a ajuda e colaboração na realização deste trabalho.

A toda minha família, que sempre confiou e acreditou na realização dos meus sonhos.

Ao meu orientador, professor Dr. André Braga, por toda dedicação, disposição, auxílio e orientação realizados a mim durante toda a execução deste trabalho.

À todos os meus colegas e amigos da graduação, por todas as experiências vividas, aprendizados e amizades construídas ao longo destes anos e todos os momentos compartilhados. Em especial quero agradecer aos meus amigos do grupo Resistência, Alan Nascimento, David Tavares, Gregório Neto, Jorge Lucas, Michael Douglas, Natan Nobre e Paulo Miranda, que estiveram comigo desde o início desta jornada, proporcionando inúmeros momentos ímpares que irei recordar para sempre em minha vida.

À todos os professores da Universidade Federal do Ceará, Campus Quixadá, que colaboraram no processo de minha formação profissional e pessoal, com muito empenho e profissionalismo, compartilhando seus conhecimentos e vivências.

À todos os funcionários da Universidade Federal do Ceará, Campus Quixadá, que contribuíram de forma direta e indireta para a minha formação ao longo desta jornada.

À todos que fizeram parte desse ciclo vivido ao longo destes 5 anos e contribuíram de alguma forma na minha formação profissional e pessoal.

“Só se pode alcançar um grande êxito quando
nos mantemos fiéis a nós mesmos.”

(Friedrich Nietzsche)

RESUMO

A evolução tecnológica observada nos últimos anos em veículos e robôs móveis, permitiu a criação de diversos dispositivos automatizados que auxiliam as pessoas na execução de diferentes tarefas, como o transporte de cargas pesadas e a limpeza de cômodos. A automação de máquinas que conseguem se locomover e atuar de forma independente também permitiu benefícios na exploração submarina de regiões profundas e na pesquisa e investigação interplanetária. Um fator crítico para a navegação de veículos autônomos é a determinação de sua localização. O conhecimento de uma estimativa da posição e orientação do dispositivo, permitem uma maior segurança na tomada de decisão do controle realizado. Determinar a posição de um veículo é uma tarefa desafiadora, principalmente na exploração de lugares remotos ou de ambientes fechados, já que não se pode contar com o auxílio do GPS para esses casos. A partir disso, neste trabalho é proposto o desenvolvimento de um protótipo de veículo autônomo, que pode atuar de forma manual ou autônoma e que consegue navegar e determinar a sua posição em um ambiente fechado. O dispositivo desenvolvido pode ser controlado através de uma aplicação *mobile*, onde é possível enviar mensagens através do protocolo MQTT, para movimentar e parar o veículo e também alterar o seu modo de operação. Para estimar a posição do protótipo no ambiente, foi desenvolvido um modelo matemático para a movimentação do veículo e este foi aplicado a um Filtro de Kalman Estendido, para utilizar as medidas de distância, aceleração e velocidade angular provenientes dos sensores na determinação da localização do veículo. Testes na execução dos diferentes modos de operação e na determinação da posição em trajetórias distintas foram realizados, para observar tanto o funcionamento dos modos, como a posição do veículo ao longo do tempo, perante a execução de um movimento. Com os resultados obtidos, foi possível observar que a identificação da localização, aconteceu com alguns erros nas trajetórias em determinados casos, mas que os valores de posição final após o movimento estiveram bem próximos dos valores reais coletados. Já os testes nos modos de execução, detectaram uma resposta satisfatória do dispositivo na execução do comando após o envio da mensagem.

Palavras-chave: Veículo Autônomo. Filtro de Kalman. Localização Espacial. Robótica.

ABSTRACT

The technological evolution observed in recent years in vehicles and mobile robots has allowed the creation of several automated devices that help people to perform different tasks, such as transporting heavy loads and cleaning rooms. The automation of machines that can move around and act independently also allowed benefits in underwater exploration of deep regions and in interplanetary research and investigation. A critical factor for the navigation of autonomous vehicles is the determination of their location. The knowledge of an estimate of the position and the orientation of the device, allows greater security in the decision making of the performed control. Determine the position of a vehicle is a challenging task, especially when exploring remote places or closed environments, since GPS aid cannot be used on for these cases. From this, this work proposes the development of an autonomous vehicle prototype, which can act manually or autonomously and can navigate as well as determine its position in a closed environment. The developed device can be controlled through a mobile application, where it is possible to send messages through the MQTT protocol, to move and stop the vehicle and also change its operation mode. To estimate the position of the prototype in the environment, a mathematical model was developed for the vehicle movement and this was applied to an Extended Kalman Filter, to use the measurements of distance, acceleration and angular velocity from the sensors to determine the location of the vehicle. Tests in the execution of the different operation modes and in the position determination in different trajectories were carried out, to observe both the functioning of the modes and the position of the vehicle over time, during the movement executing. With the obtained results, it was possible to observe that the identification of the location happened with some errors in the trajectories in certain cases, but that the final position values after the movement were very close to the real values collected. The tests in execution modes detected a satisfactory response from the device in executing the command after sending the message.

Keywords: Autonomous Vehicle. Kalman Filter. Spatial Location. Robotics.

LISTA DE FIGURAS

Figura 1 – Formatos de energia em um sensor.	24
Figura 2 – Componentes de um sensor ultrassônico.	26
Figura 3 – Funcionamento do Sensor Infravermelho.	28
Figura 4 – Sensor de distância VL53L0X.	29
Figura 5 – Diagrama de um acelerômetro comum.	30
Figura 6 – Fluxograma dos procedimentos.	44
Figura 7 – Fluxo de informações no sistema.	51
Figura 8 – Estrutura do veículo.	52
Figura 9 – Circuito de controle dos motores.	54
Figura 10 – Circuito de conexão com o MPU6050.	55
Figura 11 – Circuito de conexão com os sensores de distância.	56
Figura 12 – Circuito de identificação de estado.	56
Figura 13 – Protótipo do veículo autônomo.	58
Figura 14 – Opções do menu do aplicativo	60
Figura 15 – Tela do modo de operação manual	61
Figura 16 – Tela do modo de operação autônomo.	62
Figura 17 – Tela do modo de comandos.	63
Figura 18 – Modelo de posição do veículo	64
Figura 19 – Decomposição do vetor aceleração	65
Figura 20 – Resultados do experimento 1	73
Figura 21 – Resultados do experimento 2	75
Figura 22 – Resultados do experimento 3	76
Figura 23 – Resultados do experimento 8	77

LISTA DE TABELAS

Tabela 1 – Mensagens para alterar o modo de operação do veículo	58
Tabela 2 – Mensagens do modo de operação manual	59
Tabela 3 – Mensagens do modo de comandos	59

LISTA DE QUADROS

Quadro 1 – Comparação entre os trabalhos	43
Quadro 2 – Descrição dos Experimentos	72
Quadro 3 – Resultados dos Experimentos	78

LISTA DE CÓDIGOS-FONTE

Código-fonte 1 – Aquisição dos dados via protocolo MQTT	85
Código-fonte 2 – Determinação da posição com Filtro de Kalman	87

LISTA DE ABREVIATURAS E SIGLAS

MQTT	<i>Message Queue Telemetry Transport</i>
ABS	<i>Anti-lock Braking System</i>
GPS	<i>Global Positioning System</i>
IMU	<i>Inertial Measurement Unit</i>
KF	<i>Kalman Filter</i>
GPIO	<i>General Purpose Input/Output</i>
PWM	<i>Pulse-Width Modulation</i>
IoT	<i>Internet of Things</i>
UART	<i>Universal asynchronous receiver/transmitter</i>
I2C	<i>Inter-Integrated Circuit</i>
SPI	<i>Serial Peripheral Interface</i>
LED	<i>Light Emissor Diode</i>
MEMS	<i>Micro-Electro-Mechanical Systems</i>
PID	Proporcional Integral Derivativo
DC	<i>Direct Current</i>
EKF	<i>Extended Kalman Filter</i>
HTTP	<i>Hypertext Transfer Protocol</i>
LIDAR	<i>Light Detection And Ranging</i>
RTK GPS	<i>Real Time Kinematic GPS</i>
SLAM	<i>Simultaneous Localization And Mapping</i>
ROS	<i>Robot Operating System</i>
MRUV	Movimento Retilíneo Uniformemente Variado

LISTA DE SÍMBOLOS

mA	Corrente Elétrica
cm	Distância
m	Distância
nm	Espessura
kHz	Frequência
Hz	Frequência
$k\Omega$	Resistência elétrica
V	Tensão Elétrica

SUMÁRIO

1	INTRODUÇÃO	18
1.1	Objetivos	20
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	Veículo Autônomo	21
2.2	Placa de desenvolvimento ESP32	22
2.3	Sensores	23
2.3.1	<i>Sensores de Distância</i>	25
2.3.1.1	<i>Sensores Ultrassônicos</i>	25
2.3.1.2	<i>Sensores Infravermelhos</i>	27
2.3.1.3	<i>Sensor de Distância Laser</i>	28
2.3.2	<i>Acelerômetros</i>	29
2.3.3	<i>Giroscópios</i>	31
2.4	Atuadores	31
2.4.1	<i>Motores Elétricos</i>	32
2.5	Filtro de Kalman	33
2.6	Protocolo de Comunicação <i>Message Queue Telemetry Transport</i> (MQTT) 36	
3	TRABALHOS RELACIONADOS	38
3.1	<i>Research on EKF-Based Localization Method of Tracked Mobile Robot (QU et al., 2016)</i>	38
3.2	<i>Implementación de un robot de navegación autónoma terrestre para evasión de obstáculos usando el pixhawk (CHAMORRO; GONZÁLEZ, 2019)</i>	39
3.3	<i>Development of a Prototype of Autonomous Vehicle for Agriculture Applications (CHEMHENGCHAROEN et al., 2019)</i>	40
3.4	Desenvolvimento de um protótipo para veículos autônomos com localização e mapeamento simultâneos (VELOSO, 2019)	40
3.5	<i>Development and Implementation of an Autonomously Driven Vehicle Prototype (BILL; SHAHNASSER, 2019)</i>	41
3.6	Análise Comparativa	42
4	PROCEDIMENTOS METODOLÓGICOS	44

4.1	Selecionar os componentes	44
4.2	Realizar testes nos componentes	45
4.3	Montar o dispositivo	46
4.4	Implementar o controle do veículo	46
4.5	Desenvolver a aplicação mobile	47
4.6	Implementar o sistema de localização	48
4.7	Realizar experimentos das funções em vários cenários	48
4.8	Avaliar o desempenho do sistema nos experimentos	49
5	MATERIAIS E MÉTODOS	50
5.1	Modelo da proposta	50
5.2	O veículo autônomo	52
5.2.1	<i>Circuito de controle dos motores</i>	<i>53</i>
5.2.2	<i>Circuito de conexão com os sensores</i>	<i>54</i>
5.2.3	<i>Circuito de identificação de estado</i>	<i>55</i>
5.2.4	<i>Protótipo do veículo</i>	<i>57</i>
5.2.5	<i>Implementação do algoritmo</i>	<i>57</i>
5.3	A aplicação mobile	59
5.3.1	<i>Modo de operação manual</i>	<i>61</i>
5.3.2	<i>Modo de operação autônomo</i>	<i>62</i>
5.3.3	<i>Modo de comandos</i>	<i>62</i>
5.4	O modelo matemático de movimentação do veículo	63
5.4.1	<i>Modelo de predição do Filtro de Kalman Estendido</i>	<i>65</i>
5.4.2	<i>Modelo de atualização do Filtro de Kalman Estendido</i>	<i>67</i>
5.5	O sistema de localização	69
6	EXPERIMENTOS E RESULTADOS	71
6.1	Configuração do cenário	71
6.2	Definição da variação de tempo	71
6.3	Descrição dos experimentos	72
6.4	Resultados dos experimentos	73
6.4.1	<i>Experimento 1</i>	<i>73</i>
6.4.2	<i>Experimento 2</i>	<i>74</i>
6.4.3	<i>Experimento 3</i>	<i>75</i>

6.4.4	<i>Experimento 8</i>	77
6.5	Análise dos resultados	78
6.6	Experimentos dos comandos do <i>app</i>	79
7	CONCLUSÕES E TRABALHOS FUTUROS	80
7.1	Considerações Finais	80
7.2	Trabalhos Futuros	81
	REFERÊNCIAS	82
	APÊNDICE A – CÓDIGOS-FONTES DO SISTEMA DE LOCALIZAÇÃO	85

1 INTRODUÇÃO

Os avanços tecnológicos em inúmeras áreas da ciência provocaram uma evolução significativa nos aparelhos eletrônicos, processos de produção e financeiros e também na indústria veicular (RODRIGUES, 2017). O impacto positivo causado pela evolução da tecnologia permitiu uma melhoria em diversos aspectos dos veículos automotores como a questão do conforto e segurança dos passageiros e na eficiência do veículo (RODRIGUES, 2017).

Dentro deste contexto de evolução tecnológica, os componentes mecânicos no ramo automotivo estão sendo sucedidos cada vez mais por equipamentos mecatrônicos que oferecem funcionalidades adicionais, através da implantação de *softwares* embarcados nestes elementos, aumentando assim a autonomia entre os sistemas constituintes de um automóvel. (BUECHEL *et al.*, 2015).

Segundo Jung *et al.* (2005), os sistemas embarcados são caracterizados por possuírem finalidades bem específicas, funcionarem de forma autônoma e serem geralmente desenvolvidos para componentes de pequenas dimensões. Estes sistemas já são encontrados em inúmeros equipamentos do dia-a-dia das pessoas, incluindo alguns sistemas incorporados aos veículos, como, por exemplo, a injeção eletrônica e os sistemas de frenagem *Anti-lock Braking System* (ABS) (JUNG *et al.*, 2005).

Alguns sistemas embarcados desenvolvidos apresentam características de execução em tempo real. Esse aspecto está relacionado a capacidade de realizar tarefas que precisam ser executadas em uma quantidade de tempo específica, geralmente bastante curta (JUNG *et al.*, 2005). Algumas dessas tarefas que necessitam ser executadas em tempo real não podem ter o limite máximo de tempo de execução excedido, pois isto pode levar a consequências catastróficas no ambiente do sistema que está sendo controlado. Esses sistemas são conhecidos como sistemas críticos ou *hard real-time systems* e alguns exemplos são, o controle de uma planta nuclear, sistemas de controle de voo em uma aeronave e aplicações automotivas como em veículos terrestres autônomos inteligentes (BUTTAZZO, 2011).

A automação dos veículos terrestres é um campo de estudo que tem se tornado cada vez mais significativo (WEI, 2015). Segundo Bill e Shahnasser (2019), desenvolver veículos autônomos torna possível a resolução de diversos problemas existentes nas estradas, que precisam ser encarados pelos motoristas e os demais indivíduos no trânsito.

Além de ser utilizado em veículos particulares das pessoas, os sistemas embarcados desenvolvidos para controlar veículos autônomos podem ser aplicados em diversas áreas, como

aplicações agrícolas, veículos de construção, otimização de serviços de entrega e entre outros (RODRIGUES, 2017).

Os veículos autônomos também são essenciais para exploração e obtenção de informações em ambientes inacessíveis aos seres humanos, como regiões submarinas com grandes profundidade, exploração de minérios em regiões perigosas, áreas devastadas por catástrofes, regiões afetadas por radiação nociva ao ser humano e na coleta e investigação de dados de outros planetas (CORDEIRO *et al.*, 2013).

Um dos grandes desafios na área de automação veicular e de robôs móveis é descobrir a posição e orientação desses equipamentos no ambiente em que se encontram a medida que realizam suas ações (QU *et al.*, 2016). Esta é uma tarefa crucial para realizar a tomada de decisão do veículo, pois de acordo com a informação de localização adquirida ele pode determinar qual será a próxima tarefa a ser executada. No entanto, os principais sensores empregados na determinação da localização, como bússolas, o *Global Positioning System* (GPS) e unidades de medição inercial (*Inertial Measurement Unit* (IMU)) apresentam incertezas nos valores mensurados e informações ruidosas (COSTA, 2013).

O Filtro de Kalman (*Kalman Filter* (KF)) é uma técnica probabilística que considera a influência do ruído do processo e na medição dos sensores, para assim obter a estimativa ótima das variáveis de estado do sistema linear e dessa forma buscar resolver o problema de localização em robôs móveis (QU *et al.*, 2016).

Mediante a este contexto, neste trabalho é apresentado o desenvolvimento de um protótipo de um veículo autônomo que, por meio do processamento dos dados de sensores, consegue realizar a interpretação das informações do ambiente em que se encontra e assim determinar a sua localização através da implementação de um Filtro de Kalman. Além disso, o dispositivo consegue executar tarefas conforme os comandos enviados de um aplicativo *mobile* que podem controlar a trajetória que ele deve seguir ou permitir que o veículo se auto-controle, detectando e desviando dos obstáculos encontrados no seu caminho.

Com este trabalho espera-se contribuir para a comunidade de pesquisa e desenvolvimento na área de robôs inteligentes, através do desenvolvimento de um dispositivo capaz de determinar a sua localização em um ambiente conhecido e de futuramente realizar algumas das principais funcionalidades requeridas num veículo autônomo, como detectar obstáculos em um ambiente e realizar o desvio destes mesmos obstáculos.

1.1 Objetivos

O objetivo geral deste trabalho é desenvolver um protótipo de um veículo autônomo que consegue navegar, orientar e se localizar em um ambiente *indoor* controlado. Além disso, o veículo produzido é capaz de seguir por um percurso determinado previamente por comandos enviados de um aplicativo *mobile* e também encontrar os obstáculos que estejam no seu caminho. O presente trabalho apresenta os seguintes objetivos específicos:

- a) Projetar e construir o protótipo do dispositivo físico, utilizando-se de sensores e atuadores para realizar a percepção do ambiente e realizar o deslocamento do dispositivo.
- b) Implementar os procedimentos que realizam as funcionalidades do controle do veículo, como detecção de obstáculos e percorrer uma trajetória delimitada por comandos do usuário.
- c) Desenvolver uma aplicação *mobile* para enviar os comandos para o veículo.
- d) Realizar a implementação de um Filtro de Kalman para estimar a localização do robô em um ambiente *indoor*.
- e) Validar o protótipo desenvolvido através de testes realizados sobre as funcionalidades implementadas em diferentes cenários.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos mais importantes que são utilizados neste trabalho. Na Seção 2.1 é apresentado o conceito de Veículo Autônomo, listando suas principais características e funcionalidades. Posteriormente a Seção 2.2, apresenta detalhes sobre a placa de desenvolvimento ESP32, tais como seus componentes e principais aplicações. A Seção 2.3 aborda sucintamente o conceito de Sensores e especifica as principais categorias de sensores utilizados nesse projeto, sendo estes, os Sensores de Distância, Acelerômetros e Giroscópios. Cada um destes conceitos é relatado em uma subseção desta Seção 2.3. Na sequência, a Seção 2.4 descreve o conceito de Atuadores e identifica quais elementos deste tópico são empregados na constituição deste trabalho. A Seção 2.5 por sua vez esclarece sobre o método probabilístico empregado neste trabalho para identificar a localização do dispositivo. Já a seção 2.6 explica sobre o protocolo de comunicação MQTT.

2.1 Veículo Autônomo

O veículo autônomo pode ser definido como um veículo que possui um sistema de controle e orientação. Este é integrado de diversos sensores, atuadores e outros componentes elétricos e mecânicos que possibilitam a interpretação do ambiente ao seu redor em tempo real. Com essas características, o veículo é capaz de relacionar-se com os componentes do ambiente onde este está inserido, e assim realizar o deslocamento de um determinado local a um certo destino de forma totalmente segura e independente, ou seja, sem a intervenção humana (RODRIGUES, 2017).

A capacidade de autonomia por parte dos veículos, mesmo que de forma parcial, é uma realidade cada vez mais comum devido ao crescimento tecnológico nos sistemas de transporte automotivo (MOLINA, 2018). Este desenvolvimento poderá provocar uma série de melhorias no trânsito, beneficiando tanto a condutores, passageiros e pedestres, de modo que o sistema autônomo que controla o veículo pode garantir uma condução mais segura e eficiente (MOLINA, 2018). Além disso, os veículos autônomos são capazes de viabilizar a locomoção independente de pessoas que por algum motivo não podem dirigir um veículo, garantindo a estas, uma maior acessibilidade para oportunidades de emprego e educação (LITMAN, 2020).

Além das vantagens apresentadas anteriormente, a automação veicular também pode ser empregada para auxiliar os motoristas em inúmeros procedimentos essenciais na condução

do veículo, como: garantir uma distância segura entre os veículos, realizar ultrapassagens e mudanças de faixa de forma segura, estacionar o veículo em uma vaga, controlar automaticamente a velocidade de acordo com as circunstâncias do trânsito e entre outras (JUNG *et al.*, 2005).

Todas as ações listadas que são desempenhadas de forma automatizada, dependem do nível de automação de direção do veículo. Segundo Litman (2020) existem cinco níveis de automação quando se trata na execução de tarefas necessárias durante a condução de um veículo, sendo elas:

- 1 Assistência ao motorista: o veículo é totalmente controlado pelo motorista, no entanto podem existir alguns mecanismos que auxiliam à direção no veículo.
- 2 Automação parcial: o veículo possui algumas funções automatizadas, como, por exemplo, a aceleração, porém o motorista ainda é o elemento responsável por conduzir e realizar o monitoramento do veículo no ambiente.
- 3 Automação condicional: o sistema já consegue monitorar o ambiente, porém o motorista deve estar sempre pronto para assumir a direção quando solicitado.
- 4 Automação alta: nesse nível de automação o sistema computacional consegue controlar todo o veículo e realizar todas as tarefas fundamentais da condução, porém somente em algumas condições específicas. Portanto, o motorista pode optar por realizar o controle do veículo para as demais circunstâncias ao qual a automação não cumpre todos os requisitos.
- 5 Automação total: no nível mais alto de automação o veículo está apto a conduzir e controlar todas as ações em todos os cenários de forma automática, não havendo mais a necessidade da intervenção do motorista.

Neste trabalho, o veículo autônomo é o principal elemento da pesquisa, cuja algumas das funcionalidades básicas são desenvolvidas e implementadas em um protótipo. Este por sua vez, através da análise de dados de sensores, busca realizar a condução e o controle do veículo de forma automatizada em certas condições.

2.2 Placa de desenvolvimento ESP32

A ESP32 é uma placa de desenvolvimento expressiva na área de sistemas embarcados. Ela apresenta uma grande quantidade de periféricos, como 36 pinos gerais (*General Purpose Input/Output* (GPIO)) que podem ser utilizados para entrada e saída de dados, 16 canais *Pulse-Width Modulation* (PWM), sensores integrados ao chip dependendo da placa e entre outros componentes e interfaces de comunicação que possibilitam a ampla utilização deste componente

em projetos de diversas áreas (BABIUCH *et al.*, 2019).

Uma série de características da placa ESP32 fazem com que ela seja altamente empregada em projetos de sistemas embarcados, principalmente aqueles relacionados à *Internet of Things* (IoT) (MAIER *et al.*, 2017).

Dentre os principais aspectos da placa, destacam-se o baixo custo, o baixo consumo de energia que apresenta em média uma corrente de operação de apenas 80 mA, os recursos de conectividade por Wi-Fi, *Bluetooth* e *Universal asynchronous receiver/transmitter* (UART) integrados a própria placa, interfaces de comunicação *Inter-Integrated Circuit* (I2C) e *Serial Peripheral Interface* (SPI) para interligação com outras placas, sensores e outros equipamentos, circuitos DAC e ADC para conversão de dados entre o analógico e o digital e vice-versa. Além disso, ela pode ser programada em diversas linguagens de programação, como C, C++, Lua e entre outras (MAIER *et al.*, 2017).

Para realizar a programação na placa ESP32, podemos utilizar-se da plataforma de programação para o Arduino, o Arduino IDE. Nesta plataforma, um ambiente de desenvolvimento integrado possibilita a criação e a escrita dos programas na memória da placa de maneira simples. A Espressif, que é a empresa responsável pela placa ESP32 desenvolveu um *plugin* para o chip da placa ESP32 que instala o suporte para o desenvolvimento de microcontroladores ESP32 neste ambiente do Arduino IDE (BABIUCH *et al.*, 2019).

De acordo com Babiuch *et al.* (2019), uma segunda escolha para programação do microcontrolador da ESP32, seria utilizar o Espressif IoT Development Framework para desenvolvimento de aplicativos na ESP32 de forma nativa. Este *framework* é utilizado em projetos mais complexos e proporciona uma configuração básica para ESP, a compilação do código-fonte desenvolvido e o envio do *firmware* para a placa.

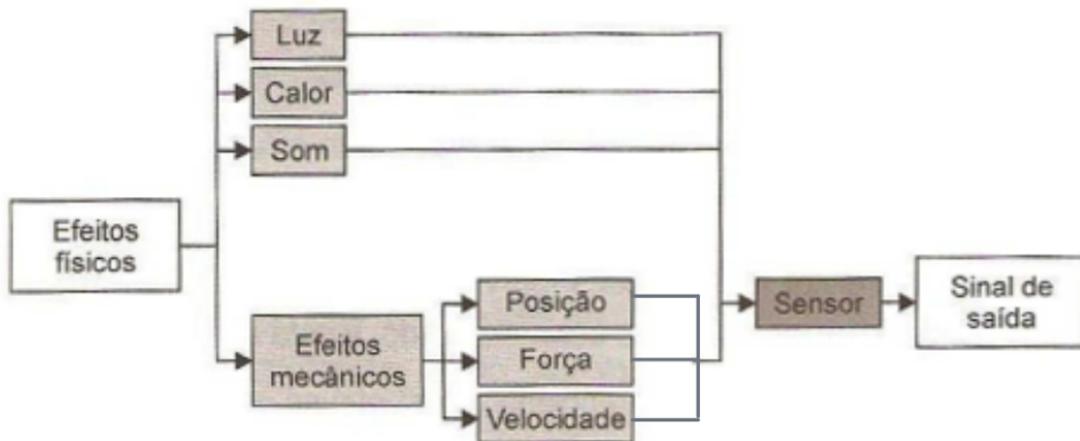
Para este trabalho, a placa de desenvolvimento ESP32 foi a escolhida para ser utilizada no processamento do sistema embarcado, realizando todo o controle dos componentes que constituem esse sistema, tais como sensores e atuadores. A escolha deste dispositivo se deu pelas diversas vantagens proporcionadas neste equipamento, como conectividade sem fio, baixo consumo de energia e grande quantidade de pinos de entrada e saída.

2.3 Sensores

Os sensores são dispositivos utilizados para realizar a detecção e a conversão de um determinado parâmetro físico, como temperatura, pressão, velocidade, aceleração, corrente

elétrica e entre outros, para outro formato de sinal, que normalmente é a forma elétrica (BHUYAN, 2010). A Figura 1 ilustra o fluxo básico da utilização de um sensor, com os sinais e fenômenos físicos sendo transformados em um novo tipo de sinal, após passarem por esse instrumento (THOMAZINI; ALBUQUERQUE, 2020).

Figura 1 – Formatos de energia em um sensor.



Fonte: Adaptado de Thomazini e Albuquerque (2020)

Um transdutor por sua vez é um componente que é empregado para transformar uma grandeza física em um sinal de tensão ou corrente elétrica, para que esta possa ser interpretada por algum sistema (BHUYAN, 2010). Comumente os termos “sensor” e “transdutor” são usados como sinônimos, apesar de que o sensor é um elemento mais simples que apenas realiza a transformação da característica física, enquanto que o transdutor possui um sensor e outros elementos que tornam o sinal propício a ser utilizado por um controlador (THOMAZINI; ALBUQUERQUE, 2020).

Os sensores ou transdutores podem ser classificados segundo os seus sinais de saída, podendo ser do tipo analógico ou digital. Os sensores analógicos são aqueles que podem gerar na saída, um sinal que assume qualquer valor dentro do intervalo de operação de valores possíveis desse dispositivo, ou seja, valores contínuos. (THOMAZINI; ALBUQUERQUE, 2020). Alguns exemplos de sensores analógicos, são aqueles utilizados para mensurar velocidade, temperatura, ângulo, distância e entre outras medidas.

Já os sensores do tipo digital, só podem exibir dois valores discretos na saída, ou seja, binários, que representam o “0”, nível baixo ou o “1”, nível alto. Alguns exemplos de aplicações onde esse tipo de sensor são empregados: a percepção da passagem de objetos em uma esteira

ou em outro equipamento, *encoders* que auxiliam para quantificar uma certa distância, controlar a velocidade de algum dispositivo, mensurar ângulos, etc. (THOMAZINI; ALBUQUERQUE, 2020).

Para realizar a interpretação e compreensão do ambiente, alguns sensores são aproveitados neste trabalho, para medir a distância até um certo obstáculo, controlar a velocidade do dispositivo e encontrar o ângulo correto em uma curva. Cada um desses sensores empregados na constituição do projeto são detalhados nas seções a seguir.

2.3.1 Sensores de Distância

Os sensores de distância ou de proximidade são um tipo de sensor capaz de mensurar a medida de distância entre um determinado ponto de referência e um objeto alvo. Esse tipo de sensor é geralmente empregado em aplicações como: medir as dimensões de um objeto, detectar obstáculos em robôs móveis e veículos autônomos; manter uma distância segura entre um veículo e outro componente da estrada; determinar trajetórias a serem seguidas (CHAMORRO; GONZÁLEZ, 2019).

Alguns modelos conhecidos de dispositivos medem a distância entre objetos por meio de ondas sonoras, como é o caso do sonar e do sensor ultrassônico e também através de raios de luz como o *laser*, e sensores infravermelhos (CHAMORRO; GONZÁLEZ, 2019). Em ambos esses casos, um pulso é emitida pelo sensor, que quando refletido no objeto, retorna até o próprio dispositivo, onde é capturado. A partir do tempo calculado entre a transmissão e a recepção da onda, a distância entre o sensor e o objeto é possível de ser calculada, utilizando-se da velocidade do meio utilizado, como a velocidade da luz, para sensores fotoelétricos e a velocidade do som para sensores ultrassônicos (FLEMING, 2008).

A seguir é apresentado um estudo comparativo dos principais tipos de sensores de distância, com o intuito de explanar sobre as vantagens e desvantagens de cada forma de mensuração de distância, e assim colaborar com a escolha desses dispositivos, utilizados neste trabalho para detectar e descobrir a distância até obstáculos próximos ao veículo.

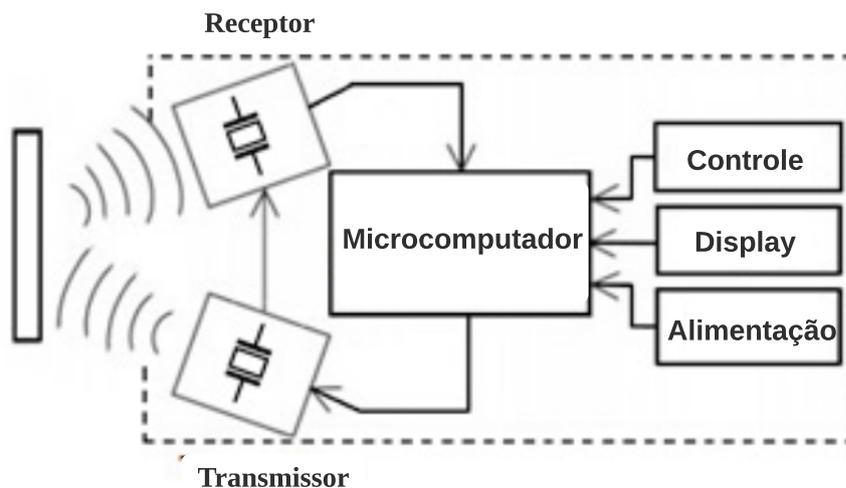
2.3.1.1 Sensores Ultrassônicos

Os sensores ultrassônicos são sensores que empregam ondas sonoras e os fundamentos de medir o tempo de propagação dessas ondas, para realizar a mensuração de distâncias, detectar mudanças de posição e efetuar a medição de fluxo de líquidos (KOVAL *et al.*, 2016).

Este tipo de sensor utiliza de ondas sonoras, conhecidas como ultrassom, e que não são perceptíveis à audição humana, pois as frequências dessas ondas são superiores à 20 kHz e as frequências audíveis ao ouvido humano estão no intervalo entre 20 Hz à 20 kHz (MORAIS *et al.*, 2017). Apesar disso, a propagação do ultrassom possui aspectos bastante semelhantes com à do som audível no ambiente, como a velocidade em que estas ondas são propagadas (KOVAL *et al.*, 2016).

Segundo Koval *et al.* (2016) a operação de um sensor ultrassônico se baseia no princípio de realizar a medição desde o tempo de envio de certos pulsos ultrassônicos, até o momento em que esses mesmos sinais são recebidos após serem refletidos em algum objeto. Para isso são necessários dois blocos básicos, um transmissor e um receptor. Ambos podem ser observados na Figura 2 que apresenta o diagrama de composição de um sensor ultrassônico.

Figura 2 – Componentes de um sensor ultrassônico.



Fonte: Adaptado de Koval *et al.* (2016)

O bloco transmissor, responsável por emitir o sinal ultrassônico, pode ser produzido através de transdutores magnetostritores. No entanto este tipo de transdutor possui uma limitação referente a faixa de frequência em que ele opera, podendo funcionar apenas em frequências com valores abaixo de 30 kHz. Para ser empregado em atividades que operam em outras faixas de frequências, o módulo transmissor deve ser fabricado com transdutores piezoelétricos. Estes últimos utilizam-se do efeito piezoelétrico inverso, que a partir de um sinal elétrico gera uma vibração mecânica que produz as ondas de ultrassom. Essas ondas então são propagadas e

refletidas em um determinado material, retornando ao sensor, onde são recebidas por um receptor ultrassônico que realiza a transformação dessas ondas mecânicas em sinais elétricos (KOVAL *et al.*, 2016).

Os sensores ultrassônicos são empregados em diversos trabalhos para identificar a distância entre dois corpos, no entanto tais dispositivos tendem a apresentar falhas no valor mensurado quando apontados para objetos inclinados, pois tal situação faz com que as ondas sonoras refletidas no material demorem mais tempo para serem recebidas no receptor, impactando diretamente no cálculo da distância ao material. Por esse motivo, tais sensores não foram empregados nesse trabalho, já que os valores de distância aos anteparos são essenciais para a tomada de decisão do veículo e para estimar a sua localização no Filtro de Kalman.

2.3.1.2 Sensores Infravermelhos

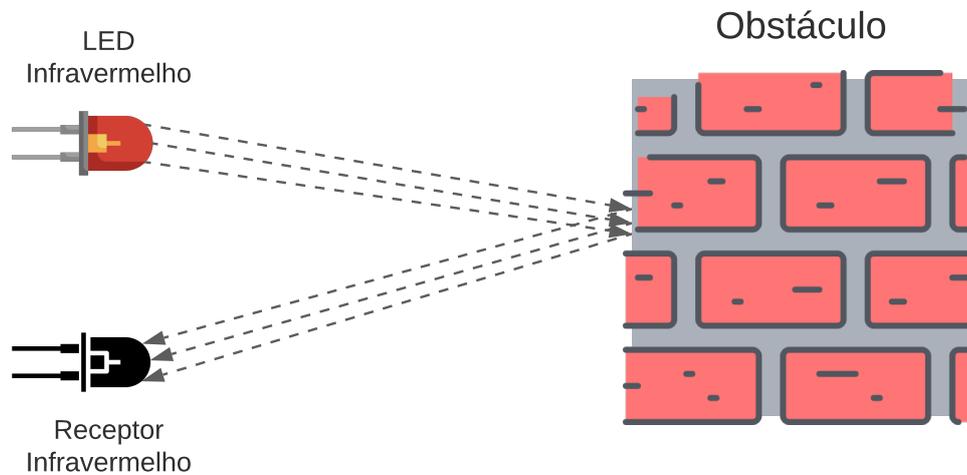
Os sensores infravermelhos são sensores utilizados para realizar a medição da distância entre o sensor e um obstáculo. Estes possuem um menor tempo de resposta, quando comparados aos sensores ultrassônicos. Neste tipo de sensor um pulso infravermelho é emitido e em seguida refletido em uma determinada superfície para que então seja retornado ao sensor. A partir da multiplicação entre a velocidade da luz, e o tempo para que o sinal seja recebido, é possível calcular a distância para um determinado objeto, no qual a luz foi refletida (MUSTAPHA *et al.*, 2013).

De acordo com Mustapha *et al.* (2013), os sensores infravermelhos possuem um diodo emissor de luz (*Light Emissor Diode (LED)*), que possui a função de irradiar a luz infravermelha. Esta luz é refletida em um objeto e retorna ao sensor para uma lente de foco. Esse componente direciona o feixe de luz para um detector sensível à posição, que se utiliza do efeito fotoelétrico, para transformar a energia da luz do feixe em energia elétrica.

O funcionamento mais simples dos sensores infravermelhos é ilustrado na Figura 3. Nessa Figura 3, o LED infravermelho emite os raios, que são refletidos no obstáculo e retornam ao receptor infravermelho, que realiza a captura desses sinais, para poderem ser processados posteriormente para que a distância ao obstáculo seja encontrada (SHINE *et al.*, 2018).

Os sensores infravermelhos são utilizados neste trabalho, para detectar obstáculos e anteparos próximos ao veículo. De acordo com os valores de distância obtidos, o robô pode escolher em seu modo autônomo qual ação deve ser tomada para evitar os obstáculos encontrados em seu caminho. Além disso, as distâncias para os anteparos do ambiente são empregadas no

Figura 3 – Funcionamento do Sensor Infravermelho.



Fonte: Elaborado pelo autor.

Filtro de Kalman para determinar sua posição no ambiente.

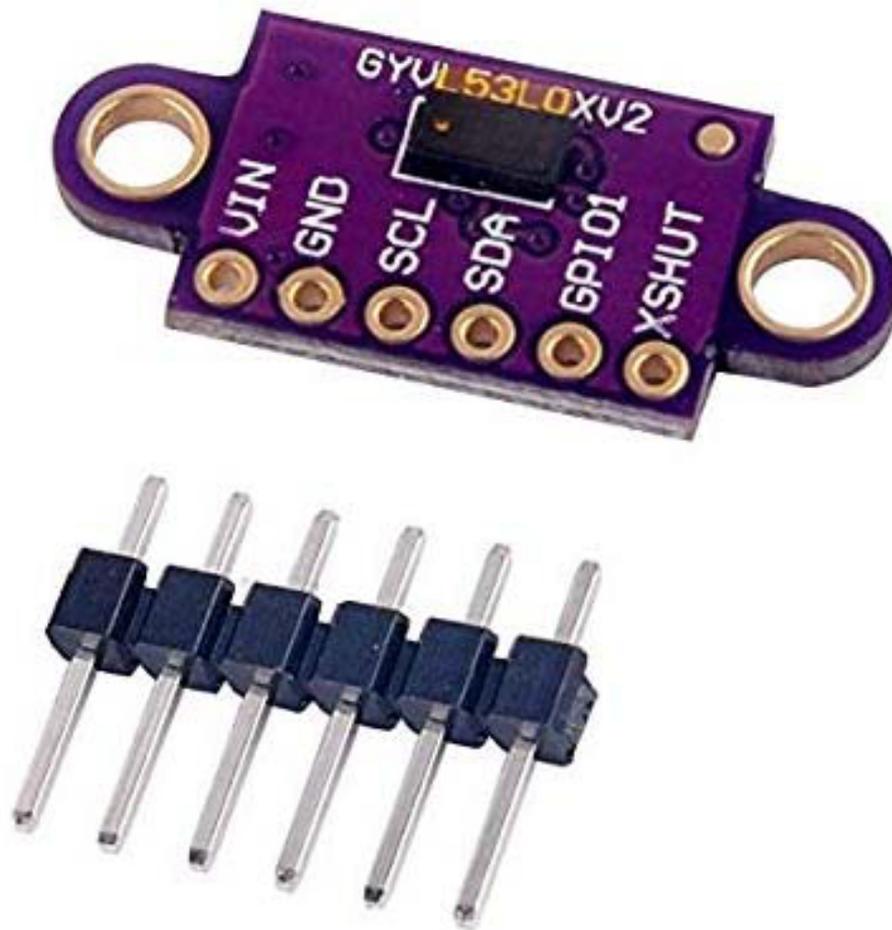
2.3.1.3 Sensor de Distância Laser

O módulo VL53LOX é um sensor de distância a laser que apresenta dimensões bem pequenas e uma precisão imensamente alta nas leituras realizadas (FAUZI *et al.*, 2018). Ele utiliza um laser infravermelho de 940 *nm* integrado a um microcontrolador para realizar a sua operação e assim detectar obstáculos que estejam até 2 metros de distância desse sensor. Para realizar o controle desse dispositivo é possível se utilizar de sua interface de comunicação I2C e dessa forma transferir os dados de distâncias obtidos para um outro componente (LAKOVIĆ *et al.*, 2019). A Figura 4 ilustra o sensor que está sendo explicado.

De acordo com Laković *et al.* (2019), de modo distinto aos sensores que medem distâncias com ondas ultrassônicas, o campo de visão desse sensor é bem mais estreito, com apenas 25°, aumentando dessa forma a probabilidade do sensor conseguir detectar um objeto que esteja à sua frente, sem que as ondas emitidas sejam refletidas em outros objetos próximos ao sensor. Outra vantagem desse sensor é que ele não apresenta problemas de “imagem dupla”, no qual não é possível identificar se o objeto detectado está muito perto ou muito distante do sensor.

O sensor VL53LOX possui uma fragilidade nos valores de leitura realizados quando o dispositivo encontra-se em condições de extrema intensidade luminosa (FAUZI *et al.*, 2018). Isso ocorre devido à tecnologia empregada para a medição da distância desse sensor, que observa o tempo entre o envio da luz (fótons) emitida pelo laser em direção ao alvo e a recepção dos

Figura 4 – Sensor de distância VL53L0X.



Fonte: Adaptado de Laković *et al.* (2019)

raios luminosos que são retornados ao sensor após serem refletidos nesse corpo (LAKOVIĆ *et al.*, 2019).

Neste trabalho, o módulo de distância VL53L0X é o sensor empregado para mensurar as distâncias aos obstáculos que estão próximos do protótipo do veículo desenvolvido, e assim auxiliar na lógica de controle do veículo e na localização deste dispositivo de acordo com os valores recebidos.

2.3.2 Acelerômetros

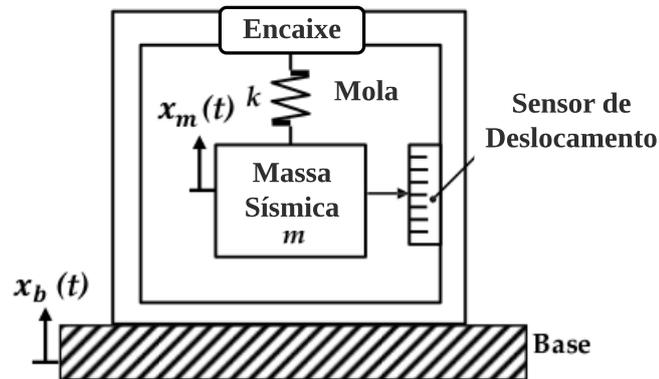
O acelerômetro é um dispositivo capaz de mensurar a aceleração realizada sobre um determinado corpo ou objeto, podendo este estar em repouso ou em deslocamento (MENDES *et al.*, 2018). A movimentação observada pelo acelerômetro pode ocorrer em qualquer um

dos três eixos, X, Y e Z, sendo Y o eixo vertical, X e Z os eixos horizontais onde X, indica movimentações para a direita e esquerda, enquanto que o eixo Z corresponde a deslocamentos para frente e para trás.

A Figura 5 apresenta a estrutura básica de um acelerômetro comum. No acelerômetro apresentado na Figura 5 os seguintes elementos estão presentes: uma mola, um sensor de deslocamento, que normalmente é um capacitor, acoplado a uma base b e uma massa sísmica m (LI *et al.*, 2018). De acordo com a Lei de Hooke e a Segunda Lei de Newton é possível estabelecer uma relação entre a aceleração do corpo a , o valor da massa m , o deslocamento dessa massa x_m , o deslocamento da base x_b e o coeficiente elástico k da mola. Esta relação é apresentada na Equação 2.1.

$$ma = k(x_m - x_b). \quad (2.1)$$

Figura 5 – Diagrama de um acelerômetro comum.



Fonte: Adaptado de Li *et al.* (2018)

Observando a Equação 2.1, nota-se que a aceleração medida nesse tipo de aparelho depende do coeficiente elástico da mola k e da massa sísmica m . Logo a escolha desses materiais impacta diretamente na atuação do dispositivo e de como a medição da aceleração é realizada (LI *et al.*, 2018).

Para este trabalho, a unidade de medição inercial MPU-6050, foi utilizada. Esta possui um acelerômetro de três eixos que permite determinar a aceleração do protótipo do veículo desenvolvido em relação aos eixos horizontais, no qual o veículo é deslocado durante o seu funcionamento.

2.3.3 Giroscópios

Segundo Forhan e Senna (2010), o giroscópio é um tipo de sensor que possibilita mensurar a velocidade de rotação de um determinado objeto ou corpo em torno de seu eixo. Em outras palavras, o giroscópio consegue indicar a velocidade angular de uma estrutura, ou seja, o deslocamento angular daquele objeto, utilizando como referência o próprio corpo do sensor.

Utilizando o giroscópio é possível descobrir a posição angular de um certo objeto, realizando a integração no domínio do tempo, do sinal de velocidade angular (FORHAN; SENNA, 2010). Logo, os giroscópios são elementos essenciais nos sistemas de navegação inercial, que são empregados em diversas aplicações, como sistemas aeroespaciais e militares utilizados em veículos aéreos, aquáticos e terrestres (SCHLISCHTING *et al.*, 2018).

De acordo com a aplicação em que o giroscópio vai ser utilizado, as tecnologias empregadas em sua constituição são alteradas, de forma que as aplicações mais robustas e críticas como os sistemas de foguetes e satélites utilizam giroscópios ressonantes e ópticos, enquanto que aqueles empregados em telefones celulares usam a tecnologia *Micro-Electro-Mechanical Systems* (MEMS), referentes a sistemas microeletromecânicos (SCHLISCHTING *et al.*, 2018).

Neste trabalho, o giroscópio é um componente que também se encontra na unidade de medição inercial MPU-6050, que será utilizado para determinar o ângulo da posição do veículo, auxiliando na orientação do dispositivo ao longo do percurso e também quando uma curva precisar ser executada.

2.4 Atuadores

Os atuadores são elementos apropriados para realizar a transformação de energia de um determinado tipo para outro distinto do primeiro, como, por exemplo, transformar energia elétrica em forma de movimento ou mudança de posição (CHAMORRO; GONZÁLEZ, 2019). Os atuadores podem ser categorizados de acordo com a energia que manipulam para efetuar o movimento, podendo ser pneumáticos, hidráulicos ou elétricos. Além disso, o movimento gerado por essas entidades pode ser linear, como para empurrar ou puxar um certo objeto, ou rotativo em que acontece uma movimentação correspondente a um giro ou rotação.

Segundo Thomazini e Albuquerque (2020) os atuadores são dispositivos que possuem a capacidade de transformar uma certa variável controlada. Eles são responsáveis por receber um determinado sinal provindo de um controlador e a partir deste realizar uma ação sobre o sistema

que está sendo controlado. Uma característica particular dos atuadores é que estes normalmente operam em condições de potência elevada.

De acordo com Bhuyan (2010) existe uma grande quantidade de atuadores, que variam em diversas características, como o tamanho e a quantidade de força gerada. Portanto, durante um projeto que envolve atuadores, os seguintes aspectos devem ser analisados para a escolha do atuador correto para aplicação desenvolvida:

1. O tipo de energia manipulada (elétrica, hidráulica ou pneumática).
2. A confiabilidade necessária.
3. A força mecânica empregada, se é torque ou impulso.
4. As funções de controle deste atuador, como um controlador Proporcional Integral Derivativo (PID) ou do tipo *on/off*, entre outras.
5. O custo do atuador.

De acordo com estas características, o atuador ideal pode ser escolhido para o projeto, de forma a suprir as necessidades para a aplicação desenvolvida (BHUYAN, 2010). Em Thomazini e Albuquerque (2020) são apresentados alguns exemplos de atuadores e seus respectivos tipos. São eles:

- Válvulas (pneumáticas, hidráulicas).
- Solenóides.
- Cilindros (pneumáticos, hidráulicos).
- Relés (eletromecânicos, estáticos).
- Motores (servomotores, eletromagnéticos).

Em um carro autônomo, os motores são elementos fundamentais e devido a esta importância, neste trabalho os motores eletromagnéticos ou motores elétricos, são explicados a seguir na Seção 2.4.1.

2.4.1 Motores Elétricos

Os motores elétricos possuem a função de converter a potência elétrica recebida na entrada em potência mecânica para a saída. Para isso, é necessário que um modelo dinâmico seja utilizado para representar a relação entre a entrada elétrica e o torque eletromagnético (BILGIN *et al.*, 2019).

Vários tipos de motores elétricos podem ser utilizados em aplicações de tração (BILGIN *et al.*, 2019). Segundo El-Refaie (2013), os requisitos necessários para escolha de

motores ou geradores elétricos empregados em aplicações de tração são bastante exigentes em relação aos seguintes aspectos: densidade de potência, custo e eficiência.

Os motores elétricos também podem ser empregados em uma variedade de sistemas, que necessitam de movimentação mecânica para o seu funcionamento, como é o caso de sistemas para aplicações automotivas, que controlam, por exemplo, os limpadores de para-brisa do veículo e também em sistemas de automação residencial para controlar algumas partes da casa, como janelas e portas (MAUSBACH; BEHRENS, 2017).

Nos motores elétricos, existem alguns tipos de perdas mecânicas que precisam ser consideradas sobre esse componente, como aquelas causadas pelo atrito com o ar. Além dessas, a corrente elétrica e o fluxo magnético acabam ocasionando respectivamente, perdas nos enrolamentos de cobre e nos núcleos magnéticos do motor pela variação do fluxo magnético nessas partes do núcleo. Estas últimas por sua vez são mais expressivas e podem acabar afetando a eficiência do motor (OKAMOTO *et al.*, 2016).

No presente trabalho, os motores elétricos, conhecidos como motores *Direct Current* (DC) ou motores de corrente contínua são controlados por sinais enviados pela placa ESP32 e utilizados para realizar a movimentação do veículo, mediante à transformação de energia elétrica em energia mecânica.

2.5 Filtro de Kalman

O Filtro de Kalman (KF) é um modelo matemático desenvolvido por Rudolph Kalman, cujo objetivo é empregar os valores mensurados ao longo do tempo de determinadas grandezas que possuem valores ruidosos e incertezas para obter resultados que buscam alcançar os valores reais dessas mesmas grandezas (THRUN *et al.*, 2006). Esta técnica realiza estimativas desses valores utilizando um controle realimentado, em que os estados do processo para um determinado instante no tempo dependem dos valores do estado anterior e dos valores ruidosos que foram mensurados (COSTA, 2013).

O KF é capaz de produzir boas suposições dos valores reais das grandezas físicas observadas, realizando a estimativa da incerteza do valor predito e em seguida calculando a média ponderada entre o valor que foi previsto e o valor mensurado. Com isso, o valor que possuir uma incerteza menor, receberá um peso maior no cálculo da média. Os valores produzidos por esse método tendem a se aproximar mais dos valores reais que as medidas mensuradas por um sensor, por exemplo, visto que a média ponderada calculada gera uma estimativa de incerteza melhor do

que os valores empregados no seu cálculo (CRUZ, 2013).

Thrun *et al.* (2006) explica que a técnica do Filtro de Kalman pode ser vista como um algoritmo que busca encontrar as inferências exatas sobre um sistema dinâmico linear, por meio de um modelo Bayesiano equivalente ao modelo de Markov. No entanto, no KF, o espaço de estados das variáveis que não são observadas é contínuo e todas as variáveis podem ser apresentadas através de uma distribuição normal.

O método do Filtro de Kalman pode ser subdividido em duas etapas, que delimitam as suas equações: predição e atualização. A etapa de predição tem como objetivo obter uma estimativa a priori do próximo passo no tempo, realizando uma projeção do estado atual e da estimativa da covariância à frente no tempo. Já a etapa de medição ou atualização é utilizada para realizar a “alimentação” do filtro, incorporando o novo valor da estimativa a priori para gerar uma estimativa a posteriori (CRUZ, 2013).

As Equações 2.2 e 2.3 representam como é calculado o estado e a covariância do estado na etapa de predição do KF, onde:

- $X_{k|k-1}$: estado X a priori no passo k .
- A : matriz de transição de estados.
- X_{k-1} : estado X a posteriori para o passo $k - 1$.
- $P_{k|k-1}$: covariância do estado a priori para o passo k .
- P_{k-1} : covariância do estado a posteriori para o passo $k - 1$.
- Q : covariância de transição de estado.
- F : matriz de mapeamento da covariância Q para o estado a posteriori X .

$$X_{k|k-1} = A \cdot X_{k-1} \quad (2.2)$$

$$P_{k|k-1} = A \cdot P_{k-1} \cdot A^T + F \cdot Q \cdot F^T \quad (2.3)$$

Já a etapa de atualização do estado, no KF pode ser descrita através das Equações 2.4, 2.5 e 2.6, onde:

- K_k : constante de Kalman no passo k .
- $P_{k|k-1}$: covariância do estado a priori para o passo k .
- H : matriz de mapeamento do estado real para o valor obtido pelo sensor.
- R : covariância do sensor.
- P_k : covariância do estado a posteriori para o passo k .

- $X_{k|k-1}$: estado X a priori no passo k .
- X_k : estado X a posteriori para o passo k .
- z_k : leitura do sensor para o passo k .

$$K_k = P_{k|k-1} \cdot H^T \cdot (H \cdot P_{k|k-1} \cdot H^T + R)^{-1} \quad (2.4)$$

$$P_k = P_{k|k-1} - K_k \cdot H \cdot P_{k|k-1} \quad (2.5)$$

$$X_k = X_{k|k-1} + K_k \cdot (z_k - H \cdot X_{k|k-1}) \quad (2.6)$$

O KF é um método matemático bastante eficaz, porém ele possui uma limitação de ter que ser utilizado em sistemas dinâmicos lineares. Porém, existe uma adaptação do Filtro de Kalman, conhecido como Filtro de Kalman Estendido (*Extended Kalman Filter* (EKF)) que pode ser empregado em sistemas cujo comportamento não é linear. Este algoritmo tem como objetivo, assim como o Filtro de Kalman, fornecer estimativas das grandezas físicas variáveis do sistema e apresenta etapas semelhantes às do KF tradicional, tornando assim este método adequado para executar uma fusão de sensores empregados em um projeto ou sistema (CRUZ, 2013).

A principal alteração que ocorre no EKF em relação ao KF, é que as matrizes de transições lineares no Filtro de Kalman são trocadas por generalizações não lineares dessas mesmas variáveis. Além disto, no EKF as matrizes de mapeamento de estados e covariância são substituídas pelas matrizes com os jacobianos das funções não lineares que realizam esse mapeamento nesse tipo de sistema (THRUN *et al.*, 2006).

Qu *et al.* (2016) afirma, que pelo algoritmo do Filtro de Kalman Estendido possuir uma ótima capacidade de diminuir a interferência causada pelos ruídos e conseguir se adaptar à estimativa ideal para sistemas onde não existe uma relação linear entre as medições e o modelo, este método passou a ser frequentemente empregado na determinação da localização de robôs móveis.

Neste trabalho, uma implementação do algoritmo do EKF será realizada para efetuar a fusão dos dados obtidos pelos sensores empregados no protótipo e assim encontrar uma estimativa da posição real do dispositivo.

2.6 Protocolo de Comunicação MQTT

O protocolo MQTT é um protocolo centralizado de comunicação da camada de aplicação desenvolvido para realizar a comunicação entre dispositivos que possuam uma quantidade de recursos limitada. Sua arquitetura é baseada num modelo de publicação-assinatura em tópicos, de modo que as mensagens publicadas por um cliente em um determinado tópico são entregues a todos os clientes que se inscreveram nesse tópico onde a mensagem foi publicada (THANGAVEL *et al.*, 2014).

De modo semelhante ao protocolo de aplicação *Hypertext Transfer Protocol* (HTTP), o MQTT também utiliza-se do protocolo de transporte TCP/IP nas camadas de rede subjacentes, porém como afirmado anteriormente, o MQTT foi planejado para dispositivos com restrições de recursos, por isso seu *payload* é bem menor que o do HTTP (THANGAVEL *et al.*, 2014).

Segundo Soni e Makwana (2017), o protocolo MQTT apresenta três níveis de qualidade do serviço, QoS0, QoS1 e QoS2, que estão relacionados à garantia da distribuição da mensagem que foi publicada. O primeiro nível (QoS0) não apresenta garantia de entrega da mensagem e tenta no máximo uma vez repassar a mensagem. Já no nível superior QoS1, a mensagem é enviada no mínimo uma vez e existe uma confirmação da recepção da mensagem enviada. Por fim no nível QoS2, a mensagem é entregue exatamente uma vez, havendo um *handshake* de quatro vias entre as partes envolvidas nesse processo, o que pode ocasionar um atraso na comunicação.

Na arquitetura do protocolo MQTT, existem duas entidades principais no processo de comunicação: o cliente e o *broker*. O cliente é um elemento que pode ser publicador ou assinante e deve sempre estabelecer uma conexão com o servidor (*broker*). Um cliente pode publicar mensagens, inscrever-se em um tópico para receber mensagens, cancelar sua inscrição de um tópico e se desconectar do servidor (SONI; MAKWANA, 2017).

O *broker* por sua vez é o componente responsável por controlar toda a distribuição de mensagens entre os clientes. Ele possui a função de receber as mensagens publicadas pelos clientes editores, e repassar estas mensagens para os clientes inscritos no tópico onde a mensagem foi publicada. Além disso, o *broker* também é encarregado de processar as solicitações dos clientes de assinatura e cancelamento nos tópicos disponíveis (SONI; MAKWANA, 2017).

Para este trabalho, o protocolo de aplicação MQTT vai ser utilizado em duas vertentes. O primeiro é para realizar a comunicação entre a aplicação *mobile* e a plataforma embarcada que controla o sistema de direção do veículo. O segundo é para publicar os valores mensurados pelos

sensores de distância, acelerômetro e giroscópio, para que a implementação do Filtro de Kalman possa se utilizar desses valores na localização do veículo.

3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados os trabalhos encontrados na literatura que se relacionam com este trabalho e que assim colaboraram para o desenvolvimento do presente trabalho descrito.

Foram analisados as principais tecnologias e equipamentos empregados em cada um dos trabalhos de acordo com o objetivo escolhido e como estas escolhas impactaram de forma positiva ou não nos resultados obtidos em cada um destes trabalhos de pesquisa.

No final do capítulo é realizada uma comparação entre os trabalhos descritos nesta seção e o trabalho proposto, de modo a averiguar as principais semelhanças e diferenças encontradas em relação às tecnologias e técnicas aplicadas em cada um dos estudos e as que são empregadas nessa proposta.

3.1 *Research on EKF-Based Localization Method of Tracked Mobile Robot (Qu et al., 2016)*

Em Qu *et al.* (2016) é apresentado uma solução para estimar a posição e o ângulo da direção de um robô móvel. Para isso, os dados de odometria são combinados com as medições de um sensor distância em um Filtro de Kalman Estendido, de modo a reduzir o erro encontrado nos valores da odometria.

No trabalho de Qu *et al.* (2016), foi desenvolvido um modelo cinemático para determinar a estimativa da localização do robô móvel, a partir das velocidades angulares das rodas do dispositivo, através de *encoders* acoplados nesses componentes. Em relação ao modelo de medição, foi utilizado um modelo baseado nos valores obtidos por um sensor a laser, que foi posto sobre um servo motor cujo ângulo de rotação vai de 0 a 180 graus, permitindo assim capturar as distâncias entre o robô e os obstáculos que estejam ao seu redor.

Os modelos desenvolvidos no trabalho de Qu *et al.* (2016), foram aplicados no método de localização de um EKF e foram realizados experimentos através de simulações no Matlab, e numa plataforma de um robô móvel. Os resultados dos experimentos demonstraram, que a posição e o ângulo do robô obtidos pelo EKF conseguem reduzir consideravelmente o erro acumulativo gerado na odometria tradicional.

No presente trabalho, também busca-se desenvolver um modelo para a movimentação do protótipo e um para a medição das distâncias, além de utilizar a técnica do EKF para determinar a estimativa da localização do veículo. Porém, diferentemente do trabalho de Qu *et al.* (2016), no

presente trabalho será utilizado os valores de aceleração e velocidade angular de uma IMU para determinar a movimentação do veículo ao longo do tempo. Além disso, no presente trabalho serão utilizados 3 sensores de distância à laser e os seus dados serão combinados para determinar as medições de distâncias e detectar os obstáculos nas proximidades do veículo. Outrossim, o presente trabalho também objetiva permitir o controle do veículo desenvolvido com uma aplicação *mobile*.

3.2 *Implementación de un robot de navegación autónoma terrestre para evasión de obstáculos usando el pixhawk (CHAMORRO; GONZÁLEZ, 2019)*

Em Chamorro e González (2019) é demonstrado um protótipo de um veículo autônomo terrestre, que contém um sistema de navegação e um para evasão de obstáculos que pode ser empregado em aplicações que necessitam do rastreamento da trajetória desse veículo. Para o desenvolvimento desse projeto foi utilizado uma placa Pixhawk mini, que realizava o controle de um chassi de 6 rodas utilizado como protótipo. Este por sua vez conseguia realizar uma determinada rota que era escolhida através do *software* de estação terrena QGroundControl. No trabalho também foram acoplados sensores de proximidade para tornar possível a detecção e o desvio de obstáculos pelo veículo inteligente.

Foram realizados testes para investigar o funcionamento do protótipo desenvolvido nas funções de controle manual e de navegação automática em um percurso definido com e sem obstáculos. Notou-se que devido a uma falha de *hardware* na unidade de medição inercial da Pixhawk mini que contém os sensores acelerômetro, giroscópio e magnetômetro, houve inúmeras leituras erradas realizadas por esses sensores da placa, o que conseqüentemente levou a um funcionamento incorreto do modo de navegação autônoma do dispositivo (CHAMORRO; GONZÁLEZ, 2019).

Neste trabalho o objetivo é realizar o controle do veículo de forma autônoma e manual, assim como o trabalho de Chamorro e González (2019). Porém no trabalho de Chamorro e González (2019), o veículo era controlado manualmente através da configuração de um rádio transmissor para envio dos sinais que agem sobre os atuadores (motores) do dispositivo e neste trabalho uma aplicação *mobile* foi desenvolvida para realizar o controle total do veículo, incluindo de qual modo ele deve operar (manual ou autônomo).

3.3 *Development of a Prototype of Autonomous Vehicle for Agriculture Applications* (CHEMHENGCHAROEN *et al.*, 2019)

O trabalho de Chemhengcharoen *et al.* (2019) apresenta o desenvolvimento de um protótipo de veículo autônomo, cujo objetivo é ser utilizado em aplicações do campo da agricultura, mais precisamente o veículo produzido foi empregado na tarefa de espalhar fertilizantes em plantações. Os autores utilizaram uma controladora Pixhawk e um Arduino para realizar o processamento dos dados obtidos pelos sensores e controlar os atuadores responsáveis pela movimentação do veículo e pelas ações relacionadas ao dispositivo agrícola espalhador de fertilizante. Para evitar do veículo colidir com alguma obstrução em sua rota, dois sensores *Light Detection And Ranging* (LIDAR) foram inseridos no protótipo para ser possível detectar e evitar estas barreiras.

Os autores do trabalho ainda realizaram um estudo comparativo entre o GPS convencional e o *Real Time Kinematic* GPS (RTK GPS), onde concluíram que o último apresenta uma precisão bem melhor em comparação com o primeiro e que, portanto é mais recomendado para missões agrícolas. Já no teste da navegação autônoma do dispositivo, um percurso em formato de S foi projetado através da marcação de pontos no *software* Mission Planner e como resultado o veículo distribuidor de fertilizantes agrícolas conseguiu um erro de posição inferior a 16 *cm* utilizando o RTK GPS com uma precisão de 1 metro.

Ambos os trabalhos, tanto o proposto, quanto o de Chemhengcharoen *et al.* (2019), possuem o objetivo de encontrar a posição do veículo em meio ao ambiente onde ele está atuando. Porém, o objetivo no trabalho proposto é implementar funções de um Filtro de Kalman para estimar a posição e orientação do veículo em um ambiente *indoor* a partir dos valores obtidos por seus sensores de distância e a unidade de medição inercial, de modo distinto a proposta de Chemhengcharoen *et al.* (2019) cuja aplicação é voltada para a agricultura e portanto em um ambiente aberto e por isso utiliza-se do GPS para indicar a localização do veículo.

3.4 *Desenvolvimento de um protótipo para veículos autônomos com localização e mapeamento simultâneos* (VELOSO, 2019)

Veloso (2019) retrata o desenvolvimento de um protótipo de um veículo autônomo, que consegue se adaptar e navegar em um ambiente com obstáculos, identificando estes através de sensores de distância. As medidas recebidas através dos sensores, também são utilizadas para

determinar a localização do veículo em tempo real, e realizar o mapeamento do ambiente em que o veículo se encontra, processo conhecido como *Simultaneous Localization And Mapping* (SLAM).

No desenvolvimento do projeto foram utilizados dois sensores ultrassônicos e um sensor infravermelho para realizar a detecção de obstáculos na trajetória do veículo. Além de uma unidade de medição inercial MPU-9250 para determinar a orientação do veículo em relação aos eixos. Para a plataforma de desenvolvimento e controle do veículo foi escolhido um Raspberry Pi 3, que realiza o processamento dos dados coletados pelos sensores, ativa os motores do veículo e envia os dados obtidos para um servidor, onde uma aplicação *web* consegue realizar a visualização destas informações e exibir o mapa que é criado a cada iteração do veículo com o ambiente (VELOSO, 2019).

O presente trabalho se assemelha com o de Veloso (2019) em relação ao objetivo de encontrar a localização do veículo no ambiente em que ele se encontra. No entanto, o projeto de Veloso (2019) não possui uma aplicação responsável por realizar o controle do veículo, como o que acontece no trabalho proposto. Além disso, o presente trabalho utiliza métodos de probabilidade como é o caso do Filtro de Kalman para estimar e prever a localização do veículo.

3.5 *Development and Implementation of an Autonomously Driven Vehicle Prototype* (BILL; SHAHNASSER, 2019)

No trabalho de Bill e Shahnasser (2019) é apresentado o desenvolvimento de um veículo autônomo de baixo custo, utilizando um carro de controle remoto. A automatização do modelo foi realizada utilizando duas plataformas de sistemas embarcados. Um Arduíno foi empregado na supervisão e controle dos motores que realizam a movimentação do veículo, e também na interface com os sensores infravermelhos que permitem realizar a detecção de obstáculos próximos ao dispositivo. A placa Jetson TX2 foi aplicada com uma câmera monocular para a função de detecção de objetos, utilizando a biblioteca de *deep learning*, TensorFlow. No trabalho essa funcionalidade é usada para detectar pessoas e controlar o veículo para manter uma determinada distância da pessoa encontrada.

Para que as informações possam ser compartilhadas entre as entidades do projeto de Bill e Shahnasser (2019), foi estabelecida uma rede de comunicação TCP/IP, desenvolvida com o *framework Robot Operating System* (ROS). A arquitetura *publisher-subscriber* desta biblioteca

permitiu que os módulos desenvolvidos do carro autônomo procedessem de forma independente uns dos outros, garantindo assim que uma falha em um dos módulos, não prejudicasse o funcionamento das demais em execução.

O presente trabalho, semelhante ao que foi apresentado em Bill e Shahnasser (2019) também emprega a independência dos módulos que realizam as funcionalidades do veículo. Para isso o código de controle do veículo foi dividido em módulos responsáveis pela aquisição dos dados, controle dos motores e comunicação com as demais partes do sistema. Além disso, a arquitetura *publisher-subscriber* também está presente neste trabalho através do protocolo de comunicação MQTT, que trabalha com a assinatura e publicação dos dados em tópicos para assim realizar a troca de informação entre a aplicação *mobile* e o sistema embarcado e entre este e o sistema de localização. Todavia o trabalho proposto ainda possui a funcionalidade de comandar o veículo remotamente e também estimar a sua localização, o que não é apresentado no trabalho de Bill e Shahnasser (2019).

3.6 Análise Comparativa

Nesta Seção é apresentado uma comparação entre este trabalho e os trabalhos relacionados elencados neste capítulo.

O Quadro 1 demonstra uma análise comparativa entre os trabalhos relacionados e o trabalho proposto. Os seguintes aspectos de cada trabalho são observados: desenvolve o modelo físico; consegue detectar obstáculos; consegue controlar o veículo remotamente; realiza a estimativa de localização do veículo em ambiente *indoor*. De todos os trabalhos relacionados, apenas o de Qu *et al.* (2016) não desenvolve o modelo físico, já que os autores focaram na simulação do veículo e utilizaram para testes um dispositivo pronto. Todos os trabalhos, incluindo o que está sendo proposto, consegue detectar obstáculos na trajetória do veículo. Já no quesito que verifica se o modelo físico pode ser comandado ou controlado remotamente de alguma forma, apenas os trabalho de Chamorro e González (2019) e Chemhengcharoen *et al.* (2019) possuem essa característica, que também é proporcionada na solução proposta. O presente trabalho também disponibiliza a funcionalidade de estimar a posição do veículo em um ambiente *indoor*, o que só está presente nos trabalhos de Veloso (2019) e de Qu *et al.* (2016).

Quadro 1 – Comparação entre os trabalhos

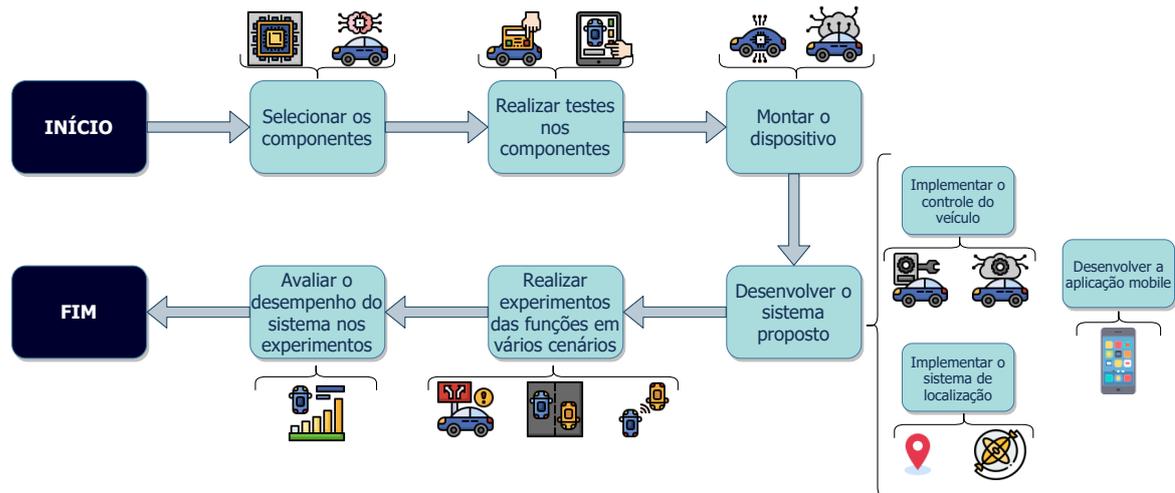
Trabalho	Desenvolve o modelo físico	Detecta obstáculos	Consegue controlar o veículo remotamente	Realiza a estimativa de localização do veículo em ambiente <i>indoor</i>
Qu <i>et al.</i> (2016)	Não	Sim	Não	Sim
Chamorro e González (2019)	Sim	Sim	Sim	Não
Chemhengcharoen <i>et al.</i> (2019)	Sim	Sim	Sim	Não
Veloso (2019)	Sim	Sim	Não	Sim
Bill e Shahnasser (2019)	Sim	Sim	Não	Não
Trabalho Proposto	Sim	Sim	Sim	Sim

Fonte: Elaborado pelo autor.

4 PROCEDIMENTOS METODOLÓGICOS

Neste capítulo são apresentados os passos que serão executados com o objetivo de desenvolver o dispositivo proposto neste trabalho. A sequência em cada tarefa será realizada para a concepção deste projeto, é indicada na Figura 6.

Figura 6 – Fluxograma dos procedimentos.



Fonte: Elaborado pelo autor.

Cada uma das etapas apresentadas no fluxograma são detalhadas ao longo deste capítulo, iniciando da Seção 4.1 que explica a parte de seleção dos componentes para o dispositivo até a Seção 4.8, onde é descrito como a validação do protótipo será realizada.

4.1 Selecionar os componentes

Inicialmente é efetuado um estudo sobre os principais dispositivos necessários para realizar a montagem de um protótipo de veículo autônomo, tais como, sensores, motores, chassis, baterias, controladores e entre outros componentes. Para esta análise foram observados e verificados, quais os componentes empregados no desenvolvimento de veículos autônomos de outros trabalhos, tais como aqueles produzidos nos trabalhos relacionados desta proposta e em outros da literatura.

Após determinar a *hardware* requisitado no projeto do veículo autônomo, é preciso definir a quantidade e o modelo dos componentes que vão ser empregados para este trabalho, como, por exemplo, o chassis (de quatro ou duas rodas), qual o tipo do sensor de distância utilizado (ultrassônico, fotoelétrico ou combinar diferentes modelos de cada tipo), qual a forma

de alimentação do circuito (baterias ou pilhas), entre outras questões. Para analisar esse quesito do projeto, alguns pontos precisam ser examinados, tais como: o custo de cada dispositivo, a compatibilidade entre os aparelhos, o nível de precisão e resolução dos sensores, a faixa de valores de operação dos sensores, o consumo energético de cada equipamento, as dimensões de cada objeto e entre outras condições, cuja avaliação é extremamente importante para o prosseguimento das próximas etapas do trabalho.

4.2 Realizar testes nos componentes

Após realizar a escolha dos materiais empregados na constituição do protótipo, é indispensável que testes individuais sejam realizados em cada componente, para averiguar se estes estão funcionando corretamente. Nesta etapa, os componentes elétricos, eletromecânicos e eletrópticos precisam ser analisados para verificar se nenhum destes está danificado de alguma forma. Para isso então, os circuitos são examinados se estão atuando de forma exata, exibindo as correntes e tensões desejadas para cada extremidade e se as conexões elétricas existentes em cada dispositivo estão bem feitas.

Para realização destes testes e experimentos, alguns aparelhos eletrônicos são necessários para efetuar tais aferições, como, por exemplo, um multímetro. Este equipamento permite medir e avaliar a tensão, corrente e outras grandezas elétricas de um dispositivo, acoplando suas pontas de prova sobre os pontos cuja variável deseja ser examinada.

Além de verificar os circuitos elétricos dos componentes do projeto, nesta fase algumas funções básicas de cada dispositivo também vão ser testadas, para analisar se as especificações e características designadas em cada dispositivo estão válidas. Para este passo, a mensuração de uma distância conhecida será efetuada pelos sensores de distância, o controle de rotação dos motores para os diferentes sentidos será observado, a saída das variáveis de aceleração e velocidade angular da IMU será observada, para cenários com o veículo em movimento ou parado. Cada um destes testes individuais pode ser realizado primeiramente em um microcontrolador, cuja programação é menos complexa, como o Atmega328p presente na placa Arduino Uno, para depois ser testado na controladora que será realmente empregada para o controle do veículo.

4.3 Montar o dispositivo

Terminada a etapa de realização de testes dos componentes, o próximo passo é montar o protótipo do veículo que está sendo proposto neste trabalho. Para isso um planejamento do projeto do dispositivo precisa ser realizado, para a definição de onde cada componente será posicionado, de modo que todas as conexões necessárias entre estes elementos consigam ser executadas, para que o veículo atue de forma satisfatória, procedendo às funções implementadas e planejadas nas devidas situações.

A localização de alguns componentes no veículo, como os sensores de distância utilizados na detecção de obstáculos, é um fator crucial do projeto. O ângulo de detecção destes equipamentos com sua posição definirá quais os campos de decisão do veículo que vão permitir encontrar os obstáculos no caminho e auxiliar enquanto uma curva ou desvio destes mesmos obstáculos estiver sendo realizada.

Após a montagem do protótipo, um teste de integração será realizado, com o objetivo de verificar se todos os componentes estão conectados e funcionando de modo correto. Além disso, a calibração e configuração dos sensores de distância e da unidade de medição inercial, também é feita nesta etapa, para que assim, todos os componentes estejam preparados para realizar seus papéis nas funções de controle do veículo implementadas.

4.4 Implementar o controle do veículo

A medida que o dispositivo é montado, os algoritmos que realizam o controle do veículo são desenvolvidos. Nesse caso, algumas funções serão implementadas e executadas para realizar a configuração dos pinos da placa, utilizados em cada componente. Além destas, também serão desenvolvidas funções que ficarão sendo executadas dentro de um laço durante toda a execução do programa. São elas: realizar as leituras dos valores obtidos por cada sensor; receber as mensagens de controle enviadas do aplicativo via protocolo MQTT; alterar o controle dos motores do veículo de acordo com os valores recebidos pelo aplicativo ou pela lógica de navegação autônoma; publicar os valores de distância, aceleração e velocidade angular aferidos pelos sensores para um tópico MQTT definido.

Neste trabalho as seguintes funções de um veículo semi-autônomo serão elaboradas e testadas:

- **Realizar a ação desejada:** de acordo com a ação demandada pelo usuário do aplicativo, o

veículo deve iniciar a realizar a manobra desejada, podendo ser esta, mover para a frente, mover para trás, parar o dispositivo, girar para a direita no sentido horário, girar para a direita no sentido anti-horário, girar para a esquerda no sentido horário e girar para a esquerda no sentido anti-horário.

- **Detectar obstáculos na trajetória:** enquanto realiza algum percurso no modo autônomo, alguns obstáculos podem ser encontrados pelo veículo e este, utilizando-se dos sensores de distância, deve identificar esses bloqueios e parar o veículo, evitando assim a colisão.
- **Desviar dos obstáculos:** após realizar a detecção, o veículo deve preparar e executar uma manobra evasiva para desviar do obstáculo encontrado no caminho.
- **Realizar o comando esperado:** um modo de operação, em que o dispositivo obedece aos comandos emitidos pelo usuário da aplicação *mobile*. Neste modo, cada comando já possui um tempo estabelecido para ser efetuado, e após sua execução o veículo se encontrará em uma nova localização. Os comandos possíveis neste trabalho são: mover para a frente, mover para a trás, girar o veículo 90° para a direita, girar o veículo 90° para a esquerda, girar o veículo 180° para a direita, girar o veículo 180° para a esquerda.

Os algoritmos desenvolvidos precisam configurar de forma correta os pinos, sensores, atuadores e outros dispositivos internos e externos à placa ESP32, para que desse modo, seja possível realizar as aferições dos dados de distância a um possível obstáculo, velocidade exercida pelo veículo, ângulo do eixo, e entre outros por meio dos sensores escolhidos especificamente para determinar o valor destas grandezas físicas. Além disso, a lógica presente nos algoritmos também é responsável, por coordenar os momentos em que a placa deve enviar os sinais digitais que realizam o controle dos motores, responsáveis pela movimentação do dispositivo.

4.5 Desenvolver a aplicação *mobile*

Nesta etapa, uma aplicação *mobile* foi desenvolvida, para enviar os comandos ao sistema embarcado e assim realizar o controle do veículo, quando este não estiver no modo de execução autônomo. A aplicação precisa inicialmente se conectar ao *broker* MQTT utilizado para gerenciar as mensagens publicadas em cada tópico utilizado. Posteriormente de acordo com os comandos desejados e efetuados pelo usuário, um valor será publicado no tópico de controle do movimento do carro, para que esse valor seja recebido no sistema embarcado e assim a ação seja realizada no veículo.

4.6 Implementar o sistema de localização

Para ser possível identificar a localização do veículo em um ambiente *indoor* utilizando as técnicas do Filtro de Kalman, é necessário primeiramente definir um modelo matemático para indicar a posição e movimentação do veículo. Para isso os valores de giro, aceleração e distâncias obtidos pela unidade de medição inercial e pelos sensores de distância são utilizados, de modo que seja possível identificar onde se encontra o veículo em meio ao ambiente onde ele foi colocado.

Após a definição do modelo, o algoritmo do KF foi desenvolvido para receber como entrada os valores dos sensores citados anteriormente, que serão enviados via protocolo MQTT e a partir destes determinar a posição estimada do veículo no escopo do ambiente onde este está atuando.

4.7 Realizar experimentos das funções em vários cenários

Após a montagem do *hardware* do dispositivo e a implementação dos algoritmos que realizam o seu controle, uma nova etapa do projeto é iniciada. Nesta, alguns experimentos serão preparados e desempenhados para observar se as funcionalidades desenvolvidas no protótipo estão sendo executadas como o desejado.

Para cada função que foi testada, alguns cenários foram organizados, de modo a perceber o comportamento do protótipo para diferentes situações que necessitam de uma resposta precisa e hábil do veículo. A resposta do veículo para os comandos enviados pelo usuário do aplicativo também vão ser observadas, para determinar se existe um atraso significativo entre o envio do comando e a realização da ação demandada.

Cada um dos experimentos planejados nos mais diferentes cenários, foram efetuados diversas vezes, para que se possa ter dados o suficiente para uma boa avaliação do desempenho do protótipo em relação às funções desenvolvidas, e assim possíveis melhorias e correções possam ser determinadas.

Além disso, também foram realizados testes para observar se o sistema de localização está conseguindo determinar de forma satisfatória a posição e orientação do veículo no ambiente *indoor*.

4.8 Avaliar o desempenho do sistema nos experimentos

Mediante a execução dos testes e experimentos no veículo, o desempenho do dispositivo desenvolvido neste trabalho é avaliado com base em algumas métricas bem definidas, tais como, a quantidade de decisões acertadas para cada ocasião, o tempo médio necessário para executar uma ação de acordo com a velocidade com que o veículo está se movimentando, o erro entre a estimativa da posição realizada pelo sistema e a localização real do dispositivo, a verificação da trajetória formada pelos pontos identificados no sistema de localização, entre outros.

A avaliação do dispositivo, requer também analisar o desempenho do protótipo em comparação com os demais veículos desenvolvidos em outros trabalhos, tais como aqueles investigados e elencados na Seção 3 de trabalhos relacionados. Apesar de alguns possuírem objetivos distintos ao deste trabalho, alguns pontos similares na avaliação de desempenho ainda podem ser encontrados, para verificar assim, se a performance alcançada está próxima, superior ou inferior aquela obtida nesses outros trabalhos.

Todas as informações adquiridas ao longo das avaliações, são de extrema importância para a evolução deste trabalho, pois permitem observar os pontos que precisam de ajustes, para que o protótipo possa representar de forma mais fidedigna o comportamento de um veículo autônomo e controlado, cumprindo assim os objetivos elencados nesta proposta.

5 MATERIAIS E MÉTODOS

Nesta seção será detalhada a solução proposta para o protótipo do veículo autônomo desenvolvido e como os dados obtidos por esse dispositivo são utilizados para identificar sua localização no ambiente. Cada um dos módulos explica sobre os componentes e métodos utilizados na concepção da solução e como eles interagem uns com os outros para fornecer a funcionalidade objetivada com este trabalho.

5.1 Modelo da proposta

O objetivo deste trabalho é o desenvolvimento de um protótipo de veículo autônomo que pode ser controlado remotamente e que consegue realizar a estimativa da sua localização em um ambiente *indoor*. Para isso diferentes tecnologias e materiais foram analisados para serem utilizados em cada uma das partes da solução. O sistema desenvolvido pode ser dividido em três partes distintas:

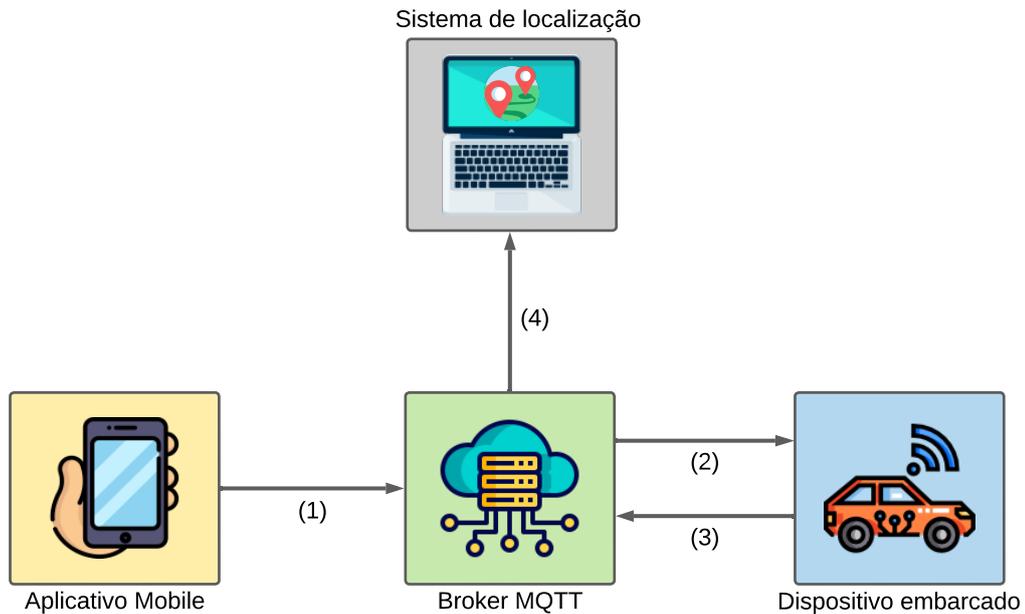
1. O sistema embarcado, responsável por realizar o controle dos componentes do veículo e por mensurar as grandezas físicas associadas ao movimento do dispositivo.
2. A aplicação *mobile* encarregada de controlar o veículo através de comandos e mensagens que são enviadas para o sistema embarcado.
3. O sistema de localização do veículo que desempenha o papel de implementar o Filtro de Kalman Estendido sobre o modelo matemático da movimentação do veículo.

Cada um destes segmentos constituintes da solução, precisam trocar informações entre eles para que o correto funcionamento do sistema ocorra. Para isso, o protocolo de comunicação MQTT foi escolhido, por se tratar de um protocolo leve, sendo bastante adotado em soluções de IoT e que, portanto já possui características funcionais para realizar a comunicação com sistemas embarcados, como baixa latência e tamanho do corpo da mensagem reduzido.

Na comunicação com MQTT, é necessária uma unidade central, denominada *broker*, responsável por receber as mensagens publicadas nos tópicos e repassar essas para os clientes assinantes. Portanto, um *broker online* público foi escolhido para intermediar a troca de mensagens entre as entidades do sistema. O fluxo de troca de informações entre as partes do sistema e o servidor *broker* MQTT é apresentado na Figura 7, onde cada número representa uma troca de informação entre as entidades.

Ao inicializar o sistema embarcado, ele se inscreve nos tópicos escolhidos para

Figura 7 – Fluxo de informações no sistema.



Fonte: Elaborado pelo autor.

receber os comandos enviados pelo aplicativo *mobile*. O usuário da aplicação então, inicia o aplicativo e envia os comandos de controle para acionar as funcionalidades no veículo, publicando mensagens em tópicos específicos do *broker* MQTT (1). Os nomes dos tópicos foram escolhidos concatenando-se o valor de um id aleatório com o nome da funcionalidade das mensagens enviadas para aquele tópico. Para o caso dos comandos de controle do veículo enviados pelo aplicativo, o tópico foi nomeado como, ID/controlCar.

Os valores emitidos nas mensagens pelo aplicativo são recebidas no sistema embarcado (2), já que este elemento está inscrito no tópico ID/controlCar. As mensagens são tratadas e conforme os valores são extraídos, uma ação é designada sobre o carro, como alterar o seu modo de funcionamento para autônomo ou manual, ou então movimentá-lo para frente, para a direita, para trás, para a esquerda e entre outras combinações.

Simultaneamente ao processo de esperar e executar os comandos recebidos, o dispositivo embarcado realiza a aferição dos valores obtidos pelos sensores, detectando as distâncias dos objetos próximos ao veículo e também da aceleração e velocidade angular (giro) exercidas sobre este componente. Os valores mensurados são publicados em tópicos específicos no *broker* MQTT (3). As distâncias aos obstáculos são enviadas ao tópico ID/distanceSensor, enquanto, que os valores de aceleração e giro são emitidos para os tópicos ID/accelerometer e ID/gyroscope, respectivamente.

As mensagens com os valores mensurados pelos sensores dos sistemas embarcados,

são enviadas do *broker* MQTT para o sistema de localização (4), que para isso necessita se inscrever nos tópicos onde estes dados são publicados. Os dados então são tratados e utilizados na implementação do EKF, para designar uma boa estimativa da real posição e angulação do veículo no ambiente em que este está introduzido.

As seções seguintes apresentam detalhes sobre a implementação realizada para cada uma das partes do sistema dessa proposta.

5.2 O veículo autônomo

Para desenvolver o protótipo do veículo autônomo, foi necessário inicialmente definir os componentes de *hardware* utilizados para formar o dispositivo. Primeiramente, foi designado para compor o corpo do veículo, um chassi de acrílico, ao qual foi acoplado duas rodas de borracha em suas laterais. Estas são utilizadas para prover o movimento do veículo, por isso estão diretamente conectadas à motores DC que precisam ser alimentados com tensões entre 3 à 6 volts. Além destas rodas, uma roda boba foi adicionada ao chassi para equilibrar a estrutura do veículo e permitir que ele se desloque. Diferentemente das outras rodas, a roda boba não possui um motor DC interligado a ela e portanto, não pode ser controlada pelo microcontrolador. A estrutura do veículo já montada com os componentes citados é apresentada na Figura 8

Figura 8 – Estrutura do veículo.



Após formar o corpo do veículo, iniciou-se a montagem dos circuitos de controle e de percepção do dispositivo. Para efetuar todo o controle lógico do sistema embarcado, o microcontrolador da placa ESP32 foi escolhido. Este componente ficou encarregado de controlar o movimento das rodas, tratar das informações adquiridas pelos sensores e enviar e receber os dados obtidos nas mensagens via protocolo MQTT. Os circuitos necessários para interligar os componentes relacionados a cada uma dessas funções e os algoritmos que as desenvolveram são apresentados nas subseções a seguir.

5.2.1 *Circuito de controle dos motores*

Os motores DC são dispositivos que não podem ser diretamente conectados aos pinos de expansão de uma placa de desenvolvimento, pois requerem um consumo elevado de corrente, o que pode ocasionar danos irreversíveis à controladora. Por esse motivo, um *driver* ponte H foi utilizado, pois, esse consegue fornecer a potência adequada e realizar o controle desses componentes através dos valores recebidos em suas entradas.

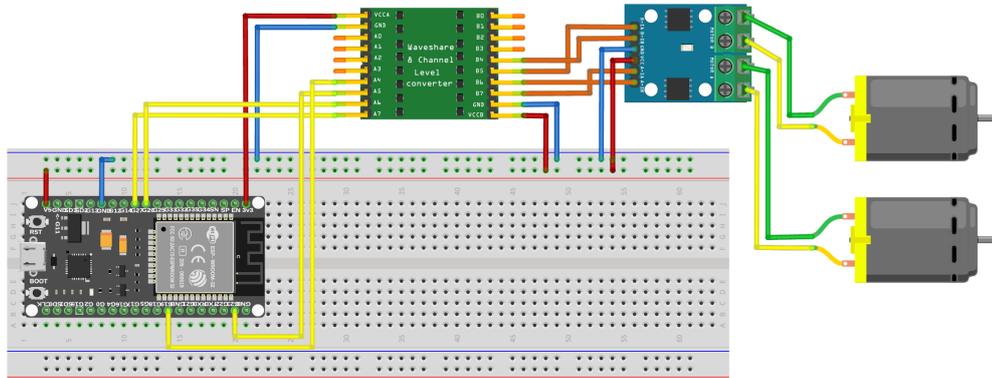
O *driver* ponte H empregado, utiliza um chip L9110s, que consegue controlar até dois motores DC. Cada motor é conduzido por duas entradas distintas, que quando apresentam o mesmo nível lógico param o movimento do motor, e quando apresentam níveis lógicos diferentes realizam o movimento para o sentido horário ou anti-horário dependendo de qual entrada encontra-se em nível lógico alto e como as conexões entre o motor e as saídas da ponte H foram realizadas.

Os pinos de entrada do *driver* ponte H, não puderam ser diretamente conectados aos pinos de GPIO da ESP32, pois esse controlador tem como valor de tensão de nível lógico alto, 3,3 V, enquanto o módulo ponte H espera como nível alto, uma tensão de 5 V. Logo a alteração dos valores desses pinos pelo microcontrolador não conseguiria impor o funcionamento correto da ponte H e conseqüentemente do movimento das rodas do veículo. Em virtude disso, um circuito conversor de nível lógico foi adicionado ao projeto para converter as tensões dos pinos de GPIO da placa para valores aceitados pelo módulo ponte H.

Todas as conexões realizadas entre a placa ESP32 e os componentes relatados a essa parte do protótipo são ilustradas na Figura 9. Observando então, a Figura 9, percebe-se que 4 pinos foram conectados as entradas do conversor de nível lógico. As saídas desse conversor foram interligadas as entradas do módulo ponte H, que por seu lado, foi conectado aos motores que realizam o movimento das rodas. Para alimentar o nível alto do conversor foi utilizado a

tensão de 5 volts, proveniente da ESP32, enquanto o nível inferior do conversor foi interligado ao pino da placa que disponibiliza 3,3 V. Na alimentação do *driver* ponte H foi utilizada uma bateria externa de 5 V.

Figura 9 – Circuito de controle dos motores.



Fonte: Elaborado pelo autor.

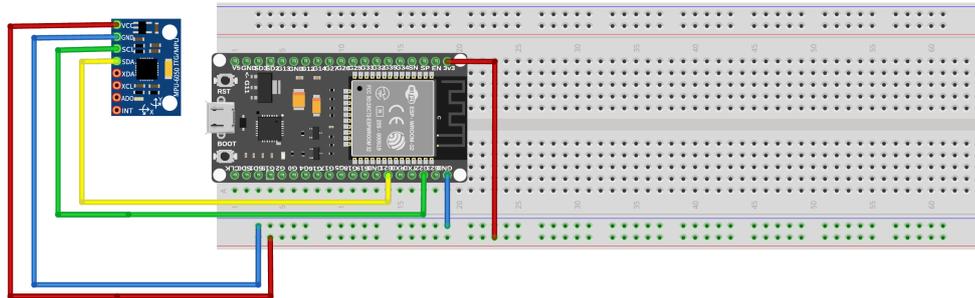
5.2.2 Circuito de conexão com os sensores

No presente trabalho dois tipos de sensores são utilizados para mensurar as grandezas físicas que auxiliam no controle do veículo: o sensor de distância a *laser* VL53L0X e a IMU MPU6050. Ambos possuem a possibilidade de se comunicar com o microcontrolador através do protocolo I2C. Com isso não são necessários configurar uma numerosa quantidade de pinos de GPIO para realizar o funcionamento desses sensores, já que a comunicação I2C, funciona como um barramento serial que possui apenas uma linha para o envio de dados (SDA) e outra para a transmissão do *clock* (SCL).

Para a unidade de medição inercial MPU6050, as únicas conexões necessárias para o funcionamento desse dispositivo são os pinos SDA e SCL para a comunicação I2C e os pinos de alimentação (VCC e GND). A Figura 10 apresenta o diagrama do circuito de conexão entre este módulo e a placa ESP32.

Os pinos para a comunicação I2C foram interligados aos seus correspondentes na ESP32, que na configuração padrão desta placa são os pinos 22 para o SCL e o pino 21 para o SDA. As demais conexões são utilizadas para alimentação, sendo o fio azul, utilizado para representar a conexão com o *ground* da placa e o fio vermelho para indicar a conexão com o pino de alimentação de 3,3 V.

Figura 10 – Circuito de conexão com o MPU6050.



Fonte: Elaborado pelo autor.

As conexões entre a placa ESP32 e o sensor de distância VL53L0X são semelhantes as que foram apresentadas na Figura 10. Neste trabalho foram utilizados três sensores deste tipo, com a finalidade de medir as distâncias aos obstáculos que estejam na frente, na lateral direita e na lateral esquerda do veículo. Todos esses dispositivos podem se conectar diretamente aos mesmos pinos de I2C da placa, com a condição de que os endereços utilizados para cada um sejam diferentes, para que deste modo a placa consiga identificar cada um deles.

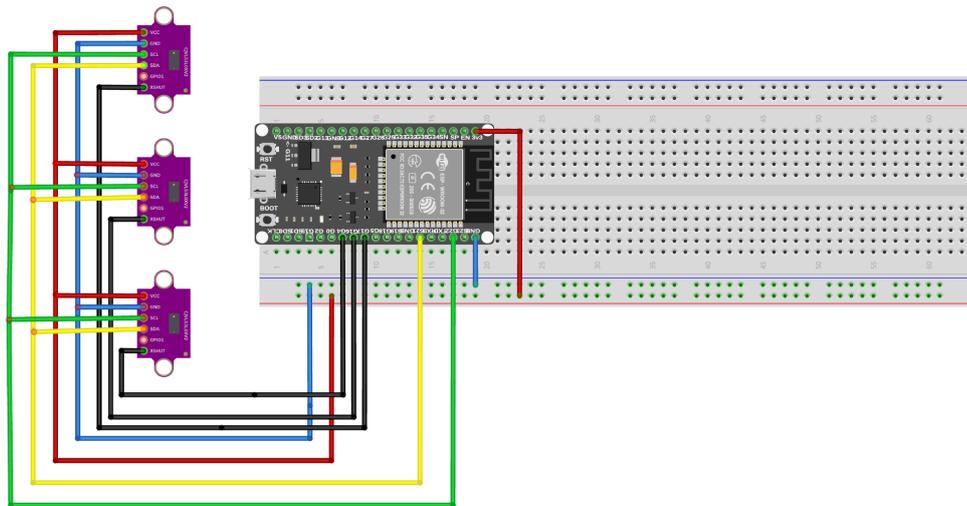
Um ponto importante em relação aos sensores de distância VL53L0X, é que para ocorrer a correta inicialização dos dispositivos se faz necessário que os demais sensores desse tipo não estejam ativos. Para realizar essa ação, existe um pino nesse sensor (XSHUT) que quando colocado em nível lógico alto, coloca o sensor em modo *standby*. Portanto, para ser possível a correta inicialização desses componentes, os pinos XSHUT de cada sensor foram interligados à pinos de GPIO da placa, com o objetivo de controlar os momentos exatos de ativar e desativar cada sensor durante a inicialização. A Figura 11 ilustra essas conexões e as demais que foram realizadas para acoplar esses módulos ao microcontrolador.

As demais conexões apresentadas na Figura 11 são similares as que foram realizadas com a IMU, incluindo a alimentação ao pino de VCC dos componentes, ao qual também foi utilizado o pino da placa que fornece 3,3 volts de tensão contínuos.

5.2.3 Circuito de identificação de estado

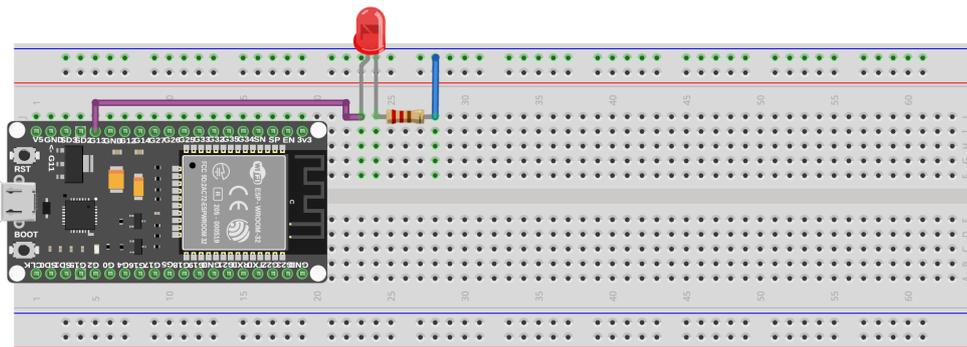
O último circuito adicionado ao sistema embarcado, é apenas um circuito para acender e apagar um LED, que serve para indicar quando ocorreu a correta inicialização de todos os componentes. Basicamente o LED está interligado a um pino de GPIO da placa e a um resistor de 1 $K\Omega$ conectado ao *ground*, como pode ser observado na Figura 12.

Figura 11 – Circuito de conexão com os sensores de distância.



Fonte: Elaborado pelo autor.

Figura 12 – Circuito de identificação de estado.



Fonte: Elaborado pelo autor.

Devido a diversos fatores, como má conexão entre os fios ou captura de sinais ruidosos, pode ocorrer dos sensores não inicializarem corretamente. Pelo fato dos sensores de distância não possuírem nenhum LED ou qualquer outro componente que indica se eles estão funcionando de modo correto, fica complicado de saber se todos os dispositivos do protótipo foram inicializados com êxito. Assim sendo, o circuito com o LED tem como objetivo, emitir um sinal visual para representar que todos os componentes estão funcionando e logo o veículo está apto a receber os comandos para efetuar suas ações.

Portanto, com esse circuito é possível identificar o estado do veículo, com relação à ocorrência ou não de erros durante a configuração e inicialização dos dispositivos constituintes do protótipo.

5.2.4 *Protótipo do veículo*

Após definir e montar os circuitos constituintes do dispositivo embarcado, o protótipo do veículo foi desenvolvido, conectando-se as partes de cada circuito apresentado a placa de desenvolvimento ESP32 e posicionando cada elemento no chassi que havia sido montado. Para ocorrer o correto funcionamento de cada circuito apresentado, foram escolhidos pinos de GPIO distintos para cada componente. Como todos os sensores utilizados no protótipo trabalham com comunicação I2C, foi possível conectar todos eles aos pinos responsáveis por esse protocolo na placa de desenvolvimento.

Foi definido também como iria se proceder à alimentação de cada componente, de modo que uma bateria de 5 V foi utilizada para alimentar a ESP32 e os sensores conectados a ela, enquanto outra bateria de 5 V foi aplicada para efetuar a alimentação do módulo ponte H e dos motores.

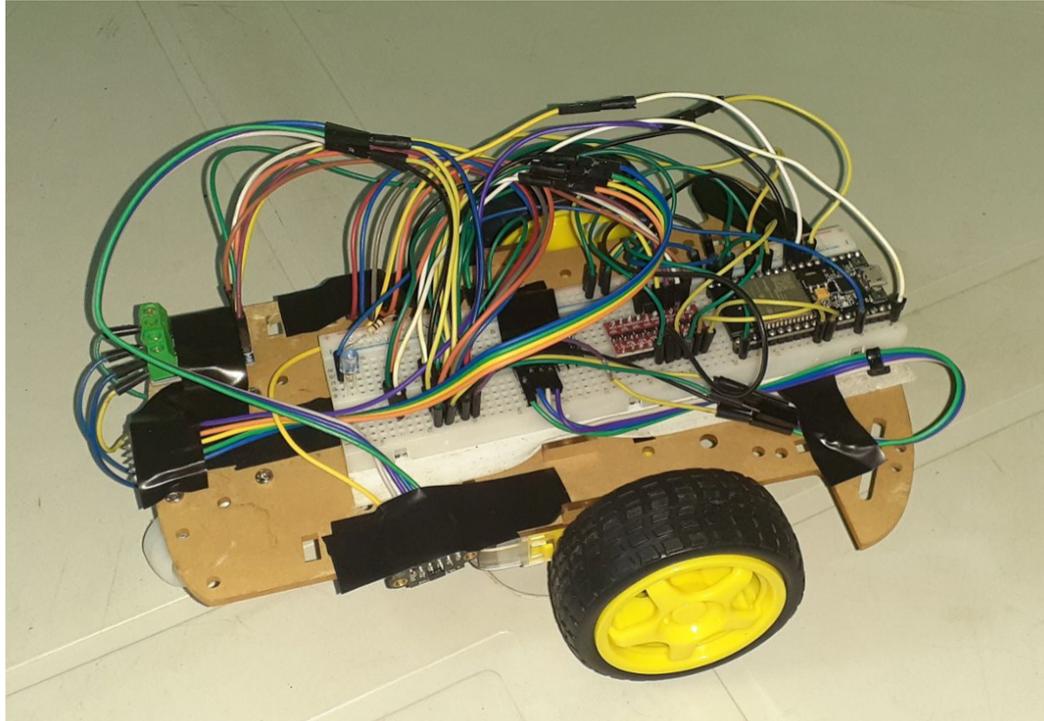
A definição do posicionamento de cada dispositivo também foi realizada nessa etapa de montagem do protótipo. A placa ESP32 ficou na parte superior do chassi juntamente com o *driver* ponte H e a unidade de medição inercial. Já os sensores de distância foram estabelecidos em pontos estratégicos, para ser possível detectar os obstáculos próximos ao veículo. Um sensor foi acoplado na parte da frente do veículo, enquanto que os demais foram instalados nas laterais do dispositivo. Todo o aspecto dos componentes acoplados no veículo, pode ser melhor observado na Figura 13, que apresenta o protótipo desenvolvido.

5.2.5 *Implementação do algoritmo*

O programa de controle do veículo foi desenvolvido na linguagem de programação C++. Inicialmente é realizada a configuração de todos os pinos utilizados pelos motores, sensores e o LED. Em seguida é efetuada a conexão Wi-Fi da placa com a rede, para que se possa conectar ao *broker* MQTT e se inscrever nos tópicos definidos para receber as mensagens de controle. Posteriormente é feita a inicialização dos sensores e quando esta é realizada com sucesso, o LED acende por 3 segundos para indicar que todos os componentes estão funcionando e portanto o dispositivo está pronto para ser utilizado.

Dentro do *loop* principal da aplicação, é executada uma função que verifica se a conexão Wi-Fi ainda está mantida, para que em seguida, seja analisado se alguma mensagem foi publicada no tópico de controle. As mensagens válidas para alterar a configuração do robô

Figura 13 – Protótipo do veículo autônomo.



Fonte: Elaborado pelo autor.

são explicadas na Tabela 1. Além disso, também são executados dentro do laço infinito, funções para ler os valores de distância dos sensores e também os valores de aceleração e giro da IMU. Os valores obtidos por todas essas unidades são tratados e encapsulados em mensagens que são publicadas no *broker*.

Tabela 1 – Mensagens para alterar o modo de operação do veículo

Mensagem	Modo de operação ativado
M	Modo manual
A	Modo autônomo
C	Modo de comandos

Fonte: Elaborado pelo autor.

Cada um dos modos de operação apresentados na Tabela 1, possuem uma implementação diferente que realiza o controle do veículo de forma particular.

No modo de operação autônomo, o veículo começa seguindo para frente e ao encontrar um obstáculo em seu caminho ele é parado e analisa os valores de distância dos sensores laterais. Se ambos os valores de distâncias das laterais obtidas forem muito baixos, indicando obstáculos em todos os lados, um movimento é realizado para trás por uma certa quantidade de tempo. Após isso, as laterais são novamente analisadas e caso alguma delas não possua um obstáculo próximo, o veículo será rotacionado 90° para este lado e voltar a seguir em

linha reta para frente.

Nos modos de operação manual e de comandos, o veículo é iniciado parado e fica nesse estado esperando os comandos recebidos pela aplicação *mobile* para iniciar seu movimento. Basicamente, nesses dois modos o dispositivo recebe valores nas mensagens publicadas pelo aplicativo e relaciona a mensagem a execução de um determinado movimento. As mensagens válidas e a ação realizada pelo robô para o modo manual e o modo de comandos são apresentados nas Tabelas 2 e 3 respectivamente.

Tabela 2 – Mensagens do modo de operação manual

Mensagem	Comando associado
000	Parar
101	Mover para frente
111	Mover para trás
001	Girar para a direita pela frente
100	Girar para a esquerda pela frente
011	Girar para a direita de ré
110	Girar para a esquerda de ré

Fonte: Elaborado pelo autor.

Tabela 3 – Mensagens do modo de comandos

Mensagem	Comando associado
111	Mover para frente
010	Mover para trás
001	Girar para a direita 90°
100	Girar para a esquerda 90°
011	Girar para a direita 180°
110	Girar para a esquerda 180°

Fonte: Elaborado pelo autor.

A principal diferença entre os modos apresentados, é que no manual a ação é realizada por tempo indeterminado, até que um novo comando seja recebido. Por outro lado, no modo de comandos, a ação é efetuada por um espaço de tempo determinado e após esse tempo o movimento é interrompido, permitindo assim um maior controle para os testes de localização do veículo.

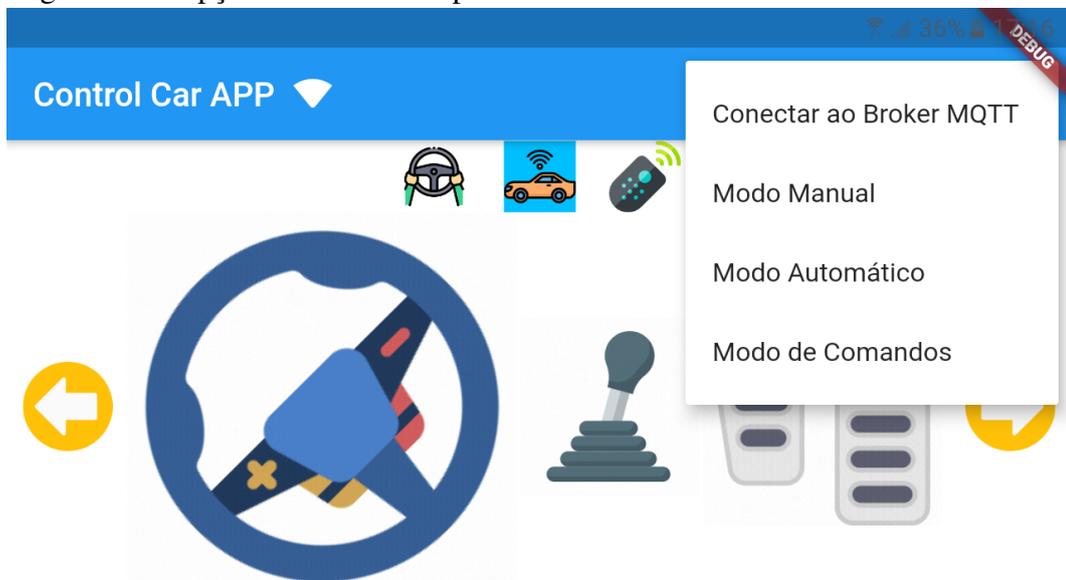
5.3 A aplicação *mobile*

Para possibilitar que o controle do veículo desenvolvido possa ocorrer de forma remota, uma aplicação *mobile* foi implementada na linguagem Dart, utilizando o Flutter como *framework* de desenvolvimento. Para então realizar o controle do dispositivo embarcado, a

aplicação se conecta ao *broker* MQTT e começa a enviar mensagens, de acordo com as ações realizadas pelo usuário. As mensagens contêm os comandos válidos apresentados nas Tabelas 1, 2 e 3 e esses comandos são recebidos no sistema embarcado para que o movimento ou ação no veículo seja executada.

O aplicativo possui 3 telas, utilizadas para representar cada um dos modos de operação do veículo. Além disso, ele também contém um menu na barra superior do aplicativo, que possui 4 opções, como pode ser observado na Figura 14. Basicamente às três opções inferiores servem para ativar cada um dos modos de operação do veículo, enquanto que a primeira opção desempenha a função de realizar a conexão com o *broker*.

Figura 14 – Opções do menu do aplicativo



Fonte: Elaborado pelo autor.

Quando o aplicativo é iniciado, a primeira ação a ser executada é efetuar a conexão com o *broker* MQTT. Quando a tentativa de conexão é efetuada com sucesso, o ícone ao lado do título do *app* aparece como na Figura 14. No entanto, se por algum motivo a conexão for interrompida o ícone é apresentado com uma faixa, como pode ser observado na Figura 15, de modo que nesse formato, os comandos enviados não serão recebidos no *broker* e consequentemente no dispositivo embarcado.

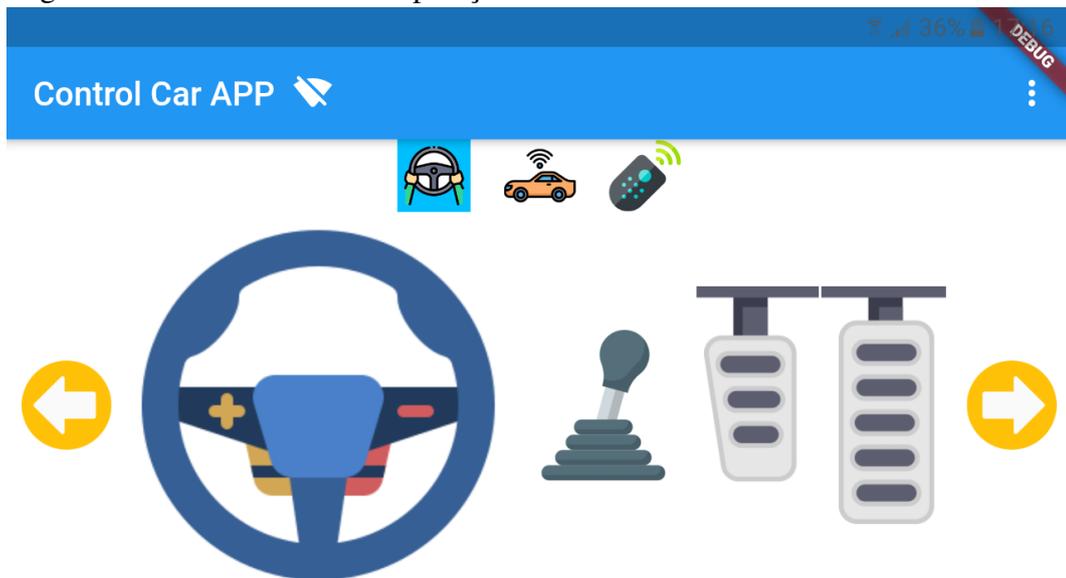
Na Figura 14, também é possível observar 3 imagens que estão logo abaixo da barra no topo do aplicativo. Essas imagens significam o modo de operação que o veículo se encontra. A primeira imagem, da esquerda para a direita representa o modo manual, a segunda o modo autônomo e a terceira o modo de comandos. Conforme o modo que está ativado, um *background*

azul é aplicado sobre a imagem representativa de cada modo.

5.3.1 Modo de operação manual

No modo de operação manual, cuja tela é ilustrada na Figura 15, os comandos para o controle do dispositivo são enviados quando o usuário realiza um toque ou pressiona as imagens dos componentes do veículo, para assim facilitar a utilização do aplicativo para o usuário.

Figura 15 – Tela do modo de operação manual



Fonte: Elaborado pelo autor.

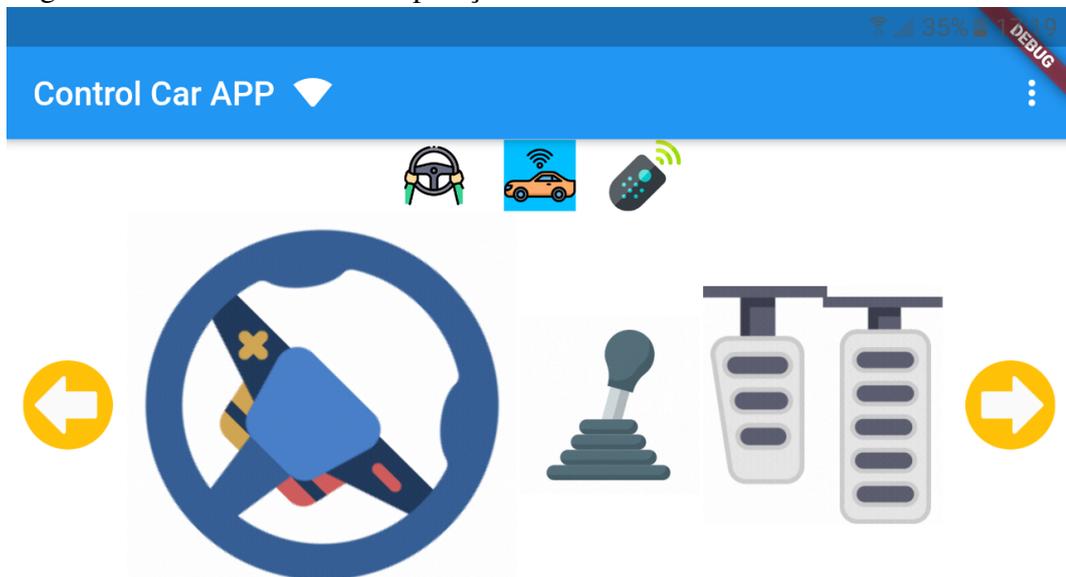
O pedal maior da Figura 15 ao ser tocado, envia um comando para acelerar, o pedal menor envia uma mensagem para parar o veículo, enquanto que um toque no volante pode efetuar uma curva para a direita ou esquerda, conforme a área que for pressionada. O câmbio que se encontra entre o volante e os pedais, é utilizado para indicar se o movimento do veículo será realizado para frente ou para trás. Quando este componente é pressionado a direção é invertida. No caso da Figura 15, ele está inclinado para a direita, representando assim um movimento para frente. Se a imagem do câmbio for tocada, ficará inclinado para a esquerda e os próximos eventos de aceleração ou curva ocorrerão para trás.

Na Figura 15 também é possível observar duas setas nas extremidades laterais da tela. Essas setas não possuem nenhuma ação sobre o dispositivo relacionada, e são utilizadas apenas para sinalizar ao usuário do aplicativo quando uma curva para a direita ou esquerda está sendo realizada. Para isso, a seta (direita ou esquerda de acordo com o movimento) permanece piscando, enquanto a ação de curva está sendo efetuada.

5.3.2 *Modo de operação autônomo*

No modo de operação autônomo, o veículo age de forma independente, logo a tela apresentada na Figura 16, para esse modo no aplicativo, apresenta os mesmos componentes do modo de operação manual, no entanto, são apenas ilustrativos e não transmitem nenhuma mensagem ao serem pressionados. De modo geral, nessa tela os componentes ficam se movendo sem que nenhum comando ou ação tenha sido executada pelo usuário, para demonstrar justamente a característica autônoma do dispositivo.

Figura 16 – Tela do modo de operação autônomo.



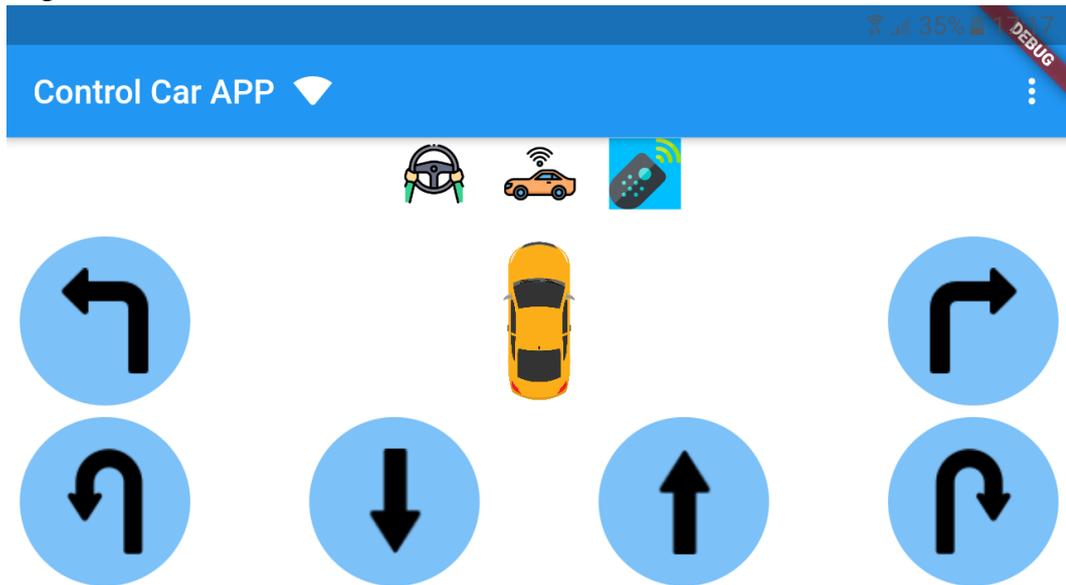
Fonte: Elaborado pelo autor.

5.3.3 *Modo de comandos*

No modo de operação em que o veículo é controlado por comandos específicos, a tela do aplicativo representada na Figura 17, apresenta botões que estão relacionados diretamente a cada ação que pode ser realizada no veículo.

As imagens das setas colocadas em cada botão que aparece na Figura 17 estão diretamente relacionadas a ação que deseja ser executada, tornando assim o seu uso bem intuitivo para o usuário. Os botões na extremidade esquerda indicam giros para a esquerda com angulações de 90 (botão superior) e 180 (botão inferior) graus. De modo similar, os botões na extremidade direita representam os mesmos movimentos, porém para o lado direito. Os botões centrais quando tocados, emitem mensagens para que o veículo desenvolva movimentos para frente (seta

Figura 17 – Tela do modo de comandos.



Fonte: Elaborado pelo autor.

para cima) ou para trás (seta para baixo).

Na Figura 17 também é possível identificar a imagem de um carro observado através de uma visão superior. Essa imagem é utilizada para indicar para o usuário do aplicativo como está a orientação do veículo. Sempre que uma ação de giro é demandada sobre o dispositivo embarcado, a imagem do carro no aplicativo é rotacionada de forma igual, proporcionando ao usuário o conhecimento sobre para qual direção a frente do veículo está apontada.

5.4 O modelo matemático de movimentação do veículo

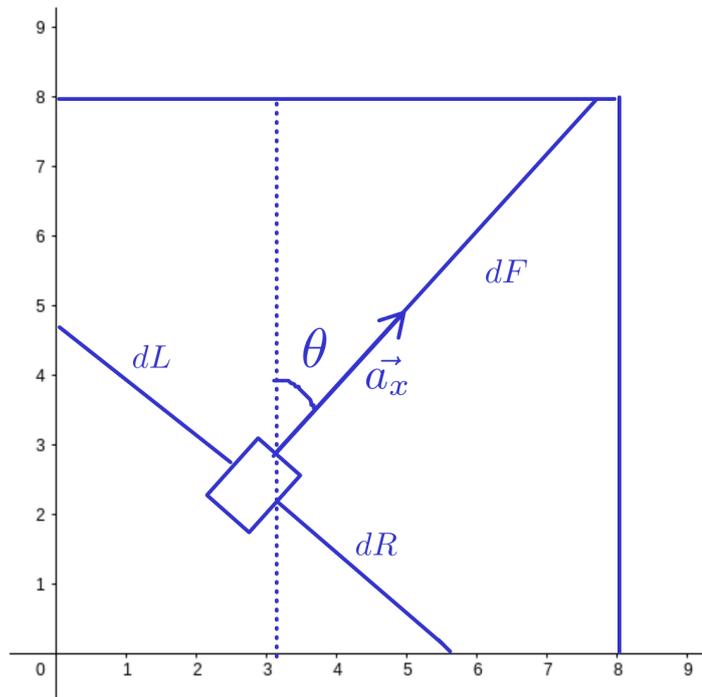
Para ser possível identificar a localização do veículo utilizando a técnica do Filtro de Kalman, é necessário desenvolver um modelo matemático que define como ocorre a movimentação do veículo desenvolvido em meio ao ambiente em que ele está inserido.

Nesta etapa em que acontece a modelagem matemática, é necessário utilizar fórmulas matemáticas que descrevam como ocorre a movimentação do veículo para diferentes configurações no espaço e baseado nos valores obtidos pelos sensores de distância e pela aceleração e giro ao qual o veículo está submetido, para que assim seja possível identificar sua localização.

Considerando que o dispositivo está inserido em um ambiente onde todos os raios de luz emitidos pelos sensores de distância são refletidos em algum ponto, podemos representar o esquema de posição do veículo como na Figura 18.

Na Figura 18, dF , dR e dL representam as distâncias capturadas pelos sensores de

Figura 18 – Modelo de posição do veículo



Fonte: Elaborado pelo autor.

distância, sendo dF a distância capturada pelo sensor frontal, dR a distância do sensor lateral direito e dL a distância capturada pelo sensor lateral esquerdo. Já o vetor \vec{a}_x representa a aceleração mensurada pelo acelerômetro no sentido em que o movimento acontece. Como o veículo pode girar em torno de seu próprio eixo, o valor de θ na Figura 18 é utilizado para representar o ângulo de giro realizado pelo robô.

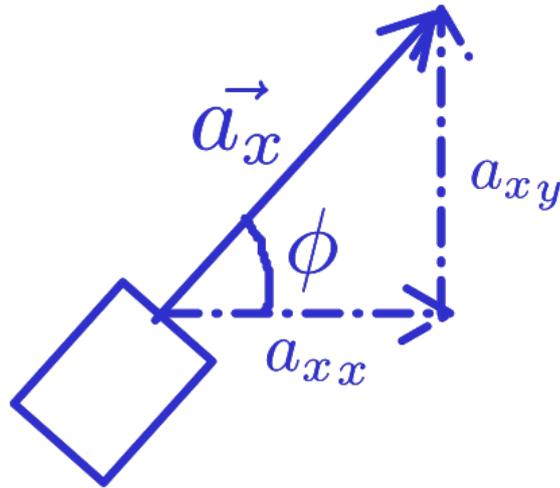
Para determinar a posição do robô baseado no seu movimento podemos recorrer às fórmulas de física do Movimento Retilíneo Uniformemente Variado (MRUV) apresentado na Equação 5.1. Nessa equação, S representa a posição final após o movimento, S_0 é a posição inicial do corpo, v_0 indica a velocidade inicial, Δt a variação de tempo e a o valor da aceleração do corpo.

$$S = S_0 + v_0\Delta t + \frac{1}{2}a\Delta t^2. \quad (5.1)$$

Tanto a aceleração a , quanto a velocidade inicial v_0 são vetores que precisam ser decompostos em seus componentes relacionados ao eixo X e ao eixo Y do sistema de coordenadas global, para que seja possível determinar a posição (x,y) do veículo. Baseado no elemento \vec{a}_x da

Figura 18, podemos definir seus componentes para o eixo X (a_{xx}) e Y (a_{xy}), como na Figura 19. Logo utilizando o ângulo ϕ , o vetor de aceleração pode ser decomposto em $a_{xx} = a_x \cos(\phi)$ e $a_{xy} = a_x \sin(\phi)$.

Figura 19 – Decomposição do vetor aceleração



Fonte: Elaborado pelo autor.

O vetor de velocidade pode ser dividido em seus componentes dos eixos de forma similar como se deu a decomposição do vetor aceleração. O valor dessas velocidades nos eixos X e Y, juntamente com a posição em cada eixo e a angulação θ são as variáveis que formam o estado do passo de predição do modelo de Filtro de Kalman Estendido utilizado.

5.4.1 Modelo de predição do Filtro de Kalman Estendido

A primeira etapa do Filtro de Kalman nesse trabalho foi desenvolver um modelo matemático que busca prever o valor do estado atual, baseado nas observações passadas e nos valores obtidos pelos sensores utilizados. Com isso, como desejamos obter a posição do robô precisamos da coordenada X e da coordenada Y. No entanto, é necessário também sabermos o ângulo de rotação do veículo, para definir sua próxima posição, fundamentando-se na Equação 5.1 e na decomposição dos vetores aceleração e velocidade.

Todos esses valores (x, y, θ) , juntamente com as velocidades exercidas pelo veículo nos eixos X e Y (v_x, v_y) formam o estado desse passo de predição.

Para o cálculo do estado predito $X_{k|k-1}$ são utilizados os valores do estado anterior

X_{k-1} e os dados das leituras da unidade de medição inercial (u_k), que contém os valores da aceleração a_{x_k} e da velocidade angular $\dot{\theta}_k$. A Equação 5.2 abaixo apresenta a matriz com as equações para o cálculo do estado predito.

$$X_{k|k-1} = \begin{pmatrix} x_{k|k-1} \\ y_{k|k-1} \\ v_{x_{k|k-1}} \\ v_{y_{k|k-1}} \\ \theta_{k|k-1} \end{pmatrix} = f(X_{k-1}, u_k) = \begin{bmatrix} x_{k-1} + \left(\frac{1}{2}\Delta t^2 a_{x_k} + v_{x_{k-1}} \Delta t\right) \cos(\theta_{k-1} + \Delta t \dot{\theta}_k) \\ y_{k-1} + \left(\frac{1}{2}\Delta t^2 a_{x_k} + v_{y_{k-1}} \Delta t\right) \sin(\theta_{k-1} + \Delta t \dot{\theta}_k) \\ v_{x_{k-1}} + \Delta t a_{x_k} \cos(\theta_{k-1} + \Delta t \dot{\theta}_k) \\ v_{y_{k-1}} + \Delta t a_{x_k} \sin(\theta_{k-1} + \Delta t \dot{\theta}_k) \\ \theta_{k-1} + \Delta t \dot{\theta}_k \end{bmatrix} \quad (5.2)$$

Além do valor do estado, nesse passo também é necessário calcular a covariância do estado a priori $P_{k|k-1}$, utilizando a Equação 2.3 e substituindo a matriz de transição de estado e a matriz de mapeamento de covariância para estado pelas matrizes com os jacobianos da função não linear ($f(X_{k-1}, u_k)$) que caracteriza o movimento do veículo.

Os valores das matrizes com os jacobianos da função f em relação X_{k-1} e em relação a u_k são apresentados nas Equações 5.3, 5.4 e 5.5 e os seus valores são empregados no cálculo da covariância $P_{k|k-1}$, demonstrado na Equação 5.6. Para simplificar as equações, iremos definir a expressão $(\theta_{k-1} + \Delta t \dot{\theta}_k)$ como α .

$$F_{p_k} = \begin{bmatrix} \frac{\partial f(X_{k-1}, u_k)}{\partial x} & \frac{\partial f(X_{k-1}, u_k)}{\partial y} & \frac{\partial f(X_{k-1}, u_k)}{\partial v_x} & \frac{\partial f(X_{k-1}, u_k)}{\partial v_y} & \frac{\partial f(X_{k-1}, u_k)}{\partial \theta} \end{bmatrix} \quad (5.3)$$

$$F_{p_k} = \begin{bmatrix} 1 & 0 & \Delta t \cos(\alpha) & 0 & -\left(\frac{1}{2}\Delta t^2 a_{x_k} + v_{x_{k-1}} \Delta t\right) \sin(\alpha) \\ 0 & 1 & 0 & \Delta t \sin(\alpha) & \left(\frac{1}{2}\Delta t^2 a_{x_k} + v_{y_{k-1}} \Delta t\right) \cos(\alpha) \\ 0 & 0 & 1 & 0 & -\Delta t a_{x_k} \sin(\alpha) \\ 0 & 0 & 0 & 1 & \Delta t a_{x_k} \cos(\alpha) \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

$$F_{u_k} = \begin{bmatrix} \frac{\partial f(X_{k-1}, u_k)}{\partial a_{x_k}} & \frac{\partial f(X_{k-1}, u_k)}{\partial \dot{\theta}_k} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \Delta t^2 \cos(\alpha) & -\left(\frac{1}{2} \Delta t^3 a_{x_k} + v_{x_{k-1}} \Delta t^2\right) \sin(\alpha) \\ \frac{1}{2} \Delta t^2 \sin(\alpha) & \left(\frac{1}{2} \Delta t^3 a_{x_k} + v_{y_{k-1}} \Delta t^2\right) \cos(\alpha) \\ \Delta t \cos(\alpha) & -\Delta t a_{x_k} \sin(\alpha) \\ \Delta t \sin(\alpha) & \Delta t a_{x_k} \cos(\alpha) \\ 0 & \Delta t \end{bmatrix} \quad (5.5)$$

$$P_{k|k-1} = F_{p_k} \cdot P_{k-1} \cdot F_{p_k}^T + F_{u_k} \cdot Q \cdot F_{u_k}^T \quad (5.6)$$

Com os valores do estado predito e da covariância do estado a priori determinados, podemos empregá-los no passo posterior e calcular a etapa de atualização do Filtro de Kalman Estendido.

5.4.2 Modelo de atualização do Filtro de Kalman Estendido

No modelo matemático para atualização e correção dos valores do estado, foi primeiramente definido uma função $h(x, y, \theta)$ que realiza o mapeamento dos valores da posição e do ângulo com as distâncias mensuradas pelos três sensores. Como o veículo foi testado em um ambiente *indoor* controlado, os valores dos limites de extensão do ambiente para os eixos X e Y são conhecidos e portanto, foram empregados para auxiliar na formação da função h .

Observando a Figura 18 é possível perceber que para calcular a distância correta, se faz necessário que o raio transmitido pelo sensor seja refletido em uma das extremidades do ambiente. Além disso, quando isso ocorre, os valores de distância podem ser determinados utilizando-se de equações de retas dos sensores que se interceptam com alguma das retas dos limites do cenário. Assim sendo, para cada medida dF , dR e dL , existem quatro funções h referentes a interseção das retas de cada um desses sensores com aquelas que delimitam o ambiente.

Considerando a característica apresentada, relacionada à interseção das retas e o ângulo θ que especifica a rotação do veículo, foi possível determinar cada uma das funções para modelar as distâncias dos sensores laterais e do sensor frontal. O valor de θ também foi crucial, para definir os momentos em que cada função deve ser empregada, pois, a partir dele é possível identificar de modo aproximado, quando cada uma das retas para dF , dR e dL está interceptando cada reta das extremidades do ambiente $x = 0$, $x = S_x$, $y = 0$, $y = S_y$, onde S_x e S_y correspondem as medidas de extremidade do cenário para os eixos X e Y respectivamente.

A descontinuidade das funções h depende da posição do veículo e do ângulo de orientação, assim para determinar a interceptação das retas dos sensores com as retas das extremidades do ambiente é necessário observar os valores da posição (x_p, y_p) e do ângulo θ .

Na determinação das interceptações entre as retas que formam o ambiente e as que representam as medidas dos sensores, foi definido que o ângulo θ inicia-se com valor 0, com

a frente do veículo apontada para a reta $y = S_y$ e cresce a medida que o veículo rotaciona no sentido horário.

As funções h encontradas para cada medida de distância observada e em cada caso que ela deve ser aplicada são apresentadas nas Equações 5.7, 5.8 e 5.9. Os valores de x_p e y_p representam as coordenadas do veículo. As constantes $Q1$, $Q2$, $Q3$ e $Q4$ representam os conjuntos $\{0 \leq \theta < \frac{\pi}{2}\}$, $\{\frac{\pi}{2} \leq \theta < \pi\}$, $\{\pi \leq \theta < \frac{3\pi}{2}\}$ e $\{\frac{3\pi}{2} \leq \theta < 2\pi\}$ respectivamente.

$$h_{dF}(x_p, y_p, \theta) = \begin{cases} \frac{|S_y - y_p|}{|\cos(\theta)|} & \left\{ Q4 \cap \left\{ \frac{y_p - S_y}{x_p} > \cot(\theta) \right\} \right\} \cup \left\{ Q1 \cap \left\{ \tan(\theta) \leq \frac{S_x - x_p}{S_y - y_p} \right\} \right\} \\ \frac{|S_x - x_p|}{|\sin(\theta)|} & \left\{ Q1 \cap \left\{ \tan(\theta) > \frac{S_x - x_p}{S_y - y_p} \right\} \right\} \cup \left\{ Q2 \cap \left\{ \frac{y_p}{x_p - S_x} \leq \cot(\theta) \right\} \right\} \\ \frac{y_p}{|\cos(\theta)|} & \left\{ Q2 \cap \left\{ \frac{y_p}{x_p - S_x} > \cot(\theta) \right\} \right\} \cup \left\{ Q3 \cap \left\{ \tan(\theta) \leq \frac{x_p}{y_p} \right\} \right\} \\ \frac{x_p}{|\sin(\theta)|} & \left\{ Q3 \cap \left\{ \tan(\theta) > \frac{x_p}{y_p} \right\} \right\} \cup \left\{ Q4 \cap \left\{ \frac{y_p - S_y}{x_p} \leq \cot(\theta) \right\} \right\} \end{cases} \quad (5.7)$$

$$h_{dR}(x_p, y_p, \theta) = \begin{cases} \frac{|S_x - x_p|}{|\cos(\theta)|} & \left\{ Q4 \cap \left\{ \frac{x_p - S_x}{S_y - y_p} > \cot(\theta) \right\} \right\} \cup \left\{ Q1 \cap \left\{ \tan(\theta) \leq \frac{y_p}{S_x - x_p} \right\} \right\} \\ \frac{y_p}{|\sin(\theta)|} & \left\{ Q1 \cap \left\{ \tan(\theta) > \frac{y_p}{S_x - x_p} \right\} \right\} \cup \left\{ Q2 \cap \left\{ (-\cot(\theta)) \leq \frac{x_p}{y_p} \right\} \right\} \\ \frac{x_p}{|\cos(\theta)|} & \left\{ Q2 \cap \left\{ (-\cot(\theta)) > \frac{x_p}{y_p} \right\} \right\} \cup \left\{ Q3 \cap \left\{ \tan(\theta) \leq \frac{S_y - y_p}{x_p} \right\} \right\} \\ \frac{|S_y - y_p|}{|\sin(\theta)|} & \left\{ Q3 \cap \left\{ \tan(\theta) > \frac{S_y - y_p}{x_p} \right\} \right\} \cup \left\{ Q4 \cap \left\{ \frac{x_p - S_x}{S_y - y_p} \leq \cot(\theta) \right\} \right\} \end{cases} \quad (5.8)$$

$$h_{dL}(x_p, y_p, \theta) = \begin{cases} \frac{x_p}{|\cos(\theta)|} & \left\{ Q4 \cap \left\{ (-\cot(\theta)) > \frac{x_p}{y_p} \right\} \right\} \cup \left\{ Q1 \cap \left\{ \tan(\theta) \leq \frac{S_y - y_p}{x_p} \right\} \right\} \\ \frac{|S_y - y_p|}{|\sin(\theta)|} & \left\{ Q1 \cap \left\{ \tan(\theta) > \frac{S_y - y_p}{x_p} \right\} \right\} \cup \left\{ Q2 \cap \left\{ \frac{x_p - S_x}{S_y - y_p} \leq \cot(\theta) \right\} \right\} \\ \frac{|S_x - x_p|}{|\cos(\theta)|} & \left\{ Q2 \cap \left\{ \frac{x_p - S_x}{S_y - y_p} > \cot(\theta) \right\} \right\} \cup \left\{ Q3 \cap \left\{ \tan(\theta) \leq \frac{y_p}{S_x - x_p} \right\} \right\} \\ \frac{y_p}{|\sin(\theta)|} & \left\{ Q3 \cap \left\{ \tan(\theta) > \frac{y_p}{S_x - x_p} \right\} \right\} \cup \left\{ Q4 \cap \left\{ (-\cot(\theta)) \leq \frac{x_p}{y_p} \right\} \right\} \end{cases} \quad (5.9)$$

A partir das funções encontradas para modelar os valores de distância entre o veículo e os limites do ambiente, foi definida uma matriz H , com o jacobiano da função h . Como no caso, existem quatro configurações diferentes para a função h de cada sensor, foi necessário encontrar as derivadas parciais para cada uma delas e utilizá-las na formação de quatro matrizes H distintas, sendo novamente escolhidas de acordo com o valor do ângulo de rotação θ do veículo.

Também foi necessário estabelecer uma matriz com as covariâncias dos sensores empregados. Esta juntamente com a matriz H e os valores do estado e da covariância à priori obtidos na etapa de predição do KF, foram aplicados nas equações da etapa de atualização do KF, bastante similares as equações 2.4, 2.5 e 2.6 que foram apresentadas na Seção 2.5. Como resultado final desse processo, temos o estado X com os valores de posição, velocidade e ângulo atualizados e a matriz P de covariância a posteriori desse mesmo estado.

5.5 O sistema de localização

Com o modelo matemático para a movimentação do veículo formulado, foi desenvolvido um programa na linguagem Python para implementar as equações necessárias e estabelecidas para as etapas de predição e atualização do Filtro de Kalman. Dessa forma, podemos encontrar uma estimativa da real posição do veículo em meio ao ambiente.

A linguagem de programação Python foi escolhida, devido às bibliotecas poderosas que permitem realizar as operações com matrizes necessárias nas etapas do KF, como multiplicação, adição, cálculo da inversa e da transposta de matrizes e entre outros cálculos, de forma simples, rápida e com uma boa precisão.

O sistema de localização possui um módulo para aquisição dos dados obtidos pelos sensores do sistema embarcado. Para isso, o sistema se conecta ao *broker* MQTT empregado neste trabalho e realiza a inscrição nos tópicos onde os dados de aceleração, velocidade e distâncias são publicados. Desse modo, sempre que o sistema embarcado mensurar alguma dessas grandezas através de seus sensores e enviar para o respectivo tópico, o valor será adquirido pelo sistema de localização.

Os valores auferidos dos tópicos, são tratados e manipulados no sistema de localização, para serem armazenados em arquivos de texto. Os dados registrados nesses arquivos são posteriormente aplicados de forma *offline* no procedimento desenvolvido, que implementa às duas etapas do Filtro de Kalman Estendido. As constantes e matrizes necessárias para os cálculos em cada uma das etapas EKF, são definidas e formadas nesta implementação.

No sistema de localização, a escolha da função h referente a cada medida dos sensores de distância foi simplificada. Para isso, os testes que foram realizados com o veículo, ocorreram sempre movimentações paralelas à um dos eixos X ou Y , sendo assim possível ajustar a verificação de qual função h deve ser utilizada para empregar somente o ângulo θ , analisando-se apenas o intervalo em que este valor está inserido.

Ao final de uma interação do filtro, as variáveis com o estado atualizado e a covariância à posteriori são retornados e salvos para serem utilizados na próxima interação dessa função, com os novos valores dos sensores.

Os códigos-fontes do sistema de localização, para realizar a aquisição dos dados via protocolo MQTT e o procedimento que implementa as etapas do Filtro de Kalman Estendido são apresentados no Apêndice A.

6 EXPERIMENTOS E RESULTADOS

Neste trabalho foram realizados experimentos para verificar o funcionamento do sistema de localização implementado sobre diferentes configurações da posição e movimentação do protótipo do veículo desenvolvido. O principal propósito dos experimentos foi averiguar se os valores de posição calculados pelo EKF a partir dos dados obtidos dos sensores, condizem com as reais posições do veículo ao longo do tempo durante a execução de um movimento perpendicular a um dos eixos cartesianos. Além disso, os experimentos também serviram para observar o funcionamento do aplicativo controlador do veículo, no que diz respeito a atrasos dos sinais propagados e a correta execução dos movimentos requisitados.

6.1 Configuração do cenário

Para ser possível a realização dos experimentos, foi criado um cenário no formato quadrangular, com 1 metro de largura e 1 metro de comprimento. Esse espaço foi utilizado como o local onde o veículo deve executar seus movimentos e para formá-lo foram utilizadas algumas caixas de papelão. Os valores de 1 metro em ambos os lados, foram escolhidos por estarem no intervalo limite de medição dos sensores de distância VL53L0X, garantindo dessa forma, que os raios de luz emitidos não ser refletidos em algum anteparo do cenário.

6.2 Definição da variação de tempo

Antes do início da realização dos experimentos, foi verificado a taxa de envio das mensagens publicadas pelo sistema embarcado com as aferições dos sensores. Para isso, o dado do sensor foi emitido juntamente com um *timestamp* do momento atual em que essa informação foi transmitida. Os valores recebidos no sistema de localização foram tratados e a diferença de tempo entre as mensagens foi observada para cada um dos tópicos MQTT utilizados. Analisando-se a diferença de tempo entre mensagens subsequentes, foi constatado que o tempo médio entre a aquisição dos dados no sistema de localização é de 0,25 segundos. A identificação desse valor é importante, pois corresponde a variação de tempo empregada na entrada no algoritmo do Filtro de Kalman.

6.3 Descrição dos experimentos

Para testar o comportamento do sistema de localização desenvolvido para um ambiente *indoor*, foram realizados experimentos sobre a movimentação do veículo dentro do cenário construído. Resumidamente, em cada experimento o veículo era posto em uma posição inicial (X, Y) , com a frente estabelecida para um determinado sentido do plano cartesiano, cuja representação era definida pelo ângulo θ do estado inicial. Para simplificar a análise, o veículo foi posicionado com uma angulação de 0, 90, 180 e 270 graus, representando respectivamente as movimentações nos sentidos $+Y$, $+X$, $-Y$ e $-X$.

Em cada um dos experimentos realizados, o veículo foi estabelecido na sua posição inicial escolhida, e em seguida era submetido a movimentações para frente, com um pequeno intervalo entre estes movimentos. Em média cada experimento desse tipo, durou cerca de 20 segundos.

Para saber a posição aproximada real do veículo, uma fita métrica foi utilizada, e os valores de posição (x, y) onde o veículo era posicionado após cada movimentação realizada, foram anotados para serem empregados posteriormente de forma comparativa com as saídas obtidas do KF.

As informações de distância, aceleração e velocidade angular mensurados pelos sensores do veículo foram transmitidos para o sistema de localização através do protocolo MQTT. A etapa de aquisição de dados nesse sistema, recebeu manipulou e armazenou os valores recebidos em arquivos de texto. O filtro foi aplicado de forma *offline* sobre os valores registrados nos arquivos e as posições obtidas a cada iteração do algoritmo foi comparada com as medidas de posição anotadas durante o experimento. O Quadro 2 apresenta uma descrição detalhada sobre cada experimento. Os valores dos pontos observados foram mensurados em metros.

Quadro 2 – Descrição dos Experimentos

Experimento	Sentido da movimentação	Pontos observados (x, y)
Experimento 1	$+Y$	(0,2; 0,2) ; (0,2; 0,6) ; (0,2; 0,95)
Experimento 2	$+X$	(0,3; 0,1) ; (0,6; 0,1) ; (0,95; 0,1)
Experimento 3	$-Y$	(0,1; 0,8) ; (0,1; 0,45) ; (0,1; 0,08)
Experimento 4	$-X$	(0,75; 0,1) ; (0,4; 0,1) ; (0,07; 0,1)
Experimento 5	$+Y$	(0,9; 0,3) ; (0,9; 0,65) ; (0,9; 0,95)
Experimento 6	$+X$	(0,3; 0,9) ; (0,65; 0,9) ; (0,95; 0,9)
Experimento 7	$-Y$	(0,9; 0,8) ; (0,9; 0,5) ; (0,9; 0,1)
Experimento 8	$-X$	(0,8; 0,9) ; (0,4; 0,9) ; (0,08; 0,9)

Fonte: Elaborado pelo autor.

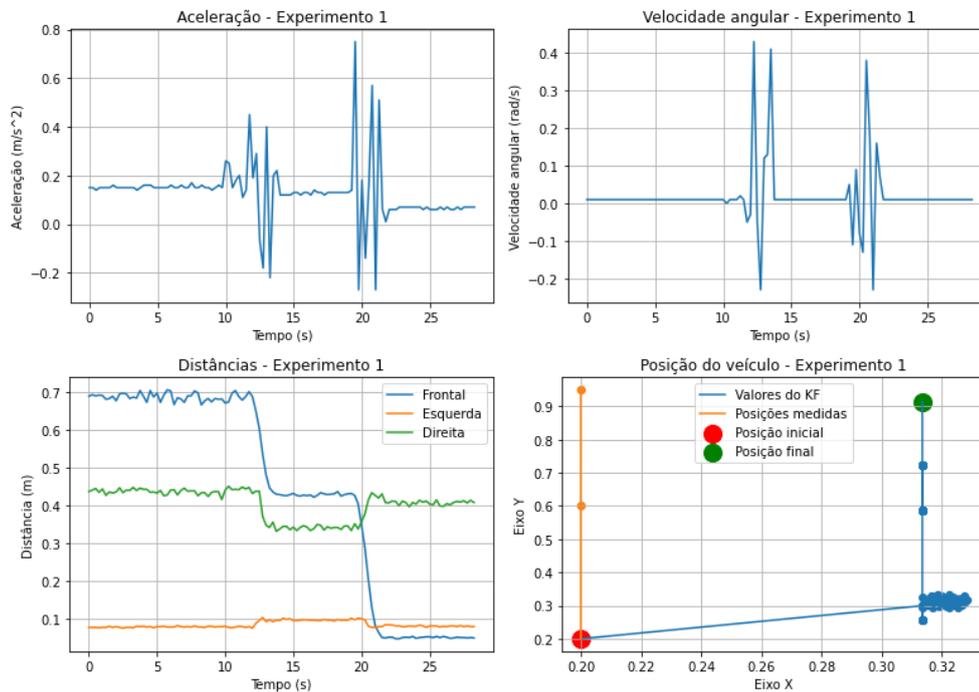
6.4 Resultados dos experimentos

Para observar os resultados obtidos pelo sistema de localização, em cada experimento realizado foi gerado um gráfico com os pontos medidos, descrevendo o real caminho realizado pelo veículo e também com os valores de posição alcançados pelo Filtro de Kalman. Além disso, também foi verificado a evolução das medidas realizadas pelos sensores ao longo do tempo, para que assim fosse possível identificar *outliers* entre estes valores e eliminá-los para não prejudicar nos cálculos do sistema de localização.

6.4.1 Experimento 1

No experimento 1, o veículo se locomoveu positivamente no eixo Y, enquanto sua posição no eixo X permaneceu inalterada, como foi descrito no Quadro 2. Os resultados deste experimento são apresentados na Figura 20.

Figura 20 – Resultados do experimento 1



Fonte: Elaborado pelo autor.

Observando os gráficos da aceleração e distâncias ao longo do tempo de execução do experimento 1, podemos notar a sincronia entre estas informações. Nos momentos em que a ação de locomoção do veículo ocorreu, os valores da aceleração sofreram alterações, e, ao mesmo tempo, a distância frontal medida pelo sensor foi reduzida. Apesar de o veículo não ter executado

nenhuma curva, os valores de velocidade angular também oscilaram durante o movimento do veículo, devido a imperfeições do próprio movimento e possíveis ruídos capturados pelo sensor.

Ainda observando o gráfico da aceleração da Figura 20, podemos reparar mais uma característica. Nos momentos em que o movimento ocorre, é possível notar primeiramente uma oscilação positiva da aceleração, seguida de uma variação negativa dessa grandeza. Tal fato, representa bem a movimentação realizada pelo veículo, pois a aceleração crescendo indica uma movimentação para frente e em seguida decrescendo representa que o veículo está cessando o movimento, ou seja, está parando. Nos demais momentos a aceleração permanece constante, com valores que idealmente deveriam ser 0, porém devido ao erro de resolução do próprio sensor, estes valores ficaram apenas próximos desse limiar.

No gráfico das distâncias, é possível perceber que como o movimento ocorreu apenas para frente neste experimento, os valores de distância para a esquerda e para a direita não sofreram grandes variações, como se era esperado. Por outro lado, o valor da distância frontal, foi sendo reduzido a cada movimento realizado, saindo de 70 *cm* no início do experimento para valores inferiores à 10 *cm* no final.

Todas essas características apresentadas sobre as medidas de aceleração, velocidade angular e distâncias também foram constatadas nos demais experimentos realizados, alterando-se apenas os valores mensurados das distâncias laterais que dependiam da posição onde o veículo era colocado.

Em relação ao valor da posição obtida pelo sistema de localização, podemos observar no gráfico de posição do veículo (localizado na parte inferior direita da Figura 20) que os valores obtidos pelo filtro representaram bem a movimentação exercida pelo veículo. No entanto, um erro próximo de 10 *cm* sobre a posição X, nas primeiras interações do algoritmo distanciou o valor da posição final obtida pelo sistema nesse eixo, do valor real medido durante a execução do experimento.

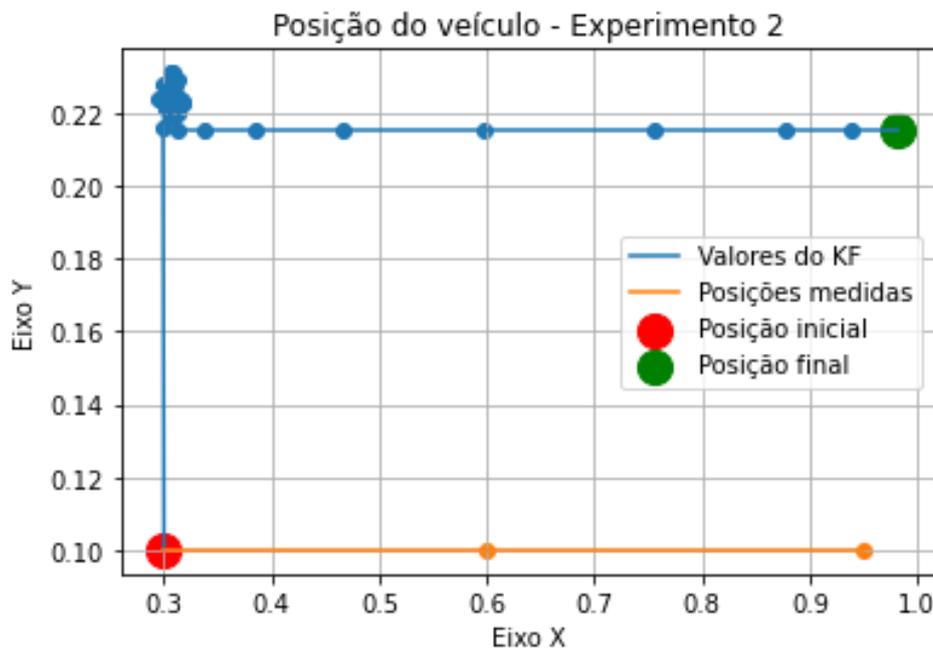
Apesar do erro apresentado no eixo X, é possível notar a evolução da execução do movimento no eixo Y e como esta culminou em um valor bem próximo do real, com uma medida de 0,915 *m* nesse eixo, quando o último ponto observado continha o valor de 0,95 *m*.

6.4.2 Experimento 2

No segundo experimento realizado, o veículo realizou um movimento positivo no eixo X e manteve sua posição fixa no eixo Y. O resultado do gráfico com as posições medidas e

as posições obtidas pelo sistema de localização são apresentados na Figura 21.

Figura 21 – Resultados do experimento 2



Fonte: Elaborado pelo autor.

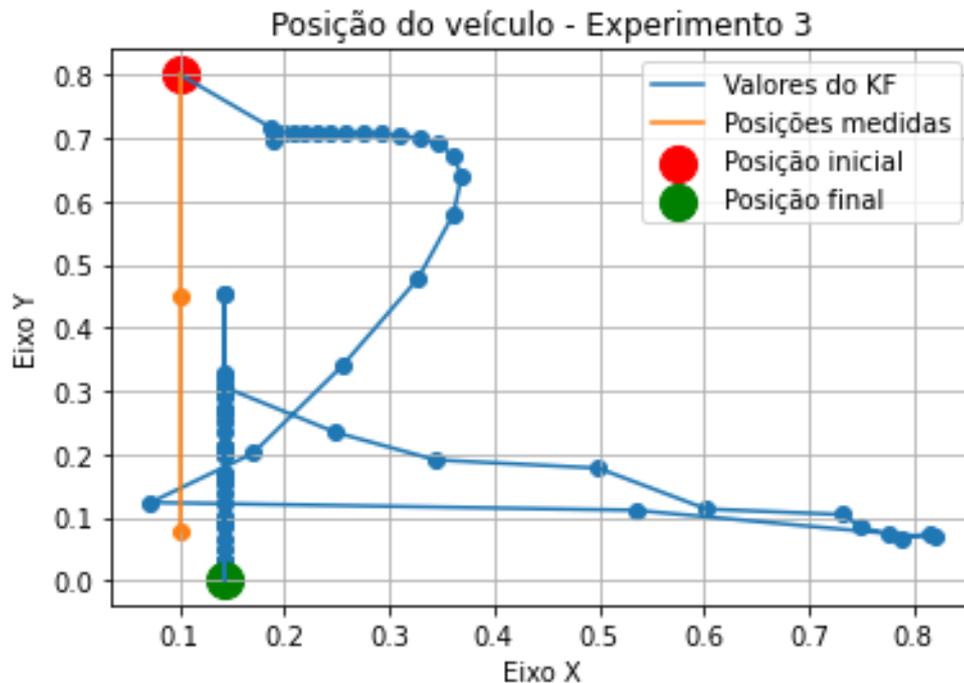
Observando o gráfico de posição da Figura 21 percebe-se que ocorreu um fator semelhante ao do experimento 1. O eixo onde o movimento não deveria ocorrer que para esse caso é o eixo Y, teve um erro de cerca de 10 *cm* no início do experimento e em seguida o movimento ocorreu no eixo X como se era esperado. O valor final da posição no eixo X (0,982 *m*), calculado pelo filtro foi bem próximo do valor real mensurado (0,95 *m*), novamente similar ao que aconteceu no primeiro experimento.

Os resultados do experimento 6 também se assemelham bastante com os dois primeiros experimentos realizados. Nele também é possível observar uma reta perpendicular a reta do movimento real. Porém, para esse experimento 6, o erro no eixo invariável foi bem maior, por volta de 70 *cm*.

6.4.3 Experimento 3

No terceiro experimento o veículo se locomoveu negativamente no eixo Y. O resultado gerado pelos valores do Filtro de Kalman são apresentados na Figura 22

Figura 22 – Resultados do experimento 3



Fonte: Elaborado pelo autor.

Na Figura 22, apesar do ponto final gerado pelo sistema de localização se aproximar bastante do valor real medido, o resultado para este experimento não foi tão satisfatório, pelo fato dos pontos apresentados não representarem o movimento exercido pelo veículo. Como pode-se notar na Figura 22, os pontos formaram uma curva no início e ao final alguns pontos no eixo X se distanciaram bastante dos valores mensurados.

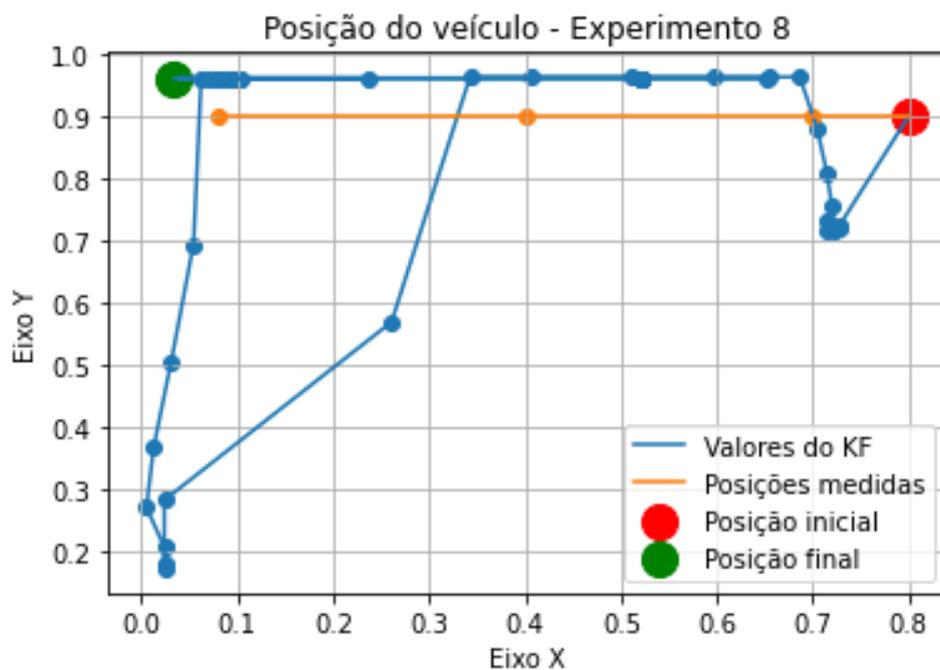
Ainda que o resultado não tenha ocorrido como o desejado, algumas trajetórias incorretas já se eram esperadas devidos as incertezas dos próprios sensores utilizados que podem acabar capturando valores inexatos, prejudicando a atuação do sistema de localização. Vale destacar também, que as matrizes de covariância não são totalmente perfeitas e seus valores foram estimados através de diversas tentativas realizadas e da observação da variância dos dados dos sensores.

A característica observada no resultado do experimento 3, ocorreu parcialmente também nos experimentos 4, 5 e 7. Nestes casos, a trajetória não formou uma curva como no experimento 3, mas acabou variando bastante entre os eixos, construindo movimentos incorretos. Porém, assim como no experimento 3 o valor da posição final também foi bem próximo do valor real.

6.4.4 Experimento 8

No oitavo experimento, o resultado das posições alcançadas pelo sistema de localização descreveram, uma trajetória bem semelhante da que ocorreu com o veículo, como pode ser observado na Figura 23. Entretanto, como pode ser reparado na Figura 23, nos pontos próximos do final do experimento, algumas posições erradas provocaram uma mudança incorreta no movimento. Apesar disso, ao final do experimento, o valor da última posição que foi de $(0,04\text{ m}; 0,96\text{ m})$ ficou bem próxima do valor real observado de $(0,1\text{ m}; 0,9\text{ m})$, com um erro de apenas 6 cm nos dois eixos.

Figura 23 – Resultados do experimento 8



Fonte: Elaborado pelo autor.

Mesmo com um erro ao final do movimento exercido, este experimento se mostrou bem condizente com a resposta esperada do sistema de localização, pois os poucos valores que levaram ao erro na movimentação podem ser removidos através de técnicas bem elaboradas para tratamento de dados e remoção de *outliers*. Além disso, mesmo com o erro nos dados o sistema conseguiu retornar para a posição correta ao final do experimento.

6.5 Análise dos resultados

A comparação e investigação dos resultados obtidos nos experimentos é uma etapa importante na evolução da solução proposta. Para isso, o Quadro 3 foi construído contendo as principais informações adquiridas em cada experimento. Os valores das medidas de posição final do veículo e da posição final do sistema estão na unidade metros (*m*).

Quadro 3 – Resultados dos Experimentos

Experimento / Movimentação	Movimentação representada pelo sistema	Posição final do veículo	Posição final do sistema	Erro absoluto
1 / +Y	Sim	(0,2 ; 0,95)	(0,31 ; 0,915)	X = 0,11 m ; Y = 0,035 m
2 / +X	Sim	(0,95 ; 0,21)	(0,981 ; 0,21)	X = 0,031 m ; Y = 0,11 m
3 / -Y	Não	(0,1 ; 0,08)	(0,14 ; 0,0035)	X = 0,04 m ; Y = 0,0765 m
4 / -X	Não	(0,07 ; 0,1)	(0,007 ; 0,03)	X = 0,063 m ; Y = 0,07 m
5 / +Y	Não	(0,9 ; 0,95)	(0,762 ; 0,975)	X = 0,138 m ; Y = 0,025 m
6 / +X	Sim	(0,95 ; 0,9)	(0,96 ; 0,09)	X = 0,01 m ; Y = 0,81 m
7 / -Y	Não	(0,9 ; 0,1)	(0,952 ; 0,03)	X = 0,052 m ; Y = 0,07 m
8 / -X	Sim	(0,08 ; 0,9)	(0,034 ; 0,96)	X = 0,046 m ; Y = 0,06 m

Fonte: Elaborado pelo autor.

Com base nos experimentos realizados, pode-se constatar que o sistema de localização atuou parcialmente como o desejado. Porém, apesar de toda a modelagem e implementação terem sido realizadas conforme o que foi planejado, o sistema ainda apresenta alguns pontos falhos que necessitam de ajustes.

As informações no Quadro 3 mostram que dos 8 experimentos realizados, em 50% deles, o movimento descrito pelos pontos estimados do sistema de localização corresponderam ao real movimento exercido pelo veículo, enquanto, que nos outros 50%, a variação dos pontos obtidos na saída do KF provocaram trajetórias incorretas no percurso estimado.

Analisando as colunas do Quadro 3 com as posições finais mensuradas e estimadas, identifica-se que os valores finais de posição alcançados pelo sistema de localização se mantiveram aproximadamente perto dos valores reais. Uma exceção a esse fato é o experimento 6, em que o valor final no eixo Y deveria estar próximo de 90 centímetros, e o valor estimado foi de 9 centímetros. Essa característica pode ser melhor observada na última coluna do Quadro 3, onde é apresentado o valor da diferença absoluta entre a posição mensurada e a posição alcançada pelo Filtro de Kalman, para os dois eixos onde o movimento aconteceu.

Observando a coluna com o erro absoluto calculado entre as posições finais, verifica-se que com exceção do caso 6, os erros podem ser considerados baixos. O erro absoluto médio para o eixo X foi 6,125 *cm*, enquanto, que para o eixo Y foi bem maior de 14,8 *cm*, mas muito

devido a um único experimento em que a posição ficou bem errada nesse eixo. Sem esse valor do experimento 6, o erro médio no eixo Y seria de apenas 6,37 *cm*.

Uma peculiaridade interessante de se destacar também, é que a maioria dos valores altos de erros aconteceram no eixo onde o movimento não deveria acontecer, o que pode representar um possível ponto de melhoria a ser realizado no sistema de localização, para evitar movimentações inexistentes devido a valores ruidosos mensurados pelos sensores.

6.6 Experimentos dos comandos do *app*

Com o propósito de avaliar o funcionamento dos comandos emitidos pelo aplicativo e das ações realizadas após cada instrução, uma série de casos de teste foram realizadas no cenário construído. Nesses testes o veículo foi submetido ao controle realizado pelo usuário do aplicativo, que poderia alterar o modo de execução do veículo e emitir os respectivos comandos existentes em cada modo de operação.

No modo de operação manual em que o usuário tem total controle sobre o veículo, e no modo de comandos, em que as ações ocorrem por períodos programados, todas as mensagens emitidas pelo *app* obtiveram êxito na execução da ação correspondente. Além disso, mesmo utilizando da rede para transmitir as mensagens pelo protocolo MQTT não houve atrasos significativos na comunicação entre as partes do sistema, que prejudicassem o funcionamento desejado do veículo nesses modos de operação.

No modo de execução autônomo, o veículo funcionou bem, detectando e evitando a colisão com os obstáculos encontrados. Porém, alguns pontos fora do ângulo de atuação do sensor de distância, fizeram com que em alguns testes o veículo não conseguisse identificar com precisão o obstáculo, levando a execução da ação incorreta. Entretanto, futuramente quando o sistema de localização estiver totalmente finalizado, este será empregado no veículo para auxiliar no correto controle de execução desse modo de operação, garantindo uma maior segurança sobre as devidas ações que necessitam ser realizadas em cada situação.

7 CONCLUSÕES E TRABALHOS FUTUROS

A automação de veículos e robôs é um campo de estudo que vem sendo muito explorado nos últimos anos. Cada vez mais, estes equipamentos estão evoluindo para permitir ao ser humano à exploração de regiões ainda inexploradas, como outros planetas e os lugares mais profundos dos oceanos. Também, é notório como esta evolução está provocando a inserção de novos aparelhos que conseguem realizar de forma autônoma atividades do cotidiano do ser humano, como transportar equipamentos, aspirar o pó de uma casa e até mesmo servir comidas em um estabelecimento. Ainda assim, existem inúmeros desafios na área de robôs móveis e um deles é determinar a sua localização em um ambiente fechado, para deste modo reduzir erros em suas trajetórias e suas funções.

7.1 Considerações Finais

A proposta neste trabalho foi desenvolver um veículo capaz de navegar e se localizar em um ambiente fechado, que pudesse atuar tanto de forma autônoma quanto controlado remotamente. O objetivo foi implementar um equipamento que através das medidas realizadas por seus sensores e de algumas informações do ambiente, conseguisse identificar sua localização. Para isso, foi desenvolvido um protótipo embarcado do veículo com sensores, atuadores e uma placa de desenvolvimento, que pode ser controlado através de um aplicativo *mobile*. O sistema embarcado também se comunica com um sistema de localização que no que lhe concerne, empregou os dados mensurados em um modelo de Filtro de Kalman Estendido para estimar a sua localização ao longo de sua trajetória.

Conforme os resultados obtidos nos experimentos realizados e apresentados, notou-se que o veículo proposto consegue ser perfeitamente controlado através do aplicativo desenvolvido. Porém, o sistema de localização ainda precisa de alguns ajustes para poder representar de forma fidedigna o real movimento exercido pelo veículo, e assim determinar as distâncias percorridas em cada eixo para identificar sua localização no ambiente.

Os resultados e experimentos também evidenciaram que o erro nas medidas de aceleração e velocidade angular adquiridas pela unidade de medição inercial, acabaram prejudicando o correto funcionamento do sistema de localização. Valores de medidas ruidosas que não condiziam com a real situação do veículo influenciaram diretamente na resposta final dos cálculos de posição realizados pelo modelo do EKF. No entanto, a utilização de dispositivos

mais robustos com uma precisão maior nas aferições realizadas e técnicas de manipulação de dados para remover os erros das medidas podem ser aplicadas para melhorar esse aspecto.

Mesmo apresentando alguns pontos falhos e identificando falsas posições em algumas situações, o sistema de localização conseguiu de forma geral atuar bem na identificação da posição do veículo. Com a realização de pequenas melhorias nas coletas das medidas utilizadas e algumas modificações na análise dos dados, o sistema é viável de ser utilizado simultaneamente a movimentação do veículo, para auxiliar no controle do modo de execução autônomo.

7.2 Trabalhos Futuros

Como trabalhos futuros, pretende-se realizar o aprimoramento do sistema de localização e aplicá-lo em novos experimentos onde a movimentação realizada pelo dispositivo seja efetuada nos dois eixos ao mesmo tempo. Além disso, também objetiva-se aplicar novos sensores no dispositivo, que permitirão obter leituras mais verídicas das grandezas físicas envolvidas na movimentação do veículo e que possibilitarão a utilização do dispositivo em cenários maiores.

Em relação à atuação do veículo no modo autônomo, com o aprimoramento do sistema de localização, pretende-se empregar esta parte da solução para identificar a posição do veículo em tempo real e assim auxiliar no controle do dispositivo, permitindo a este determinar com maior exatidão as devidas ações necessárias para diferentes situações.

Com a execução do sistema de localização acontecendo durante a movimentação do veículo, também pretende-se implementar uma nova tela no aplicativo para observar e atualizar a posição do veículo a medida que este realiza seus deslocamentos no ambiente, permitindo ao usuário checar a qualquer instante em que localização o dispositivo está atuando.

REFERÊNCIAS

- BABIUCH, M.; FOLTÝNEK, P.; SMUTNÝ, P. Using the esp32 microcontroller for data processing. In: IEEE. **2019 20th International Carpathian Control Conference (ICCC)**. [S. l.], 2019. p. 1–6.
- BHUYAN, M. **Intelligent Instrumentation: Principles and applications**. [S. l.]: CRC Press, 2010.
- BILGIN, B.; LIANG, J.; TERZIC, M. V.; DONG, J.; RODRIGUEZ, R.; TRICKETT, E.; EMADI, A. Modeling and analysis of electric motors: state-of-the-art review. **IEEE Transactions on Transportation Electrification**, IEEE, v. 5, n. 3, p. 602–617, 2019.
- BILL, L.; SHAHNASSER, H. Development and implementation of an autonomously driven vehicle prototype. In: IEEE. **2019 IEEE 2nd International Conference on Electronics Technology (ICET)**. [S. l.], 2019. p. 310–314.
- BUECHEL, M.; FRTUNIKJ, J.; BECKER, K.; SOMMER, S.; BUCKL, C.; ARMBRUSTER, M.; MAREK, A.; ZIRKLER, A.; KLEIN, C.; KNOLL, A. An automated electric vehicle prototype showing new trends in automotive architectures. In: IEEE. **2015 IEEE 18th International Conference on Intelligent Transportation Systems**. [S. l.], 2015. p. 1274–1279.
- BUTTAZZO, G. C. **Hard real-time computing systems: predictable scheduling algorithms and applications**. [S. l.]: Springer Science & Business Media, 2011. v. 24.
- CHAMORRO, A. M. M.; GONZÁLEZ, D. A. R. **Implementación de un robot de navegación autónoma terrestre para evasión de obstáculos usando el dispositivo Pixhawk**. Quito: Universidad de las Américas, 2019.
- CHEMHENGCHAROEN, P.; NILSUMRIT, P.; PONGPETRARAT, P.; PHANOMCHOENG, G. Development of a prototype of autonomous vehicle for agriculture applications. In: **Proceedings of the 7th International Conference on Communications and Broadband Networking**. [S. l.: s. n.], 2019. p. 53–57.
- CORDEIRO, R. de A. *et al.* **Modelagem e controle de trajetória de um veículo robótico terrestre de exterior**. [S. l.]: UNICAMP, 2013.
- COSTA, J. H. R. **Implementação de Fusão Sensorial para Localização de um Veículo Autônomo**. Tese (Doutorado) – Tese de Mestrado, Universidade Federal de Minas Gerais, 2013.
- CRUZ, S. M. **Implementação de um filtro de Kalman estendido em arquiteturas reconfiguráveis aplicado ao problema de localização de robôs móveis**. [S. l.: s. n.], 2013.
- EL-REFAIE, A. M. Motors/generators for traction/propulsion applications: A review. **IEEE Vehicular Technology Magazine**, IEEE, v. 8, n. 1, p. 90–99, 2013.
- FAUZI, E. M.; ASYIKIN, M. B. Z.; PRASETYA, I. Y. Analisa dan solusi noise sensor vl5310x pada berbagai kondisi cahaya. In: **Prosiding Industrial Research Workshop and National Seminar**. [S. l.: s. n.], 2018. v. 9, p. 483–487.
- FLEMING, W. J. New automotive sensors—a review. **IEEE Sensors Journal**, IEEE, v. 8, n. 11, p. 1900–1921, 2008.

- FORHAN, N. A. E.; SENNA, J. R. S. Giroscópios mems. **urlib. net/sid. inpe. br/mtc-m19**, v. 80, p. 01–25, 2010.
- JUNG, C. R.; OSÓRIO, F. S.; KELBER, C. R.; HEINEN, F. J. Computação embarcada: Projeto e implementação de veículos autônomos inteligentes. **Anais do CSBC**, v. 5, p. 1358–1406, 2005.
- KOVAL, L.; VAŇUŠ, J.; BILÍK, P. Distance measuring by ultrasonic sensor. **IFAC-PapersOnLine**, Elsevier, v. 49, n. 25, p. 153–158, 2016.
- LAKOVIĆ, N.; BRKIĆ, M.; BATINIĆ, B.; BAJIĆ, J.; RAJS, V.; KULUNDŽIĆ, N. Application of low-cost vl5310x tof sensor for robot environment detection. In: IEEE. **2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)**. [S. l.], 2019. p. 1–4.
- LI, R.-J.; LEI, Y.-J.; CHANG, Z.-X.; ZHANG, L.-S.; FAN, K.-C. Development of a high-sensitivity optical accelerometer for low-frequency vibration measurement. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 18, n. 9, p. 2910, 2018.
- LITMAN, T. **Autonomous vehicle implementation predictions**: Implications for transport planning. [S. l.: s. n.], 2020.
- MAIER, A.; SHARP, A.; VAGAPOV, Y. Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things. In: IEEE. **2017 Internet Technologies and Applications (ITA)**. [S. l.], 2017. p. 143–148.
- MAUSBACH, L. B.; BEHRENS, F. H. **Medição indireta de velocidade de motores elétricos de corrente contínua com escova**. [S. l.: s. n.], 2017.
- MENDES, M. de A.; SILVA, I. C. da; RAMIRES, V. V.; REICHERT, F. F.; MARTINS, R. C.; TOMASI, E. Calibration of raw accelerometer data to measure physical activity: a systematic review. **Gait & posture**, Elsevier, v. 61, p. 98–110, 2018.
- MOLINA, C. B. S. T. **Controle veicular autônomo (CVA)**: um sistema para prevenir acidentes no contexto de veículos autônomos. Tese (Doutorado) – Universidade de São Paulo, 2018.
- MORAIS, D.; RODRIGUES, D. L.; POSSANI, D.; CORREIA, F. C. Ondas ultrassônicas: teoria e aplicações industriais em ensaios não-destrutivos. **Revista Brasileira de Física Tecnológica Aplicada**, [S. l.], v. 4, n. 1, p. 16–33, 2017.
- MUSTAPHA, B.; ZAYEGH, A.; BEGG, R. K. Ultrasonic and infrared sensors performance in a wireless obstacle detection system. In: IEEE. **2013 1st International Conference on Artificial Intelligence, Modelling and Simulation**. [S. l.], 2013. p. 487–492.
- OKAMOTO, S.; DENIS, N.; KATO, Y.; IEKI, M.; FUJISAKI, K. Core loss reduction of an interior permanent-magnet synchronous motor using amorphous stator core. **IEEE Transactions on Industry Applications**, IEEE, v. 52, n. 3, p. 2261–2268, 2016.
- QU, J.; ZHANG, Q.; HOU, L.; ZHANG, R.; TING, K. Research on ekf-based localization method of tracked mobile robot. In: ATLANTIS PRESS. **2nd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2017)**. [S. l.], 2016. p. 582–587.
- RODRIGUES, L. C. **Fundamentos, tecnologias e aplicações de veículos autônomos**. Dissertação (B.S. thesis) – Universidade Tecnológica Federal do Paraná, 2017.

SARAVATI. **Kit Chassi 2WD ACRÍLICO Robô Carrinho 2 Rodas ZK-2**. [S. l.], 2022. Disponível em: <https://www.saravati.com.br/kit-chassi-2wd-acrilico-robo-carrinho-2-rodas-zk-2>. Acesso em: 14 jan. 2022.

SCHLISCHTING, M. W. *et al.* **Técnicas de linearização e controle da profundidade de modulação de giroscópios interferométricos a fibra óptica de malha aberta**. [S. l.: s. n.], 2018.

SHINE, N.; WIBISONO, F. A.; ANGGADJAJA, E.; ROCHADIANI, T. Food products monitoring machine using combinations of multiple autonomous sensors. In: IEEE. **2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)**. [S. l.], 2018. p. 152–157.

SONI, D.; MAKWANA, A. A survey on mqtt: a protocol of internet of things (iot). In: **International Conference On Telecommunication, Power Analysis And Computing Techniques (ICTPACT-2017)**. [S. l.: s. n.], 2017. v. 20.

THANGAVEL, D.; MA, X.; VALERA, A.; TAN, H.-X.; TAN, C. K.-Y. Performance evaluation of mqtt and coap via a common middleware. In: IEEE. **2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP)**. [S. l.], 2014. p. 1–6.

THOMAZINI, D.; ALBUQUERQUE, P. U. B. de. **Sensores industriais: fundamentos e aplicações**. [S. l.]: Saraiva Educação SA, 2020.

THRUN, S.; BURGARD, W.; FOX, D. Probabilistic robotics. **Kybernetes**, Emerald Group Publishing Limited. [S. l.], 2006.

VELOSO, P. H. O. **Desenvolvimento de em protótipo para veículos autônomos com localização e mapeamento simultâneos**. [S. l.: s. n.], 2019.

WEI, D. C. M. **Método de desvio de obstáculos aplicado em veículo autônomo**. Tese (Doutorado) – Universidade de São Paulo, 2015.

APÊNDICE A – CÓDIGOS-FONTES DO SISTEMA DE LOCALIZAÇÃO

Código-fonte 1 – Aquisição dos dados via protocolo MQTT

```
1 import random
2 from paho.mqtt import client as mqtt_client
3
4 broker = 'broker.mqtt-dashboard.com'
5 port = 1883
6 client_id = f'python-mqtt-{random.randint(0, 100)}'
7 topicDistanceSensor = "4361fd9a-0c1e-420c-b858-27ad478288e6/
   distanceSensor"
8 topicAccelerometer = "4361fd9a-0c1e-420c-b858-27ad478288e6/
   accelerometer"
9 topicGyroscope = "4361fd9a-0c1e-420c-b858-27ad478288e6/
   gyroscope"
10 distancesData = []
11 accelerationsData = []
12 gyrosData = []
13
14 def handleMessage(msg):
15     data = msg.payload.decode();
16     topic = msg.topic
17     if topic == topicDistanceSensor:
18         handleDistanceValues(data)
19     elif topic == topicAccelerometer:
20         handleAccelerationValues(data)
21     elif topic == topicGyroscope:
22         handleGyroValues(data)
23
24 def handleDistanceValues(data):
25     if data.find('|') != -1:
26         parts = data.replace(' ', '').split('|')
```

```
27     distances = []
28     for element in parts:
29         if element.find(':') != -1:
30             value = int(element.split(':')[1])
31             distances.append(value)
32
33     print(distances)
34     distancesData.append(distances)
35
36 def handleAccelerationValues(data):
37     if data.find(':') != -1:
38         acceleration = float(data.split(':')[1])
39         print(acceleration)
40         accelerationsData.append(acceleration)
41
42 def handleGyroValues(data):
43     if data.find(':') != -1:
44         gyro = float(data.split(':')[1])
45         print(gyro)
46         gyrosData.append(gyro)
47
48 def connectMqtt():
49     def on_connect(client, userdata, flags, rc):
50         if rc == 0:
51             print("Conectado com sucesso ao Broker MQTT!")
52         else:
53             print("Falha ao se conectar, código de retorno
54                 %d\n", rc)
55
56     client = mqtt_client.Client(client_id)
57     client.on_connect = on_connect
58     client.connect(broker, port)
```

```
58     return client
59
60
61 def subscribe(client: mqtt_client):
62     def on_message(client, userdata, msg):
63         print(f"Recebido `{msg.payload.decode()}` do tópicos
64             `{msg.topic}`")
65         handleMessage(msg)
66
67     client.subscribe(topicDistanceSensor)
68     client.subscribe(topicAccelerometer)
69     client.subscribe(topicGyroscope)
70     client.on_message = on_message
71
72 def runMQTT():
73     client = connectMqtt()
74     subscribe(client)
75     client.loop_forever()
76
77 runMQTT()
```

Código-fonte 2 – Determinação da posição com Filtro de Kalman

```
1
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import pandas as pd
5
6 def determinarDF(x, y, theta, Sx, Sy):
7     dF = 0
8     if theta >= 0 and theta < np.pi / 2:
```

```
9         dF = (Sy - y) / np.abs(np.cos(theta))
10     elif theta >= np.pi / 2 and theta < np.pi:
11         dF = (Sx - x) / np.abs(np.sin(theta))
12     elif theta >= np.pi and theta < 3*np.pi/2:
13         dF = (y) / np.abs(np.cos(theta))
14     elif theta >= 3 * np.pi / 2 and theta < 2 * np.pi:
15         dF = (x) / np.abs(np.sin(theta))
16     return dF
17
18 def determinarDL(x, y, theta, Sx, Sy):
19     dL = 0
20     if theta >= 0 and theta < np.pi / 2:
21         dL = (x) / np.abs(np.cos(theta))
22     elif theta >= np.pi / 2 and theta < np.pi:
23         dL = (Sy - y) / np.abs(np.sin(theta))
24     elif theta >= np.pi and theta < 3*np.pi/2:
25         dL = (Sx - x) / np.abs(np.cos(theta))
26     elif theta >= 3 * np.pi / 2 and theta < 2 * np.pi:
27         dL = (y) / np.abs(np.sin(theta))
28     return dL
29
30 def determinarDR(x, y, theta, Sx, Sy):
31     dR = 0
32     if theta >= 0 and theta < np.pi / 2:
33         dR = (Sx - x) / np.abs(np.cos(theta))
34     elif theta >= np.pi / 2 and theta < np.pi:
35         dR = (y) / np.abs(np.sin(theta))
36     elif theta >= np.pi and theta < 3*np.pi/2:
37         dR = (x) / np.abs(np.cos(theta))
38     elif theta >= 3 * np.pi / 2 and theta < 2 * np.pi:
39         dR = (Sy - y) / np.abs(np.sin(theta))
40     return dR
```

```

41
42 def determinarMatrizH(x, y, theta, Sx, Sy):
43     if theta >= 0 and theta < np.pi / 2:
44         H = np.array([
45             [0, -1/np.abs(np.cos(theta)), 0, 0, (((Sy - y)*
46                 np.tan(theta)) / np.abs(np.cos(theta)))],
47             [1/np.abs(np.cos(theta)), 0, 0, 0, (((x)*np.tan(
48                 theta)) / np.abs(np.cos(theta)))],
49             [-1/np.abs(np.cos(theta)), 0, 0, 0, (((Sx - x)*np
50                 .tan(theta)) / np.abs(np.cos(theta)))]
51         ])
52     elif theta >= np.pi / 2 and theta < np.pi:
53         H = np.array([
54             [-1/np.abs(np.sin(theta)), 0, 0, 0, ((-(Sx - x)*
55                 np.cos(theta)) / (np.sin(theta) * np.abs(np.
56                 sin(theta))))],
57             [0, -1/np.abs(np.sin(theta)), 0, 0, ((-(Sy - y)*
58                 np.cos(theta)) / (np.sin(theta) * np.abs(np.
59                 sin(theta))))],
60             [0, 1/np.abs(np.sin(theta)), 0, 0, ((-(y)*np.
61                 cos(theta)) / (np.sin(theta) * np.abs(np.sin(
62                 theta))))]
63         ])
64     elif theta >= np.pi and theta < 3*np.pi/2:
65         H = np.array([
66             [0, 1/np.abs(np.cos(theta)), 0, 0, (((y)*np.tan(
67                 theta)) / np.abs(np.cos(theta)))],
68             [-1/np.abs(np.cos(theta)), 0, 0, 0, (((Sx - x)*
69                 np.tan(theta)) / np.abs(np.cos(theta)))]],
70             [1/np.abs(np.cos(theta)), 0, 0, 0, (((x)*np.tan(
71                 theta)) / np.abs(np.cos(theta)))]
72         ])

```

```

61 elif theta >= 3 * np.pi / 2 and theta < 2 * np.pi:
62     H = np.array([
63         [1/np.abs(np.sin(theta)), 0, 0, 0, ((-(x)*np.cos(
        theta)) / (np.sin(theta) * np.abs(np.sin(theta)
        )))),
64         [0, 1/np.abs(np.sin(theta)), 0, 0, ((-(y)*np.cos(
        theta)) / (np.sin(theta) * np.abs(np.sin(theta)
        )))),
65         [0, -1/np.abs(np.sin(theta)), 0, 0, ((-(Sy - y)*
        np.cos(theta)) / (np.sin(theta) * np.abs(np.
        sin(theta)))]
66     ])
67     return H
68
69 def handlePosicao(posicao_atual):
70     posicao = []
71     for elem in posicao_atual:
72         aux = elem[0, 0]
73         posicao.append(aux)
74     return posicao
75
76 def filtroDeKalman(posicao_anterior, deltaT, u_k, P,
77     distancias_lidas):
78     x_kmenos1, y_kmenos1, vx_kmenos1, vy_kmenos1,
79     theta_kmenos1 = posicao_anterior[0],
80     posicao_anterior[1], posicao_anterior[2],
81     posicao_anterior[3], posicao_anterior[4]
82     aceleracao_medida, velocidade_angular = u_k[0], u_k[1]
83     alpha = theta_kmenos1 + (deltaT*velocidade_angular)
84     deltaX = (1/2 * (deltaT ** 2) * aceleracao_medida +
85         vx_kmenos1*deltaT) * np.cos(alpha)
86     deltaY = (1/2 * (deltaT ** 2) * aceleracao_medida +

```

```

    vy_kmenos1*deltaT) * np.sin(alpha)
82  deltaVx = aceleracao_medida * deltaT * np.cos(alpha)
83  deltaVy = aceleracao_medida * deltaT * np.sin(alpha)
84  deltaTheta = deltaT * velocidade_angular
85  posicao_predita = np.array([posicao_anterior]).
    transpose() + np.array([[deltaX], [deltaY], [deltaVx
    ], [deltaVy], [deltaTheta]])
86  F_p = np.array([
87      [1, 0, deltaT * np.cos(alpha), 0, ((-1/2 * (
          deltaT ** 2) * aceleracao_medida -
          vx_kmenos1 * deltaT) * np.sin(alpha))],
88      [0, 1, 0, deltaT * np.sin(alpha), ((1/2 * (
          deltaT ** 2) * aceleracao_medida +
          vy_kmenos1 * deltaT) * np.cos(alpha))],
89      [0, 0, 1, 0, (-1)*(aceleracao_medida * deltaT
          * np.sin(alpha))],
90      [0, 0, 0, 1, (aceleracao_medida * deltaT * np.
          cos(alpha))],
91      [0, 0, 0, 0, 1]
92  ])
93  F_u = np.array([
94      [1/2 * (deltaT ** 2) * np.cos(alpha),
95      ((-1/2 * (deltaT ** 3) * aceleracao_medida)
          - (vx_kmenos1 * (deltaT ** 2))) * np.
          sin(alpha)],
96      [1/2 * (deltaT ** 2) * np.sin(alpha),
97      ((1/2 * (deltaT ** 3) * aceleracao_medida)
          + (vy_kmenos1 * (deltaT ** 2))) * np.
          cos(alpha)],
98      [deltaT * np.cos(alpha), (-1)*(
          aceleracao_medida * deltaT * np.sin(alpha))
    ],

```

```

99         [deltaT * np.sin(alpha), (aceleracao_medida *
100             deltaT * np.cos(alpha))],
101     ])
102 Q = np.array([[0.00000784, 0],
103               [0, 0.00000038551681]])
104 P_predito = (F_p.dot(P)).dot(F_p.transpose()) + (F_u.
105             dot(Q)).dot(F_u.transpose())
106 Sx, Sy = 1, 1
107 theta = posicao_predita[4][0]
108 if theta < 0:
109     theta = theta + (np.ceil(np.abs(theta) / (2*np.pi))
110                     * 2*np.pi)
111 if theta > 2*np.pi:
112     theta = theta - (np.floor(theta / (2*np.pi)) * 2*np
113                     .pi)
114
115 hDF = determinarDF(posicao_predita[0][0],
116                   posicao_predita[1][0], theta, Sx, Sy)
117 hDR = determinarDR(posicao_predita[0][0],
118                   posicao_predita[1][0], theta, Sx, Sy)
119 hDL = determinarDL(posicao_predita[0][0],
120                   posicao_predita[1][0], theta, Sx, Sy)
121 funcaoH = [hDF, hDL, hDR]
122 Vk = np.array([distancias_lidas]).transpose() - np.
123             array([funcaoH]).transpose()
124 Hm = determinarMatrizH(posicao_predita[0][0],
125                       posicao_predita[1][0], theta, Sx, Sy)
126 Rk = np.array([[0.01, 0, 0],
127               [0, 0.01, 0],
128               [0, 0, 0.01]])
129 Sk = (Hm.dot(P_predito)).dot(Hm.transpose()) + Rk

```

```
122     if np.linalg.det(Sk) == 0:
123         novoSk = Sk + np.array([[0.01, 0, 0],
124                                 [0, 0.01, 0],
125                                 [0, 0, 0.01]])
126         inversoSk = np.linalg.inv(novoSk)
127     else:
128         inversoSk = np.linalg.inv(Sk)
129     Kk = (P_predito.dot(Hm.transpose())).dot(inversoSk)
130     posicao_atual = posicao_predita + (Kk.dot(Vk))
131     P_atual = P_predito+((Kk.dot(Sk)).dot(Kk.transpose()))
132     novaPosicao = handlePosicao(posicao_atual)
133     if novaPosicao[0] > S_x or novaPosicao[0] < 0:
134         novaPosicao[0] = x_kmenos1
135     if novaPosicao[1] > S_y or novaPosicao[1] < 0:
136         novaPosicao[1] = y_kmenos1
137     if novaPosicao[4] > 2*np.pi or novaPosicao[4] < 0:
138         novaPosicao[4] = theta_kmenos1
139     return novaPosicao, P_atual
```