

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)
CURSO DE ENGENHARIA DE COMPUTAÇÃO

LUCAS DE LIMA DA SILVA
RAPHAEL FELIX

MARVIN
TARGET COMPETITION ROBOT

OFICINA DE INTEGRAÇÃO 2 – RELATÓRIO FINAL

CURITIBA

2022

**LUCAS DE LIMA DA SILVA
RAPHAEL FELIX**

**MARVIN
TARGET COMPETITION ROBOT**

Relatório Final da disciplina Oficinas de Integração 2, do curso de Engenharia de Computação, apresentado aos professores que ministram a mesma na Universidade Tecnológica Federal do Paraná como requisito parcial para aprovação na disciplina.

Orientador: Prof. Dr. César M. Vargas Benítez
Prof. Dr. Heitor Silvério Lopes

CURITIBA

2022

RESUMO

. MARVIN

TARGET COMPETITION ROBOT. 46 f. Oficina de Integração 2 – Relatório Final – Curso de Engenharia de Computação, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2022.

Este trabalho apresenta de forma detalhada todas as etapas que envolvem o processo construtivo de um robô autônomo da categoria Trekking, desde seu planejamento até sua execução. Ainda neste, constam os resultados obtidos nos testes realizados e também o detalhamento da solução encontrada para o desenvolvimento da interface usuário-máquina, via página web. Através da interface, podem ser inseridas as coordenadas aproximadas do alvo, definir a ordem de busca e preferência.

Palavras-chave: Trekking, Autônomo, Página Web

LISTA DE FIGURAS

FIGURA 1	– Funcionamento do Sensor HR-SC04	14
FIGURA 2	– Precisão de alcance do VL53L0X	14
FIGURA 3	– Detalhes de desempenho típico do VL53L0X	15
FIGURA 4	– Ponte H modelo VNH2SP30-E	16
FIGURA 5	– Motor DC	17
FIGURA 6	– Diagramas de Blocos do Sistema.	19
FIGURA 7	– Primeiro Andar Projeto Marvin em SolidWorks.	21
FIGURA 8	– Segundo Andar Projeto Marvin em SolidWorks.	22
FIGURA 9	– Motores conectados aos eixos.	22
FIGURA 10	– Esquemático do projeto eletrônico.	25
FIGURA 11	– Pré-visualização 3D da placa.	25
FIGURA 12	– Roteamento da placa.	26
FIGURA 13	– Resultado final da placa.	26
FIGURA 14	– Diagrama de Casos de Uso.	28
FIGURA 15	– Diagrama de Casos de Uso 1.	29
FIGURA 16	– Diagrama de Casos de Uso 2.	29
FIGURA 17	– Diagrama de Casos de Uso 3.	30
FIGURA 18	– Diagrama de Casos de Uso 4.	31
FIGURA 19	– Diagrama de Sequência para os Casos de Uso UC01, UC02 e UC03.	32
FIGURA 20	– Diagrama de Sequência para o Casos de Uso UC04.	33
FIGURA 21	– Diagrama de Estados.	34
FIGURA 22	– Protótipo da Página Web.	35
FIGURA 23	– Integração Mecânica com Hardware.	36
FIGURA 24	– Cronograma do Projeto Parte 1.	39
FIGURA 25	– Cronograma do Projeto Parte 2.	40
FIGURA 26	– Cronograma do Projeto Parte 3.	41
FIGURA 27	– Cronograma do Projeto Parte 4.	41

LISTA DE TABELAS

TABELA 1	– Tabela de Preços dos Componentes	42
----------	--	----

LISTA DE SIGLAS

ROS	Robot Operating System
Wi-Fi	Wireless Fidelity
LPDDR4	Low Power Double Data Rate 4
USB	Universal Serial Bus
GPIO	General Purpose IO
I2C	Inter-Integrated Circuit
UART	Universal Asynchronous Receiver/Transmitter
SPI	Serial Protocol Interface
PWM	Pulse-Width Modulation
SoC	System-on-a-chip
TSMC	Taiwan Semiconductor Manufacturing Company
ToF	Time-of-Fight
mm	Milímetro
Hz	Hertz
m	Metro
VCSEL	Vertical Cavity Surface-Emitting Laser
nm	Nanômetro
dps	Degrees per second
m/s ²	Metros por segundo ao quadrado
μT	Microtesla
A	Amperes
ORM	Object Relational Mapping
ESP32 (1)	ESP32 Responsável pelos dados dos sensores
ESP32 (2)	ESP32 responsável pelo controle dos motores

SUMÁRIO

1	INTRODUÇÃO	9
1.1	MOTIVAÇÃO	9
1.2	OBJETIVOS	9
1.2.1	Objetivo Geral	9
1.2.2	Objetivos Específicos	9
1.2.2.1	Requisitos Funcionais	10
1.2.2.2	Requisitos Não-Funcionais	10
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	ROS	12
2.2	RASPBERRY PI	12
2.3	ESP32	13
2.4	HC-SR04	13
2.5	VL53L0X	13
2.6	MPU-9250	15
2.6.1	Giroscópio	15
2.6.2	Acelerômetro	15
2.6.3	Magnetômetro	16
2.7	VNH2SP30-E	16
2.8	GY-NEO6MV2	16
2.9	MOTOR DC	17
2.10	PÁGINA WEB COM ANGULAR	17
2.11	SERVIDOR FLASK	17
3	METODOLOGIA	19
3.1	VISÃO GERAL	19
3.2	PROJETO MECÂNICO	20
3.3	PROJETO DE HARDWARE	23
3.4	PROJETO DE SOFTWARE	27
3.5	INTEGRAÇÃO	35
4	EXPERIMENTOS E RESULTADOS	37
5	CRONOGRAMA E CUSTOS DO PROJETO	39
5.1	CRONOGRAMA	39
5.2	CUSTOS	42
6	CONCLUSÕES	43
6.1	CONCLUSÕES	43
6.2	TRABALHOS FUTUROS	44
	REFERÊNCIAS	45

1 INTRODUÇÃO

Esta sessão inicial tem como principal objetivo dispor acerca dos principais tópicos presentes neste trabalho, com a intenção de facilitar o entendimento do processo construtivo da estrutura e dispor com propriedade acerca dos conceitos envolvidos no tema. Na sequência serão apresentadas temáticas pertinentes à motivação para o desenvolvimento de veículos autônomos com interface usuário-máquina, tema deste trabalho.

1.1 MOTIVAÇÃO

Desenvolver um projeto relacionado a um dos principais segmentos de mercado, ainda em crescimento. A utilização de veículos autônomos vem ganhando cada vez mais espaço, integrando consigo outros setores de desenvolvimento e tecnologia. A aplicação de estruturas como estas tem como objetivo oferecer uma série de facilidades no cotidiano e também reduzir consideravelmente os índices de acidentes gerados por falha humana, além é claro de possibilitar a aplicação de uma infinidade de conceitos aprendidos durante o processo de graduação.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

Construção de um robô autônomo que deve chegar em três posições, de um mapa, selecionadas remotamente através de uma página web.

1.2.2 OBJETIVOS ESPECÍFICOS

Através de uma página Web o usuário, que estiver conectado na mesma rede que o robô, poderá informar a ordem e a localização dos três objetivos que o robô deverá alcançar. Ao final do percurso poderá acessar informações da rota realizada pelo robô e pontos onde ele

detectou obstáculos. Além disso, o usuário, também por meio da página Web, poderá fazer o robô finalizar o percurso a qualquer momento.

1.2.2.1 REQUISITOS FUNCIONAIS

RF01: Obter rota até um objetivo, inicialmente uma linha reta e posteriormente considerando os obstáculos.

RF02: Contornar obstáculos pela esquerda, se estiver mais próximo ao limite de campo à direita do robô do que à esquerda, senão contornar pela direita.

RF03: O usuário deverá ser alertado quando o robô atingir um objetivo.

RF04: O usuário deverá ser capaz de inserir as coordenadas dos três objetivos em uma interface gráfica.

RF05: O usuário deverá ser capaz de inserir a ordem que cada objetivo deve ser alcançado em uma interface gráfica.

RF06: O usuário deverá ser capaz de visualizar em uma interface gráfica os dados dos sensores e o percurso obtido pelo robô após ele atingir todos os objetivos.

1.2.2.2 REQUISITOS NÃO-FUNCIONAIS

RNF01: O robô deverá aceitar dados, das coordenadas e ordem dos objetivos.

RNF02: O sistema deverá ser desenvolvido no sistema Windows ou Linux.

RNF03: O controle de navegação deverá ser feito através do ROS.

RNF04: O robô deverá possuir um dispositivo para informar ao usuário que um objetivo foi alcançado.

RNF05: Todos os sensores deverão se comunicar com o ESP32(1) através de fios.

RNF06: O ESP32(1) deverá se comunicar, pela porta USB, com o Raspberry para enviar as informações recebidas pelos sensores.

RNF07: O Raspberry deverá processar, através do software ROS, todos os dados dos sensores e enviar informações, pela porta USB, sobre a velocidade dos motores para o ESP32(2).

RNF08: O ESP32(2) deverá se comunicar com as Pontes H e, enviar a informação de velocidade de cada motor recebida do Raspberry.

RNF09: Os 4 motores deverão ser alimentados com uma bateria de 12V.

RNF10: Os demais componentes deverão ser alimentados com baterias de 5V.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ROS

O ROS é um conjunto de bibliotecas de software e ferramentas, de código totalmente aberto, para criação de aplicações robóticas. Existem algumas versões disponíveis do ROS 1 tais como, Melodic Morenia e Noetic Ninjemys, além de versões para o ROS 2 como Galactic Geochelone, Humble Hawksbill e Foxy Fitzroy (ROS, s.d). Destas foi decidido utilizar a versão Noetic Ninjemys por ser recomendada pelo site oficial da ROS, ter bastante material disponível e ser voltada principalmente para o Linux Ubuntu 20.04 já configurado em um dos computadores da dupla.

2.2 RASPBERRY PI

Raspberry Pi é um computador, que roda Linux, do tamanho de um cartão de crédito sendo alguns modelos de tamanho até menor, muito utilizado em projetos devido ao seu tamanho, portabilidade, baixo custo e por ter melhor capacidade de processamento do que um microcontrolador convencional.

O Raspberry Pi 4B utiliza-se de um processador Broadcom BCM2711 com 4 núcleos Cortex-A72 (ARM v8) de 64-bit e uma frequência de 1.5 GHz, possui Wi-Fi e Bluetooth e, está disponível com 1, 2 ou 4 GB de memória RAM LPDDR4. Possui 2 portas micro HDMI com suporte a vídeos 4K e 60fps, 2 portas USB 3.0 e 2 portas USB 2.0. Além disso, ele possui 28 portas GPIO que suportam várias opções de interface, como I2C, UART, SPI, PWM e entre outros (RASPBERRY PI, 2019).

Embora possa ser utilizado em várias soluções devido a quantidade de benefícios, em algumas não é adequado devido ao considerável gasto de energia em comparação a microcontroladores comuns e a limitação na quantidade de portas IO disponíveis.

2.3 ESP32

ESP32 é um SoC com microcontrolador integrado, Wi-Fi e Bluetooth de 2,4 GHz, fabricado pela TSMC, de ultra baixa potência e tecnologia de 40 nm. Ele é projetado e otimizado para o melhor desempenho de energia, desempenho de RF, robustez, versatilidade, recursos e confiabilidade, além de ser útil para uma ampla variedade de aplicações e diferentes perfis de potência. Por esse motivo ele também é muito utilizado nos mais diversos projetos, contando com a vantagem de ser mais barato mas com menor desempenho em comparação ao Raspberry Pi 4B.

O ESP32 DevKit V1 utiliza-se de um processador Xtensa de dois núcleos 32-bit LX6, memória RAM de 520 KByte e um clock máximo de 240 MHz. Ele possui 48 pinos de GPIO que suportam várias opções de interface, como I2C, UART, SPI, PWM, Sensor de Toque e entre outros (ESPRESSIF SYSTEMS, 2016).

2.4 HC-SR04

O HC-SR04 é um sensor de distância Ultrassônico ToF que funciona de 2 cm a 4m com precisão de 3mm. ToF em tradução livre "Tempo de Voo" é um método para medir a distância entre um sensor e um objeto, baseado na diferença de tempo entre a emissão de um sinal e seu retorno ao sensor, após ser refletido por um objeto (TERABEE, s.d). O sensor ultrassônico funciona ativando o pino de IO TRIGGER para nível lógico alto por pelo menos 10 microssegundos, após isso o sensor automaticamente envia oito pulsos ultrassônicos de 40 kHz e detecta se há um sinal de pulso de volta, caso exista ele ativa um sinal de nível lógico alto no pino de IO ECHO pelo mesmo tempo que o pulso levou para ir e voltar. A Figura 1 é uma ilustração deste processo.

2.5 VL53L0X

O VL53L0X é um sensor de distância a Laser ToF de nova geração, com leituras mais rápidas e maior precisão em comparação a um sensor ultrassônico. Ele pode medir distâncias absolutas de até 2m, e com o emissor VCSEL de 940nm acoplado a filtros infravermelhos físicos internos, totalmente invisível ao olho humano, , permite maior alcance de distância e menor interferência à luz ambiente (STMICROELECTRONICS, 2016).

Detalhes das condições de medição podem ser visualizados na Figura 2 e detalhes sobre seu desempenho típico podem ser vistos na Figura 3.

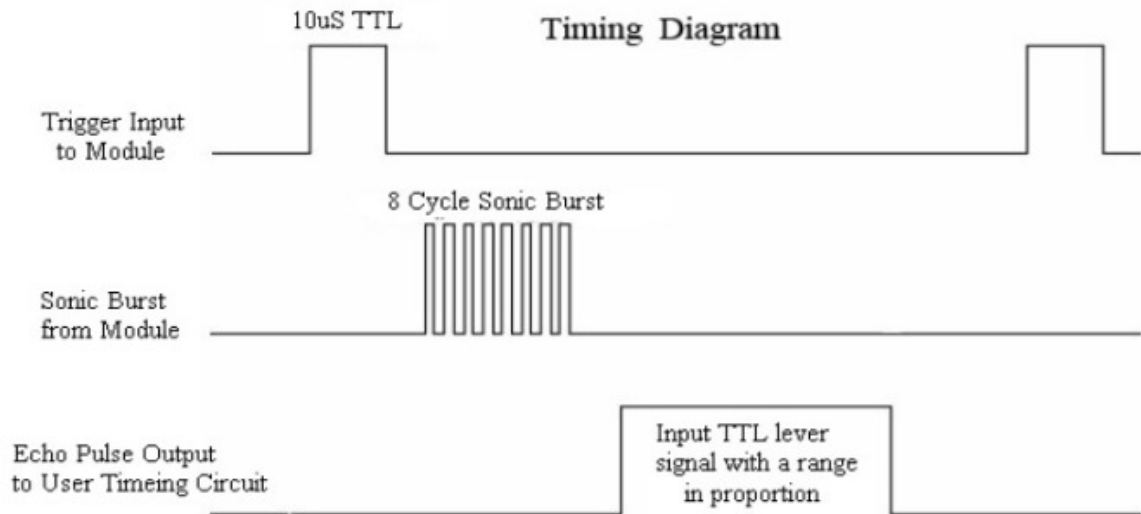


Figura 1: Funcionamento do Sensor HR-SC04.

Fonte: (MORGAN, 2014)

5.3 Ranging accuracy

5.3.1 Standard deviation

Ranging accuracy can be characterized by standard deviation. It includes Measure-to-Measure and Part-to-Part (silicon) dispersion.

Table 12. Ranging accuracy

Target reflectance level (Full FOV)	Indoor (no infrared)			Outdoor		
	Distance	33ms	66ms	Distance	33ms	66ms
White Target (88%)	at 120cm	4%	3%	at 60cm	7%	6%
Grey Target (17%)	at 70cm	7%	6%	at 40cm	12%	9%

Measurement conditions:

- Targets reflectance used: Grey (17%), White (88%)
- Offset correction done at 10cm from sensor.
- Indoor: no Infrared / Outdoor: eq. 5kLux equivalent sunlight (10kcps/SPAD)
- Nominal Voltage (2v8) and Temperature (23degreesC)
- All distances are for a complete Field of View covered (FOV = 25degrees)
- Detection rate is considered at 94% minimum

Figura 2: Precisão de alcance do VL53L0X.

Fonte: (STMICROELECTRONICS, 2016)

Table 13 details typical performance for the four example ranging profiles, as per measurement conditions in *Section 5.3: Ranging accuracy*.

Table 13. Range profiles

Range Profile	Range timing budget	Typical performance	Typical application
Default mode	30ms	1.2m, accuracy as per <i>Table 12</i>	standard
High accuracy	200ms	1.2m, accuracy < +/- 3%	precise measurement
Long range	33ms	2m, accuracy as per <i>Table 12</i>	long ranging, only for dark conditions (no IR)
High speed	20ms	1.2m, accuracy +/- 5%	high speed where accuracy is not priority

Figura 3: Detalhes de desempenho típico do VL53L0X.

Fonte: (STMICROELECTRONICS, 2016)

2.6 MPU-9250

É um sensor de movimento de 9 eixos livres que combina um giroscópio de 3 eixos, acelerômetro de 3 eixos e magnetômetro de 3 eixos, tudo em um pequeno dispositivo de 3x3x1mm. Possui uma faixa de escala do giroscópio programável pelo usuário de ± 250 , ± 500 , ± 1000 e $\pm 2000^\circ/\text{sec}$ (dps), uma faixa de escala do acelerômetro programável pelo usuário de $\pm 2g$, $\pm 4g$, $\pm 8g$ e $\pm 16g$, sendo g igual a $9,80665 \text{ m/s}^2$, e uma faixa de escala total do magnetômetro de $\pm 4800\mu\text{T}$ (INVENSENSE, 2016).

2.6.1 GIROSCÓPIO

Quando os giroscópios são girados em torno de qualquer um dos eixos de detecção, o Efeito Coriolis causa uma vibração que é detectada por um pickoff capacitivo. O sinal resultante é amplificado, demodulado e filtrado para produzir uma tensão proporcional ao deslocamento angular no eixo (INVENSENSE, 2016).

2.6.2 ACELERÔMETRO

Aceleração ao longo de determinado eixo induz deslocamento na massa de prova correspondente, cada eixo possui uma massa de prova, e sensores capacitivos detectam o deslocamento dessa massa fornecendo valores de aceleração para o eixo (INVENSENSE, 2016).

2.6.3 MAGNETÔMETRO

Usa um sensor de efeito Hall altamente sensível em cada um dos eixos, este quando sob a aplicação de um campo magnético, responde com uma variação em sua tensão de saída que é usado para aferir a variação do campo magnético no eixo (INVENSENSE, 2016).

2.7 VNH2SP30-E

É um driver de Ponte H completo, ilustrado na Figura 4, destinado a uma vasta gama de aplicações automotivas. Ela fornece uma corrente máxima de saída de 30A, possui desligamento automático em caso de sobreaquecimento, baixo consumo de energia em espera e suporta sinais PWM de até 10 kHz (STMICROELECTRONICS, 2017).

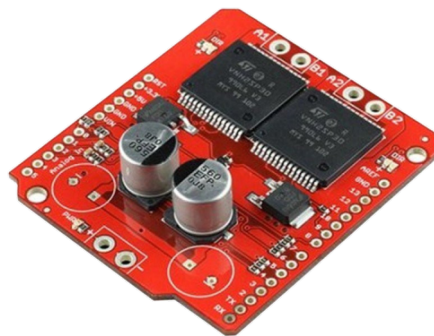


Figura 4: Ponte H modelo VNH2SP30-E.

Fonte: (BAÚ DA ELETRÔNICA, 2022)

2.8 GY-NEO6MV2

A série de chips NEO-6 é uma família de receptores GPS autônomos com o mecanismo de posicionamento u-blox 6 de alto desempenho, possui uma arquitetura compacta e opções de energia e memória que tornam os módulos NEO-6 ideais para dispositivos móveis operados por bateria com restrições de custo e espaço, sendo o chip de tamanho 16x12,2x2,4mm e o módulo de tamanho 36x26x4mm. Além disso ele possui um mecanismo de aquisição dedicado, com 2 milhões de correlacionadores, que são capazes de realizar buscas paralelas massivas de tempo/frequência no espaço, permitindo encontrar satélites instantaneamente (U-BLOX, 2011).

2.9 MOTOR DC

O motor DC utilizado possui uma caixa de redução acoplada que gera maior torque para as rodas, é alimentado com 12V e atinge uma rotação máxima de 1600 RPM. Cada par de motores suportam peso de no máximo 7Kg e cada um consome uma corrente de 0,36A sem carga, sendo 1,76A em máxima eficiência. O motor DC utilizado está ilustrado na Figura 5.

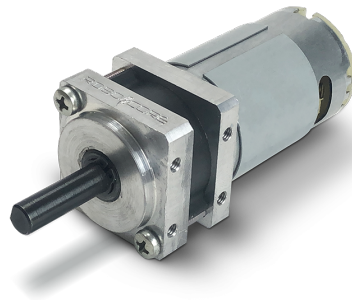


Figura 5: Motor DC.

Fonte: (ROBOCORE, 2022)

2.10 PÁGINA WEB COM ANGULAR

A página web foi feita em TypeScript com o framework de aplicações web Angular. Angular é uma plataforma de desenvolvimento criada pela Google que é baseada em componentes e que inclui uma coleção de bibliotecas integradas que possuem uma ampla variedade de funções, incluindo roteamento, gestão de formulários, comunicação cliente-servidor, e mais. Também inclui uma variedade de ferramentas de desenvolvedor que ajudam nas fases de desenvolvimento, build, testes e atualizações do código (GOOGLE, 2010-2022).

2.11 SERVIDOR FLASK

Para receber as requisições HTTP da página web, foi feito uma aplicação web implementada com Flask. Este framework de desenvolvimento web foi concebido com o objetivo de simplificar o processo criativo, sem funções tais como ORM ou autenticação, sendo por esse motivo classificado como um microframework. Estas aplicações, tipicamente provêm serviços facilitadores para desenvolvimento web através de bibliotecas para, por exemplo,

gerenciamento de sessões e acesso a banco de dados (PALLETTS, 2010). Nesta aplicação o Flask deve receber da página web os valores das posições dos objetivos informados pelo usuário.

3 METODOLOGIA

3.1 VISÃO GERAL

A concepção do projeto foi baseada no desafio da RoboCore chamado Robô Trekking, o qual define certas regras para a competição (ROBOCORE, 2018), das quais adaptamos para o contexto da disciplina de Oficina de Integração 2. Após isso, foi desenvolvida uma solução em alto nível representada no diagrama de blocos da Figura 6.

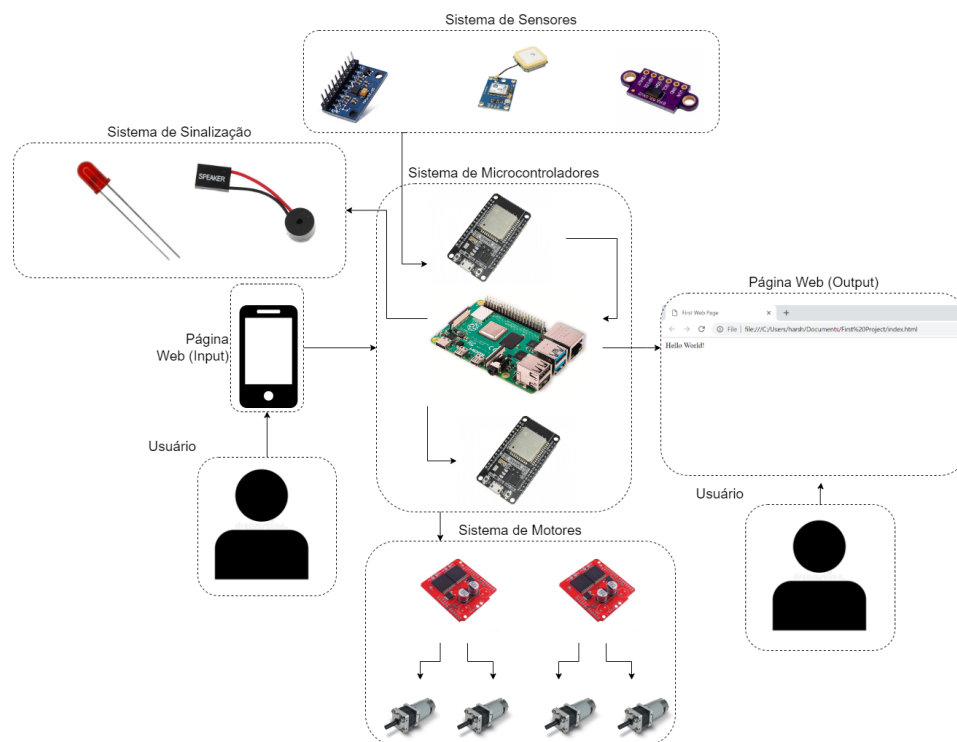


Figura 6: Diagramas de Blocos do Sistema.

Fonte: Os autores, 2022.

O Diagrama da Figura 6 mostra um ESP32 ligado ao Sistema de sensores e outro ao Sistema de motores, onde cada uma das pontes H controlam um par de motores, sendo todo o sistema controlado pelo Raspberry Pi 4B. Foi decidido usar essa configuração devido a

limitação no número de portas do Raspberry Pi 4B, separação dos sistemas, sensores e motores, em módulos e disponibilidade de microcontroladores ESP32 de um dos membros da equipe.

O projeto foi dividido em Mecânica, Hardware e Software. A Mecânica foi realizada em conjunto, um dos integrantes ficou responsável pelo Software e o outro pelo Hardware, apesar da divisão entre Software e Hardware, conforme a necessidade um membro auxiliou o outro e vice-versa.

3.2 PROJETO MECÂNICO

O projeto do Marvin, ilustrado nas Figuras 7 e 8, foi feito com auxílio do software chamado SolidWorks, levando em conta as medidas de todos os componentes que seriam utilizados e suas posições. Para o chassi foi escolhido uma caixa em MDF de 30x25x20cm, e para os eixos foram utilizados parafusos de cabeça sextavada M8 com 80mm de comprimento, fixados em mancais KFL de 8mm e acoplados aos motores utilizando acopladores flexíveis 8x8mm. O resultado final do chassi com os componentes citados acima está ilustrado na Figura 9.

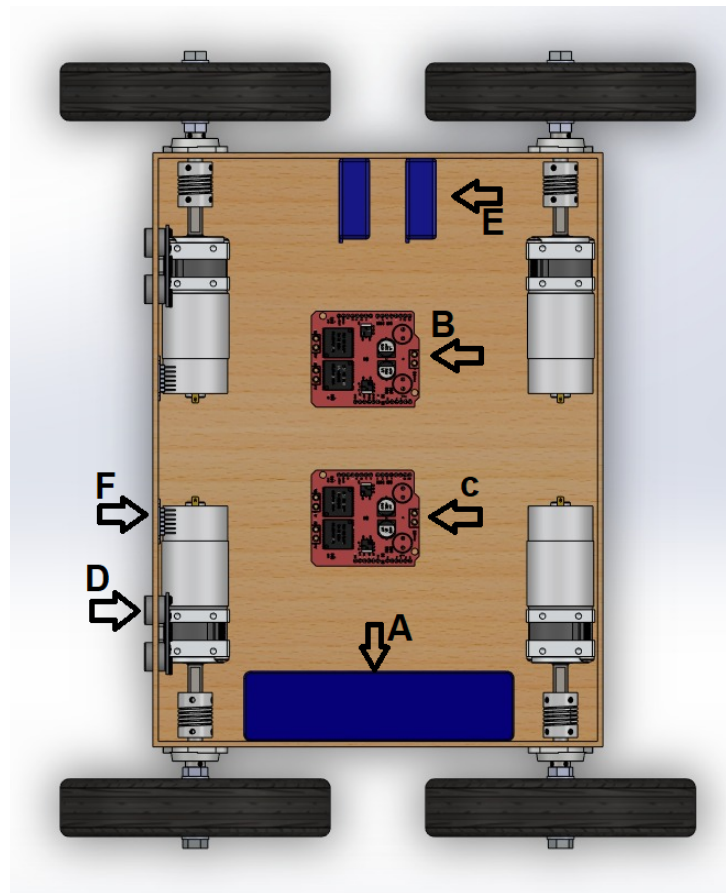


Figura 7: Primeiro Andar Projeto Marvin em SolidWorks

Fonte: Os autores, 2022.

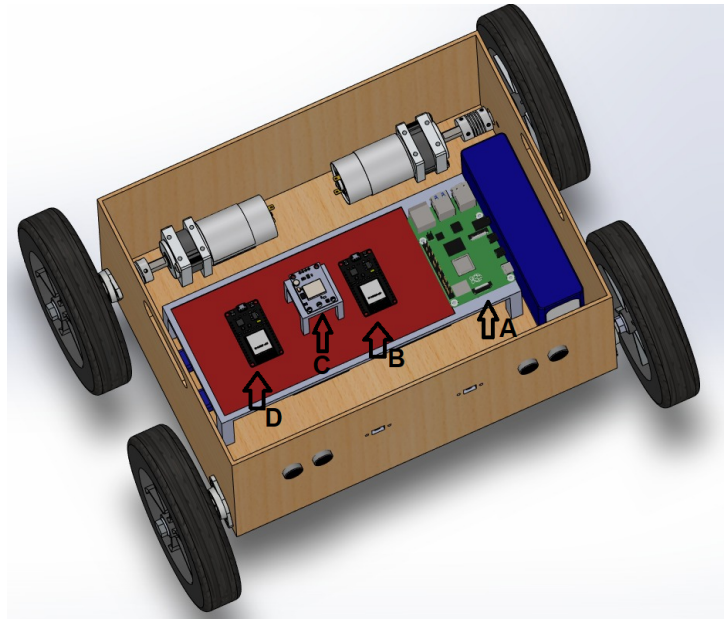


Figura 8: Segundo Andar Projeto Marvin em SolidWorks
Fonte: Os autores, 2022.



Figura 9: Motores conectados aos eixos
Fonte: Os autores, 2022.

Na Figura 7 as setas B e C indicam as posições das duas pontes H utilizadas (VNH2SP30-E), a seta A indica posição das baterias utilizadas, a seta E indica a posição que iria ficar uma bateria de 5V que foi retirada durante a integração, a seta F indica a posição de um dos sensores laser utilizados e a seta D indica a posição de um dos três sensores ultrassônicos utilizados, nessa figura são apresentados dois sensores ultrassônicos mas durante a integração foi decidido utilizar três destes sensores para obter melhores resultados.

Na Figura 8 a seta A indica a posição do Raspberry Pi 4B, a seta B indica a posição do ESP32 (1), a seta C indica a posição para o sensor MPU-9250 e GPS (GY-NEO6MV2) e a seta D indica a posição para o ESP32 (2) responsável pelo controle dos motores. Os componentes das setas B, C e D estão sob a placa de circuito impresso confeccionada para o projeto e esta placa posicionada sob uma plataforma impressa em impressora 3D, formando o segundo andar do projeto.

3.3 PROJETO DE HARDWARE

A etapa de projeto eletrônico é constituída por três microcontroladores, quatro atuadores e diversos sensores.

Os atuadores são motores posicionados em cada um dos quatro eixos, cada par de motor é controlado por uma das pontes H e todo o circuito, pontes H e motores, são alimentados por uma bateria 12V, sendo utilizado o regulador de tensão 7805 para a alimentação 5V das pontes H. Cada sinal enviado do ESP32 (2) para as pontes H passa por seu respectivo optoacoplador, para que não haja contato físico entre os diferentes circuitos e, evitar problemas com ruído nos sensores e microcontroladores causado pelas cargas indutivas dos quatro motores.

Os sensores ultrassônicos e laser são responsáveis por detectar obstáculos no caminho até os objetivos, o laser foi escolhido por ter maior precisão e velocidade, na obtenção de valores de distância, em relação ao sensor ultrassônico, porém o sensor laser é unidirecional, então foi adicionado sensores ultrassônicos próximos as laterais do robô para se obter melhores resultados quando obstáculos estiverem posicionadas em uma quina, por exemplo. Também foi adicionado um sensor ultrassônico entre os sensores laser para que, caso ocorra problema com a leitura aferida pelos sensores laser, tenha um sensor de outro tipo que garanta um mínimo de exatidão para a distância até o obstáculo.

O módulo MPU é essencial para determinar a orientação, além disso, ele pode ser utilizado para determinar se o robô sofreu uma queda durante o circuito e iniciar medidas para

retornar a posição correta, mas não foi utilizado com esse propósito neste projeto.

Os valores dos sensores são lidos e interpretados pelo ESP32 (1), que ao receber solicitações, por meio de comunicação serial, de leitura do Raspberry PI responde com o valor para o sensor requisitado. O Raspberry processa a informação recebida do ESP32 (1) e envia para o ESP32 (2), também por comunicação serial, os valores de rotação e sentido de cada motor. Todos os sensores são alimentados com tensões de 5V fornecidas pelo pino chamado Vin do ESP32 (1), além disso, as duas tensões de 5V emitidas pelo pino Vin do ESP32 (1) e ESP32 (2) foram conectadas em paralelo, garantindo que exista corrente de sobra para todos os sensores. Os dois microcontroladores ESP32 são conectados e alimentados por portas USB 3.0 do Raspberry Pi 4B.

Devido a quantidade de componentes necessários para o funcionamento adequado do robô e para manter uma boa organização do projeto eletrônico, foram realizados testes de montagem na protoboard e em seguida foi confeccionada a placa de circuito impresso, como representado através diagrama da Figura 10. Nesta, pode-se observar os modelos relativos a cada um dos elementos de circuito anexados à placa e também identificar corretamente os pontos de ligação de cada um destes. Logo após, foi feito o posicionamento de cada componente na placa buscando obter a melhor organização possível, para facilitar o processo de criação das trilhas e facilitar a posterior conexão dos componentes na placa, a Figura 11 ilustra uma pré-visualização dos componentes na placa, a Figura 12 ilustra as trilhas feitas para a placa e a Figura 13 ilustra o resultado final. Além disso, a placa conta com dois pinos que podem ser usados para alimentação de 5V por uma bateria e espaço ao redor do regulador de tensão 7805 para adição de um dissipador de calor.

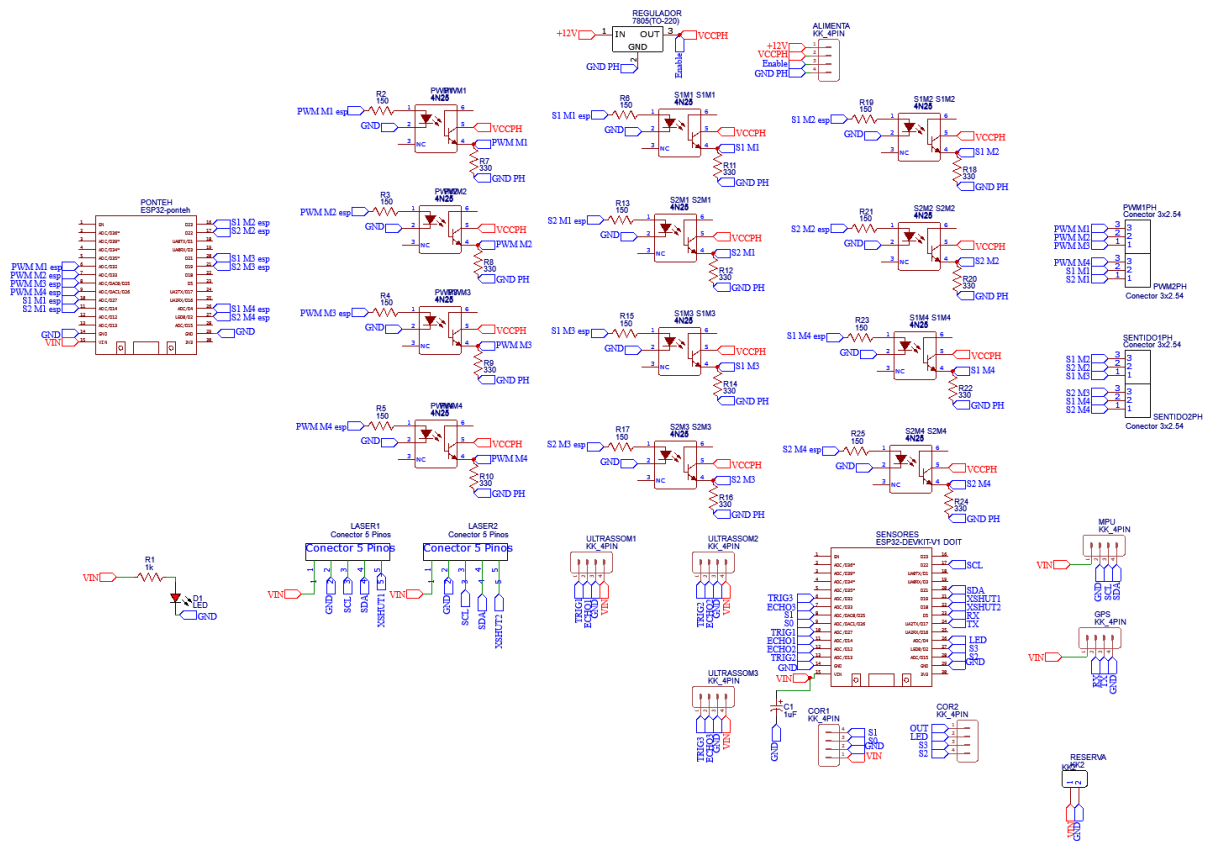


Figura 10: Esquemático do projeto eletrônico.

Fonte: Os Autores, 2022.

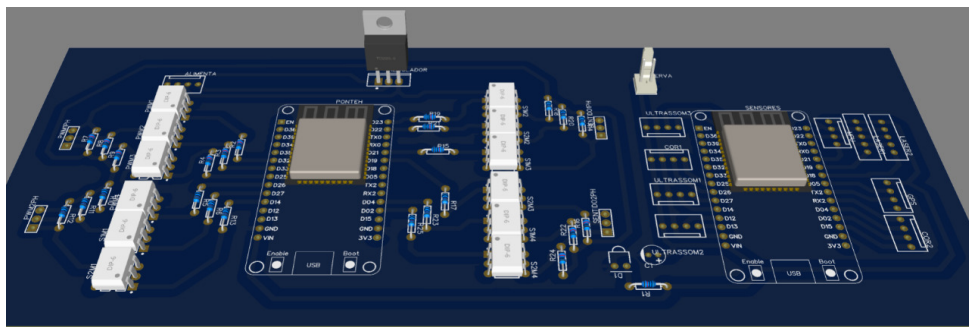


Figura 11: Pré-visualização 3D da placa.

Fonte: Os Autores, 2022.

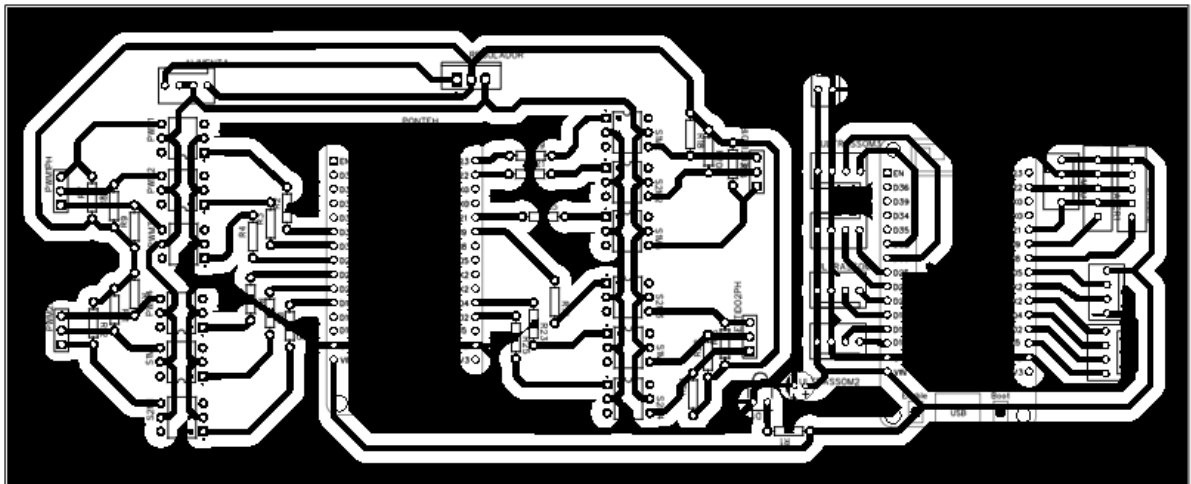


Figura 12: Roteamento da placa.

Fonte: Os Autores, 2022.

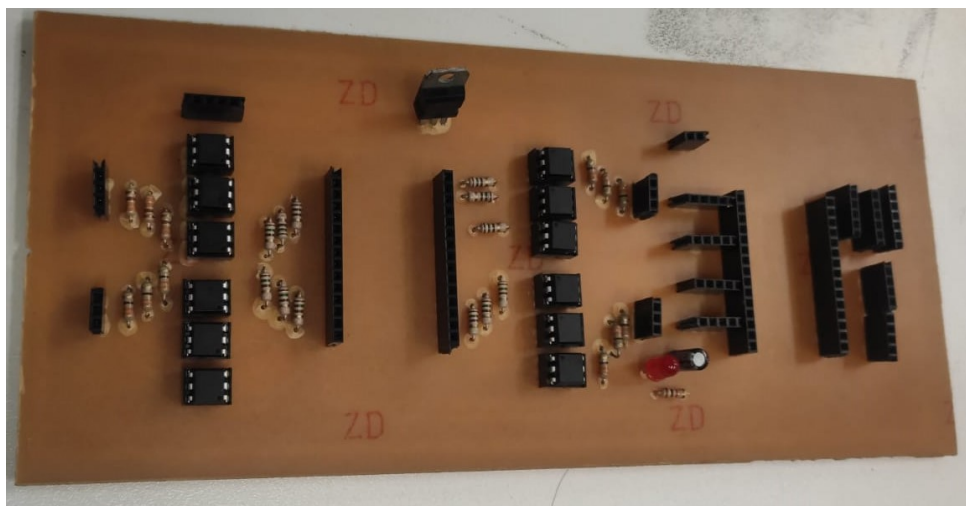


Figura 13: Resultado final da placa.

Fonte: Os Autores, 2022.

Nesta etapa foram confeccionadas três placas de circuito impresso, sendo a última placa confeccionada durante a etapa de integração. A primeira versão ficou com as trilhas muito próximas uma das outras, o que acarretou em diversos problemas de curto-circuito entre as trilhas durante os testes com o multímetro. A segunda versão foi melhorada e o espaçamento ficou excelente, mas durante a etapa de integração foi identificado que algumas trilhas estavam faltando para o funcionamento correto dos optoacopladores escolhidos pela dupla. Para a terceira versão da placa, foi considerando todo o aprendizado e erros cometidos nos processos anteriores, além da alteração no modelo de optoacoplador.

3.4 PROJETO DE SOFTWARE

O projeto de software teve início com modelagem UML, primeiramente foi realizado o diagrama de Casos de Uso, ilustrado na Figura 14, para representar a interação do usuário com a interface do sistema. Na sequência foi definido quais requisitos funcionais estavam sendo atendidos por cada Caso de Uso:

- Informar Posição de Objetivos - Neste requisito o usuário deverá ser capaz de inserir as coordenadas dos três objetivos, por meio de uma interface gráfica implementada através da página web. - RF04.
- Informar Ordem dos Objetivos - O usuário deverá ser capaz de inserir a ordem desejada que cada objetivo deve ser alcançado em uma interface gráfica - RF05.
- Consultar Trajeto e Sensores - O usuário deverá ser capaz de visualizar em uma interface gráfica os dados coletados pelos sensores e observar o percurso realizado pelo robô após ele atingir todos os objetivos da prova - RF06.
- Parar Percurso - O usuário deverá ser capaz de interromper o funcionamento do veículo a qualquer momento, através da utilização de um comando disponível na interface com o usuário.

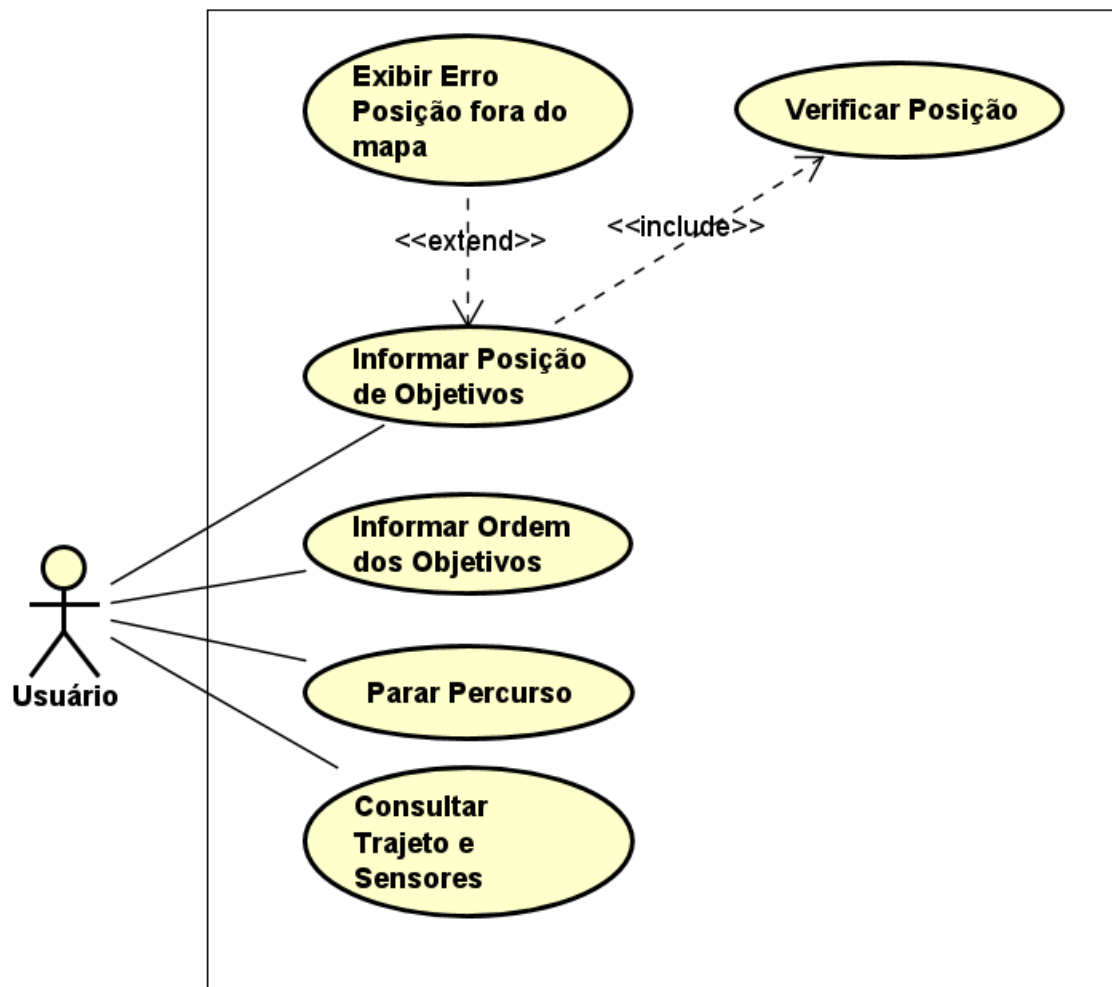


Figura 14: Diagrama de Casos de Uso.

Fonte: Os Autores, 2022.

Para cada Caso de Uso foi feita uma rotina de execução possível para o usuário. Essas rotinas são apresentadas nas Figuras 15, 16, 17 e 18.

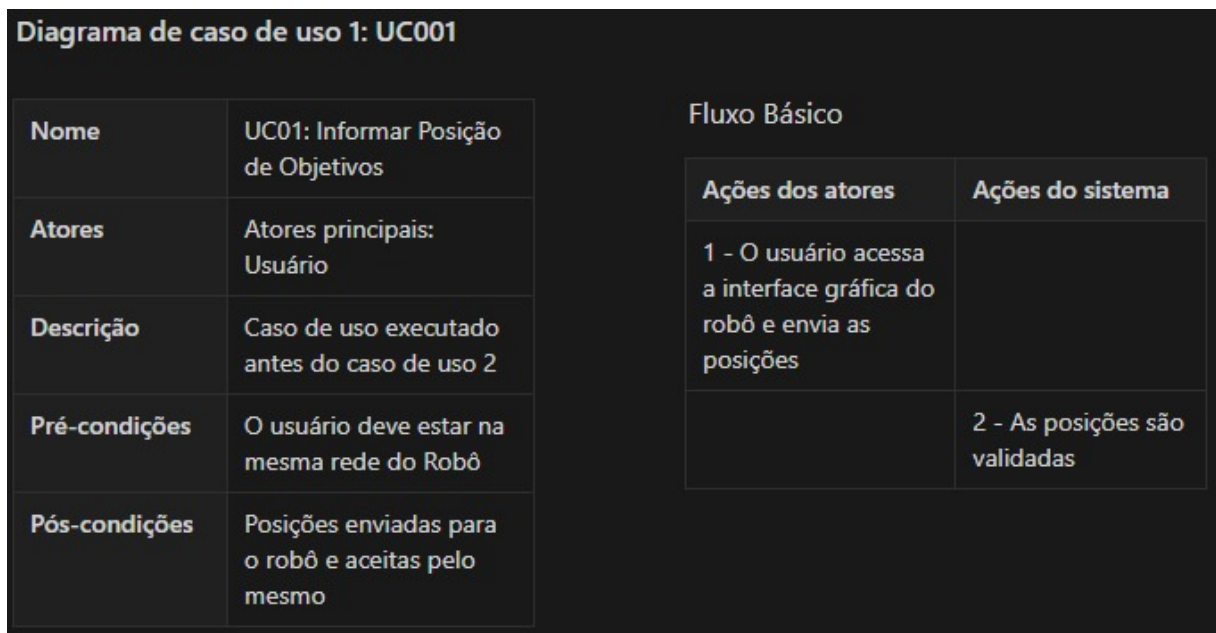


Figura 15: Diagrama de Casos de Uso 1.

Fonte: Os Autores, 2022.

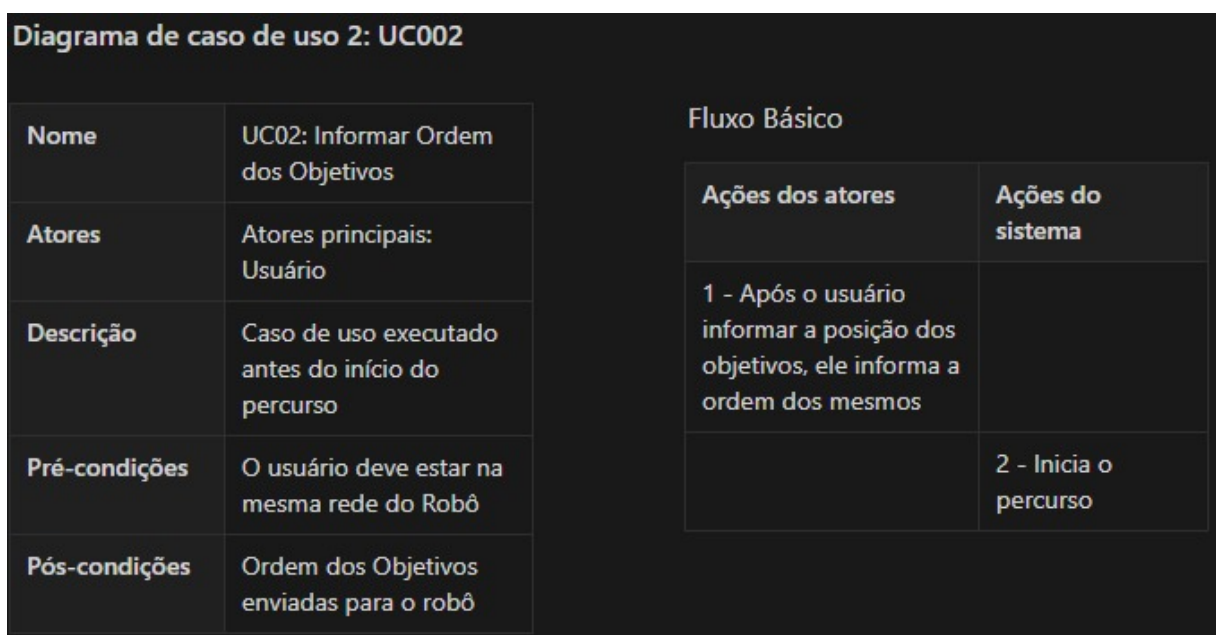


Figura 16: Diagrama de Casos de Uso 2.

Fonte: Os Autores, 2022.

Diagrama de caso de uso 3: UC003			
Nome	UC03: Consultar Trajeto e Sensores	Fluxo Básico	
Atores	Atores principais: Usuário		
Descrição	Caso de uso executado ao final do percurso		
Pré-condições	O usuário deve estar na mesma rede do Robô e o robô deve ter sinalizado o fim do percurso		
Pós-condições	O usuário recebe um arquivo com o trajeto e dados dos sensores		
		Ações dos atores	Ações do sistema
		1 - O usuário acessa a interface gráfica e solicita informações do percurso	
			2 - O sistema fornece um arquivo com dados do percurso e sensores

Figura 17: Diagrama de Casos de Uso 3.

Fonte: Os Autores, 2022.

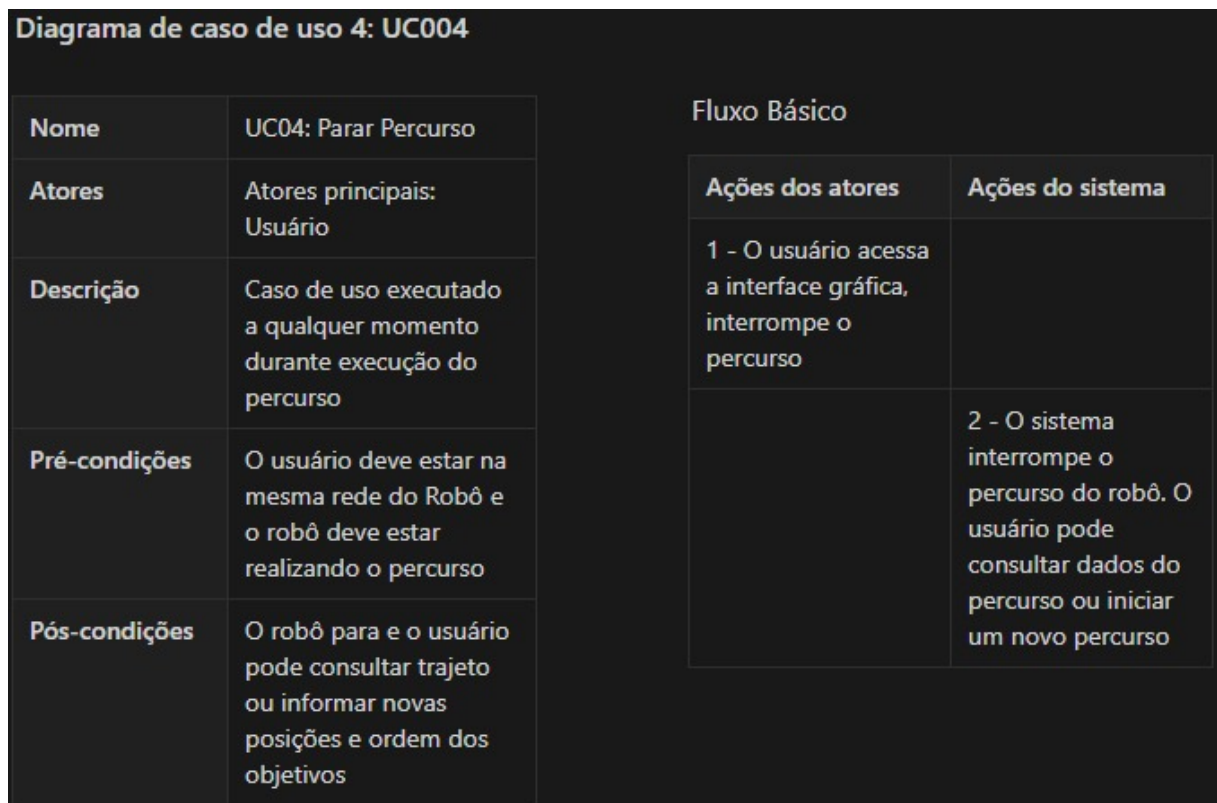


Figura 18: Diagrama de Casos de Uso 4.

Fonte: Os Autores, 2022.

Logo após identificar e especificar os Casos de Uso do software, foi realizado os diagramas de sequência para os casos de uso, ilustrados nas Figuras 19 e 20.

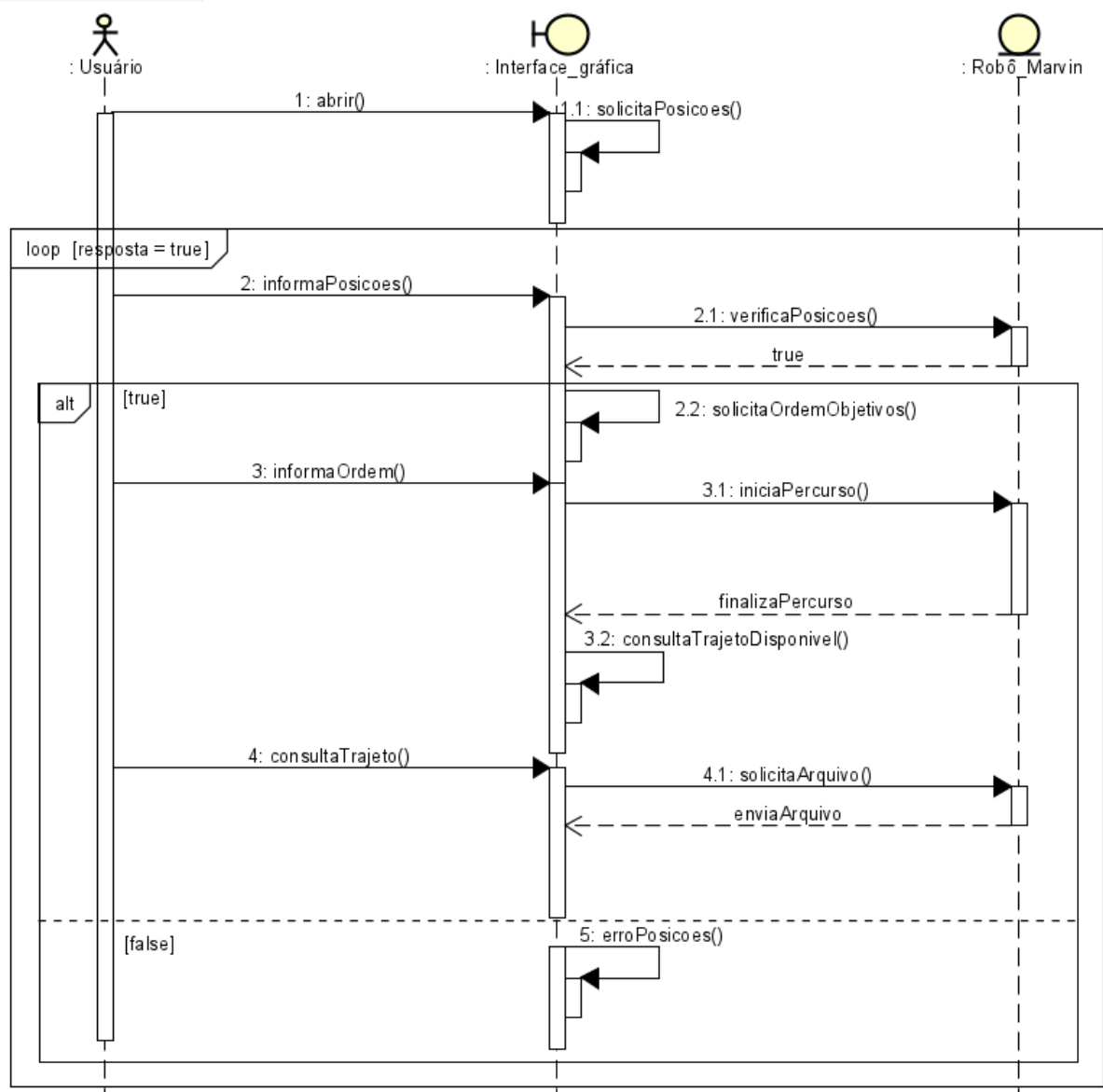


Figura 19: Diagrama de Sequência para os Casos de Uso UC01, UC02 e UC03.

Fonte: Os Autores, 2022.

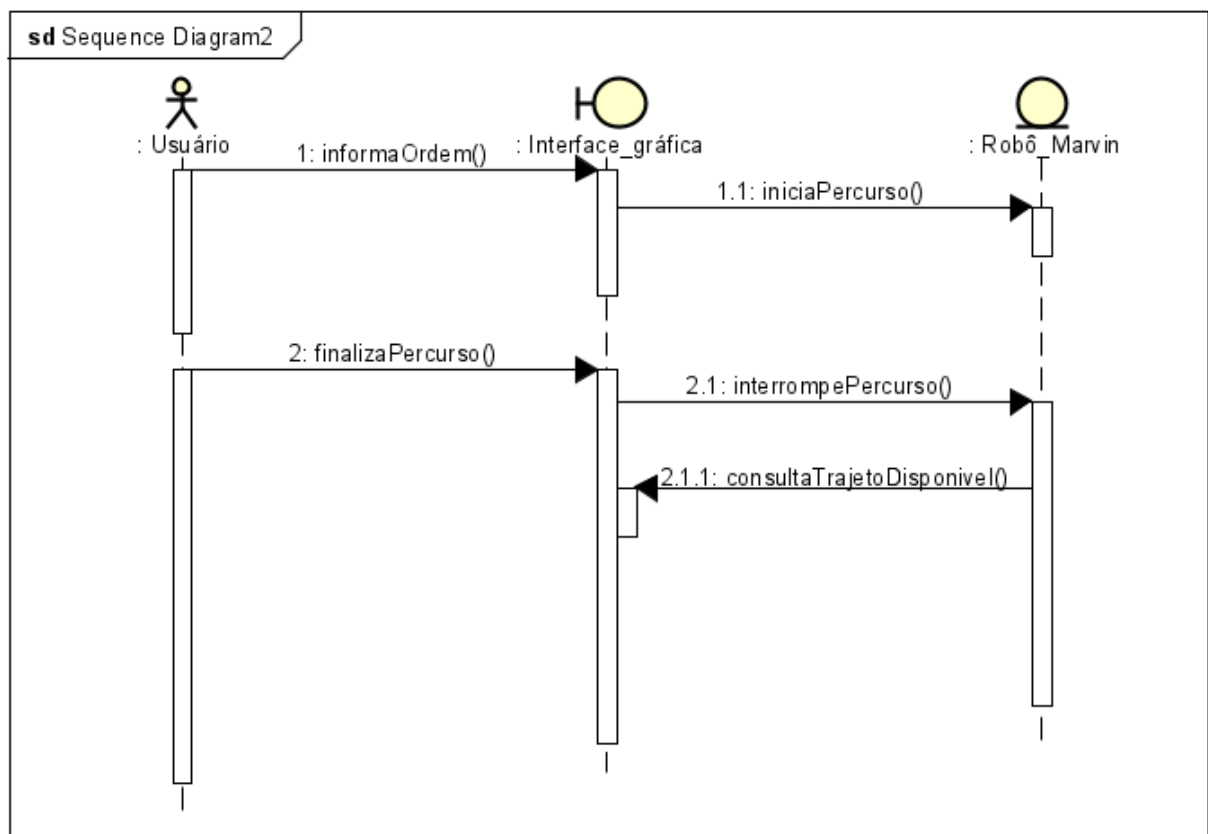


Figura 20: Diagrama de Sequência para o Caso de Uso UC04.

Fonte: Os Autores, 2022.

A partir destes diagramas foi identificado a necessidade de um diagrama de estados, ilustrado na 21, para representar o fluxo completo de funcionamento do robô.

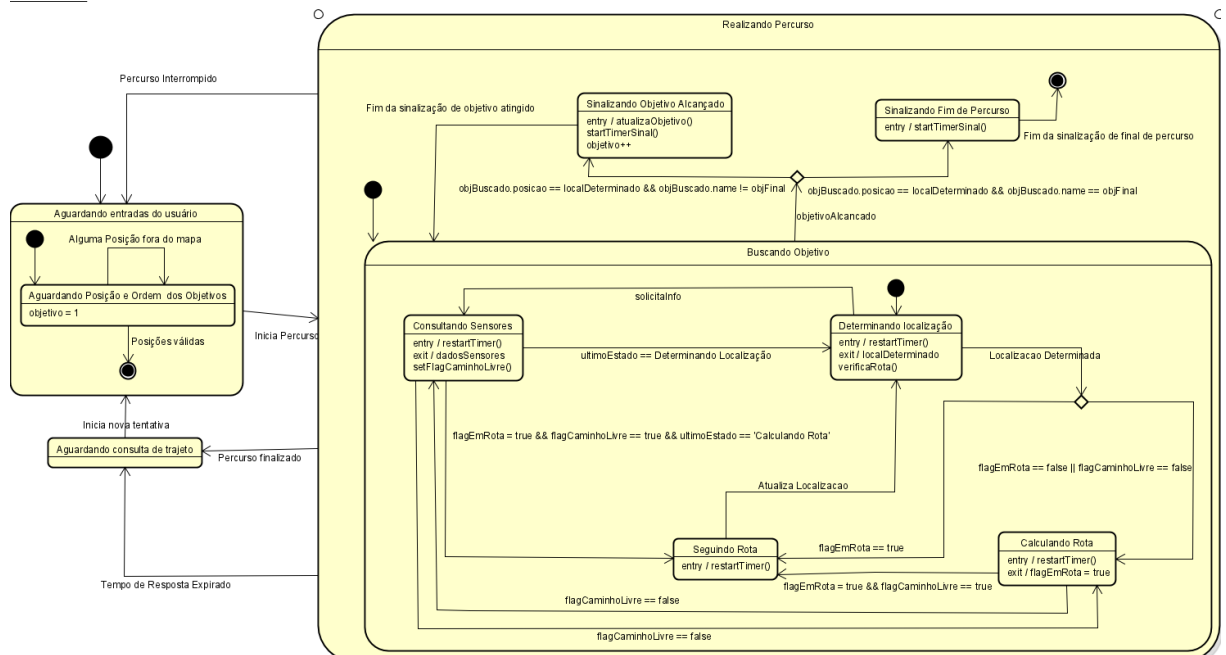
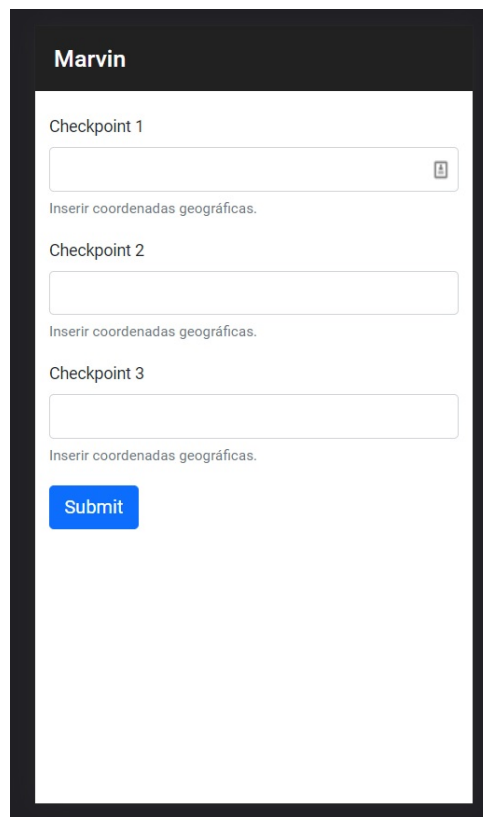


Figura 21: Diagrama de Estados.

Fonte: Os Autores, 2022.

Após a realização do projeto de software, foi iniciado a implementação utilizando as bibliotecas disponíveis no ROS e foi criado um protótipo da tela inicial da página Web, ilustrado na Figura 22.



Marvin

Checkpoint 1

Inserir coordenadas geográficas.

Checkpoint 2

Inserir coordenadas geográficas.

Checkpoint 3

Inserir coordenadas geográficas.

Submit

Figura 22: Protótipo da Página Web.

Fonte: Os Autores, 2022.

3.5 INTEGRAÇÃO

A integração entre os elementos de mecânica, hardware e software do projeto foi realizado durante o processo de implementação do robô, iniciando com as etapas de planejamento e montagem na protoboard até o dimensionamento dos componentes, implementação na placa de circuito impresso e montagem completa da estrutura, seguindo as posições dos componentes estabelecidas na etapa mecânica, mais especificamente nas Figuras 7 e 8. O resultado da integração entre mecânica e hardware está ilustrado na Figura 23.

A parte software foi integrada na sequência, através da realização de testes em ambientes controlados, com a presença de obstáculos, afim de verificar as diversas funcionalidades do robô, sua capacidade de reconhecimento e o funcionamento da interface criada para a utilização do usuário.

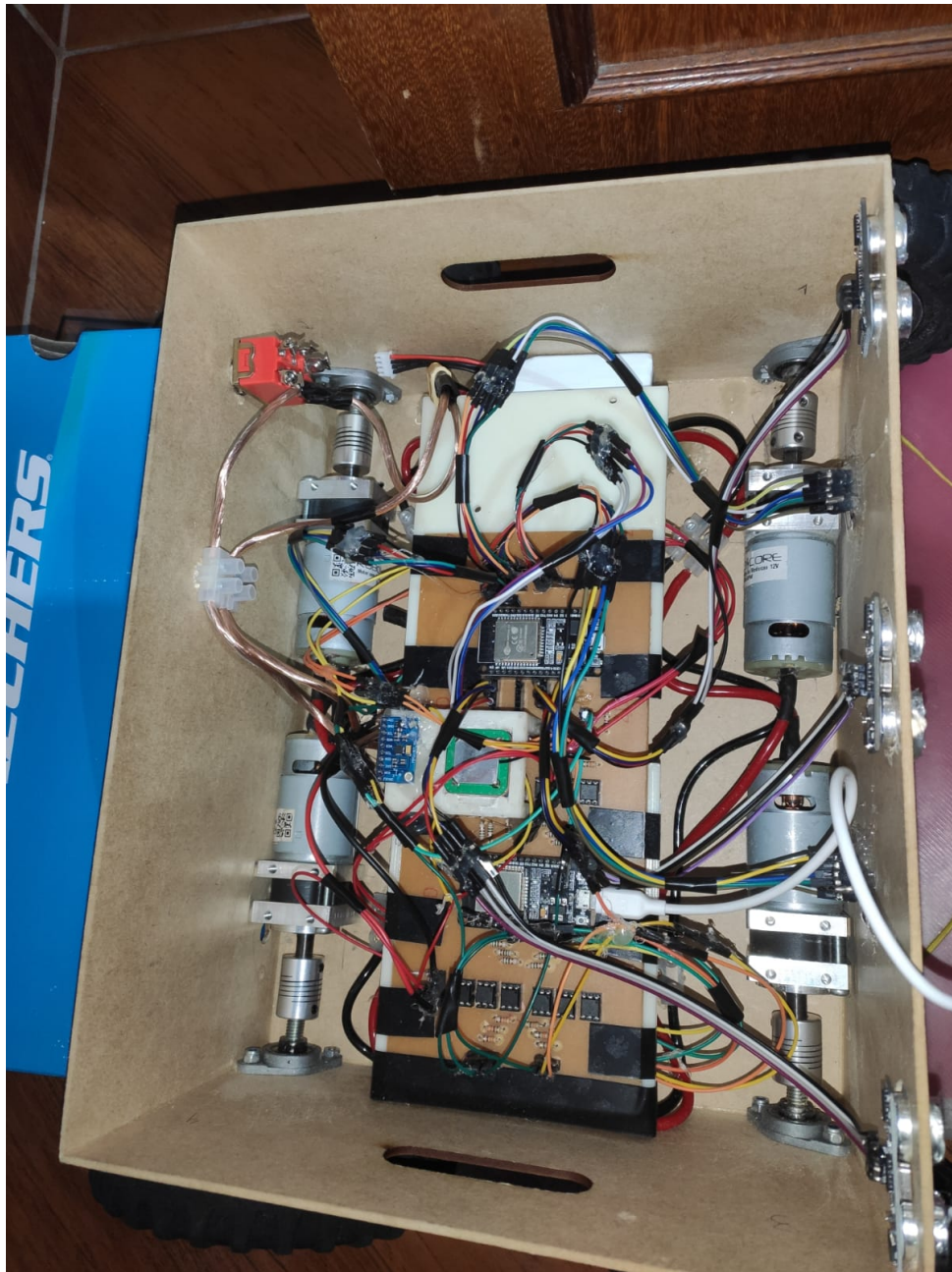


Figura 23: Integração Mecânica com Hardware.

Fonte: Os Autores, 2022.

4 EXPERIMENTOS E RESULTADOS

Durante as etapas de desenvolvimento do projeto foram avaliados os possíveis riscos associados ao processo de integração da estrutura, uma vez que uma infinidade de componentes associados a parte de hardware foram utilizados. Dessa maneira, foi considerada a possibilidade de aparecimento de ruídos e algumas incompatibilidades durante o funcionamento da estrutura, que em uma primeira análise, deveriam ser atenuados afim de facilitar o processo de integração e teste dos elementos de software utilizados.

Entre as etapas de desenvolvimento experimental do projeto devem ser citadas, principalmente, as de montagem na protoboard, implementação e fabricação da placa de circuito impresso, montagem completa e integração com a parte de software.

Considerando como uma das etapas de desenvolvimento inicial da estrutura a parte experimental de avaliação da parte eletrônica via protoboard, permitiu a visualização das funcionalidades exigidas para o funcionamento do robô. Nesta os componentes foram conectados e rotinas de funcionamento com testagem foram utilizadas, afim de verificar a adequação dos componentes a aplicação. Disto foram definidos quais e a forma de utilização que seriam posteriormente utilizados na fabricação da placa de circuito impresso.

A próxima etapa de desenvolvimento ocorreu em duas partes principais. A primeira delas foi realizada com a utilização do software EasyEDA, que possibilita a modelagem de sistemas eletrônicos integrados. A segunda parte foi feita na prática e consistiu na fabricação da placa e teste de funcionalidade dos elementos. Uma vez que todos os testes realizados nesta etapa foram bem sucedidos, a placa estava pronta para executar suas atividades dentro da aplicação.

A penúltima etapa de desenvolvimento da estrutura consistiu na integração mecânica dos elementos do sistema, sendo eles, a placa, chassis, rodas, sensores e etc. Nesta, todos os componentes foram agrupados satisfatoriamente, não apresentando maiores problemas no processo.

A última etapa de desenvolvimento foi realizada somente após a implementação

de hardware da estrutura. A parte de integração com o usuário foi realizada sem maiores dificuldades, possibilitando a obtenção de dados de percurso realizados durante as etapas de teste, permitindo o controle de funcionamento e a indicação dos pontos desejados. Os testes relativos à esta última etapa foram realizados em ambiente controlado, permitindo aferir a funcionalidade da estrutura completa, que concluiu com êxito os requisitos de deslocamento e identificação de obstáculos.

5 CRONOGRAMA E CUSTOS DO PROJETO

5.1 CRONOGRAMA

O Cronograma de realização das atividades, entregas e amostragem de resultados foi ilustrado por meio das figuras Figuras 24 e 25. Através da visualização deste cronograma é possível fazer o acompanhamento das entregas e avaliar os índices de desempenho de desenvolvimento do projeto.

Name	Prazo	Responsável	Status	# Horas Previstas	# Gasto em Horas	Σ Previsto + Erro
Idealização de projeto	March 16, 2022 → March 22, 2022	F Felix L Lucas	Concluído no prazo	3	5	3.9
Análise de Objetivos	March 16, 2022 → March 22, 2022	F Felix L Lucas	Concluído no prazo	2	2	2.6
Escopo Inicial	March 16, 2022 → March 22, 2022	F Felix L Lucas	Concluído no prazo	3	3	3.9
Pesquisa de Projetos Similares	March 16, 2022 → March 22, 2022	F Felix L Lucas	Concluído no prazo	3	1	3.9
Levantamento de Requisitos	March 23, 2022 → April 5, 2022	F Felix L Lucas	Concluído no prazo	5	7	6.5
Elaboração cronograma	March 23, 2022 → April 5, 2022	F Felix L Lucas	Concluído no prazo	6	10	7.8
Análise de Riscos	March 23, 2022 → April 5, 2022	F Felix L Lucas	Concluído no prazo	2	2	2.6
Escopo mecânico e escolha de componentes	March 23, 2022 → April 5, 2022	F Felix L Lucas	Concluído no prazo	6	6	7.8
PP: Entrega do Plano de Projeto	April 6, 2022		Entrega	30	36	39
Compra de componentes	April 6, 2022 → April 12, 2022	F Felix	Concluído no prazo	3	2	3.9
Teste das pontes-H	April 6, 2022 → April 12, 2022	F Felix	Concluído no prazo	2	1	2.6
Teste de microcontroladores	April 6, 2022 → April 12, 2022	F Felix L Lucas	Concluído no prazo	3	2	3.9
Teste de sensores	April 6, 2022 → April 12, 2022	F Felix	Concluído no prazo	2	3	2.6
Documentação do Hardware	April 6, 2022 → April 12, 2022	L Lucas	Concluído no prazo	3	3	3.9
Desenvolvimento do blog de acompanhamento	April 6, 2022 → April 12, 2022	L Lucas	Concluído no prazo	4	6	5.2

Figura 24: Cronograma do Projeto Parte 1.

Fonte: Os Autores, 2022.

👤 Name	📅 Prazo	👤 Responsável	🟡 Status	# Horas Previstas	# Gasto em Horas	Σ Previsto + Erro
📅 Desenvolvimento do blog de acompanhamento	April 6, 2022 → April 12, 2022	👤 Lucas	Concluído no prazo	4	6	5.2
🔴 E1: Site/blog de acompanhamento	April 13, 2022		Entrega	47	53	61.1
🔧 Teste dos motores 🗨 2	April 13, 2022 → April 19, 2022	👤 Lucas	Concluído com atraso	1	1	1.3
⚙️ Realização de projeto mecânico em Solid	April 13, 2022 → April 19, 2022	👤 Felix	Concluído no prazo	5	5	6.5
⚙️ Confeção do chassi do robô	April 13, 2022 → April 19, 2022	👤 Lucas 👤 Felix	Concluído no prazo	5	3	6.5
⚙️ Montagem da estrutura mecânica	April 13, 2022 → April 19, 2022	👤 Lucas	Concluído com atraso	4	5	5.2
🔴 E2: Projeto/design e montagem da estrutura mecânica	April 20, 2022		Entrega	62	67	80.6
📁 ROS - instalação e configuração inicial do software	April 20, 2022 → April 26, 2022	👤 Felix	Concluído no prazo	8	8	10.4
🔧 Projeto placa impressa	April 26, 2022 → May 3, 2022	👤 Lucas	Concluído no prazo	5	8	6.5
🔧 Confeção Placa impressa 🗨 3	April 26, 2022 → May 3, 2022	👤 Lucas	Concluído com atraso	3	20	3.9
🔧 Teste da placa 🗨 1	April 26, 2022 → May 3, 2022	👤 Felix 👤 Lucas	Concluído com atraso	2	1	2.6
🔴 E3: Projeto eletrônico e PCB. Testes de módulos isolados de hardware	April 27, 2022		Entrega	80	104	104
📁 Código: Tratamento inicial dados sensores no esp 32 seguido de envio ao ROS	April 27, 2022 → May 10, 2022	👤 Lucas	Concluído no prazo	3	3	3.9

Figura 25: Cronograma do Projeto Parte 2.

Fonte: Os Autores, 2022.

Nome	Prazo	Responsável	Status	Horas Previstas	Gasto em Horas	Previsto + Erro
■ Código: Tratamento inicial dados sensores no esp 32 seguido de envio ao ROS	April 27, 2022 → May 10, 2022	(L) Lucas	Concluído no prazo	3	3	3.9
■ Código: Fusão dados dos sensores e configuração robot localization	April 27, 2022 → May 10, 2022	(F) Felix	Concluído com atraso	4	4	5.2
■ Código: Integração dados do robot localization com navigation stack	April 27, 2022 → May 10, 2022	(F) Felix	Concluído com atraso	4	2	5.2
■ Código: Controle PID velocidade dos motores	April 27, 2022 → May 10, 2022	(L) Lucas	Concluído com atraso	5	1	6.5
■ Código: Desenvolvimento da interface gráfica	April 27, 2022 → May 10, 2022	(F) Felix	Concluído no prazo	1	1	1.3
■ Documentação do Software	April 27, 2022 → May 10, 2022	(F) Felix (L) Lucas	Concluído no prazo	2	3	2.6
● E4: Projeto do software com UML. Testes de módulos individuais de software	May 11, 2022		Entrega	98	118	127.4
■ Código: Continuação do desenvolvimento do código do ROS	May 11, 2022 → May 24, 2022	(F) Felix	Concluído com atraso	10	3	13
■ Integração mecânica + hardware + software	May 11, 2022 → May 24, 2022	(F) Felix (L) Lucas	Concluído com atraso	10	18	13
■ Testes de integração	May 11, 2022 → May 24, 2022	(F) Felix (L) Lucas	Concluído com atraso	6	9	7.8
■ Testes iniciais outdoor	May 11, 2022 → May 24, 2022	(F) Felix (L) Lucas	Concluído com atraso	4	1	5.2
■ Reavaliação da estrutura mecânica e possíveis modificações	May 11, 2022 → May 24, 2022	(L) Lucas	Concluído no prazo	10	3	13
● E5: Integração mecânica + hardware + software	May 25, 2022		Entrega	138	152	179.4
■ Análise e documentação dos resultados dos testes	May 25, 2022 → June 7, 2022	(F) Felix	Concluído com atraso	3	2	3.9
■ Testes outdoor e calibração de variáveis	May 25, 2022 → June 7, 2022	(F) Felix (L) Lucas	Concluído com atraso	7	2	9.1
■ Filmagem dos testes	May 25, 2022 → June 7, 2022	(L) Lucas	Concluído com atraso	3	1	3.9

Figura 26: Cronograma do Projeto Parte 3.

Fonte: Os Autores, 2022.

■ Filmagem dos testes	May 25, 2022 → June 7, 2022	(L) Lucas	Concluído com atraso	3	1	3.9
■ Ajustes e possíveis melhorias	May 25, 2022 → June 7, 2022	(F) Felix (L) Lucas	Concluído com atraso	10	5	13
● DP: Demonstração do funcionamento do protótipo	June 8, 2022		Entrega	161	162	209.3
■ Análise de resultados obtidos e conclusões finais	June 8, 2022 → June 21, 2022	(F) Felix (L) Lucas	Concluído com atraso	5	5	6.5
■ Desenvolvimento do relatório final	June 8, 2022 → June 21, 2022	(F) Felix (L) Lucas	Concluído com atraso	15	20	19.5
● RT: Relatório Técnico	June 22, 2022		Entrega	181	187	235.3
■ Ajustes e solução de problemas encontrados ao longo dos testes	June 22, 2022 → June 28, 2022	(F) Felix (L) Lucas	A fazer	10		13
● DF: Avaliação por uma banca de professores	June 29, 2022 → July 6, 2022		Entrega	191		248.3

Figura 27: Cronograma do Projeto Parte 4.

Fonte: Os Autores, 2022.

5.2 CUSTOS

A Tabela 1 apresenta os detalhes dos custos diretos envolvidos no projeto, como por exemplo, na compra dos componentes e no desenvolvimento das demais etapas. Vale lembrar que não foram considerados os custos dos diversos outros componentes obtidos e descartados nas etapas de testes do circuito eletrônico.

Tabela 1: Quantidade e Preços dos componentes utilizados no Projeto

Item	Quantidade	Preço(R\$)	Total(R\$)
HC-SR04	3	14,90	44,70
VL53L0X	2	44,00	88,00
MPU-9250	1	100,00	100,00
VNH2SP30-E	2	114,50	229,00
GPS GY-NEO6MV2	1	50,00	50,00
Motor DC	4	186,82	747,28
Chassi em MDF	1	40,00	40,00
Roda	4	13,00	52,00
Mancal KFL p/ eixo 8mm	4	17,90	71,60
Acoplamento Flexível 8x8mm	4	11,90	47,60
Parafuso Sextavado M8 x 80mm	4	4,00	16,00
Porca p/ Parafuso Sextavado M8	8	0,50	4,00
Componentes Eletrônicos Diversos	1	100,00	100,00
Impressões de Circuito	8	5,00	40,00
Placa de Fenolite 20x30	1	22,00	22,00
Placa de Fenolite 15x30	1	16,00	16,00
Placa de Fibra de Vidro 20x30	1	36,00	36,00
Kit para confecção de PCI	2	60,00	120,00
RaspBerry Pi 4B 2GB	1	800,00	800,00
ESP32 DevKitV1	2	30,00	60,00
Total			2648,18

Fonte: Os Autores, 2022.

6 CONCLUSÕES

6.1 CONCLUSÕES

Durante o desenvolvimento das etapas de planejamento e montagem da estrutura foram identificadas uma série de problemas. As principais etapas relacionadas, foram as de testagem de motores e demais componentes eletrônicos, onde a presença de ruídos e outras interferências dificultaram, consideravelmente, a aferição de medidas e demais parâmetros de funcionamento. Estas etapas foram fundamentais para o melhor entendimento da estrutura e para o desenvolvimento da melhor configuração possível para a aplicação. Durante este processo, foram feitos testes com técnicas utilizando capacitores em paralelo com a alimentação dos motores, com capacitores nos canais de comunicação entre o microcontrolador ESP32 e a ponte H, mas o que realmente funcionou foi utilizar um optoacoplador para cada conexão entre microcontrolador e ponte H. Outro problema enfrentado e não solucionado foi em relação ao uso do sensor MPU9250, para seu uso é necessário calibrar o sensor, salvar os dados e testar o funcionamento, mas uma calibração incorreta acarreta em diversos problemas na leitura dos dados e até descobrir que o problema foi na calibração e não no código leva muito tempo, sendo necessárias novas calibrações e testes. Outro problema foi a falta de conhecimento do sistema ROS por parte da equipe, que apesar do estudo feito e adquirido, o tempo empregado no estudo não foi suficiente para o aprendizado e posterior utilização e implementação no projeto. Por esse motivo o uso de ROS foi retirado do projeto durante a fase de integração da mecânica e hardware com o software, sendo utilizados códigos na linguagem Python para tratamento e processamento das rotinas necessárias para o funcionamento do robô em ambientes controlados.

Devido aos obstáculos e dificuldades enfrentadas pela equipe e também pela dificuldade inesperada que o projeto apresentou, a equipe não conseguiu finalizar o projeto dentro do prazo previsto pelo cronograma.

6.2 TRABALHOS FUTUROS

O desenvolvimento de veículos autônomos representa uma das principais áreas de interesse e crescimento de mercado, indicando que o estudo e utilização destas estruturas serve de base de conhecimento para uma infinidade de aplicações. Desta forma é perfeitamente possível cogitar o desenvolvimento contínuo da aplicação, através da implementação de novas funcionalidades, tais como fazer com que o robô recupere a posição correta em caso de capotamento ou que consiga sair de uma situação na qual fique preso ou enroscado.

REFERÊNCIAS

BAÚ DA ELETRÔNICA. **Monster Motor Shield**. [S.l.], 2022. Acessado em: 12/06/2022. Disponível em: <<https://www.baudaeletronica.com.br/monster-motor-shield.html>>.

ESPRESSIF SYSTEMS. **ESP32 Datasheet**. [S.l.], out. 2016. Acessado em: 11/06/2022. Disponível em: <https://www.adrobotica.com/wp-content/uploads/2019/07/Datasheet_ESP8266_esp32_en.pdf>.

GOOGLE. **Angular**. [S.l.], 2010–2022. Acessado em: 23/06/2022. Disponível em: <<https://angular.io/guide/what-is-angular>>.

INVENSENSE. **MPU-9250 Product Specification**. [S.l.], jun 2016. Acessado em: 12/06/2022. Disponível em: <<https://3cfeqx1hf82y3xcoull08ihx-wpengine.netdna-ssl.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>>.

MORGAN, E. J. **HC-SR04 Ultrasonic Sensor**. [S.l.], nov 2014. Acessado em: 12/06/2022. Disponível em: <<https://pdf1.alldatasheet.com/datasheet-pdf/view/1132203/ETC2/HC-SR04.html>>.

PALLETS. **Flask**. [S.l.], 2010. Acessado em: 23/06/2022. Disponível em: <<https://flask.palletsprojects.com/en/2.0.x/>>.

RASPBERRY PI. **Raspberry Pi 4 Model B**. [S.l.], jun. 2019. Acessado em: 11/06/2022. Disponível em: <<https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>>.

ROBOCORE. **Regras Robô Trekking**. [S.l.], may 2018. Acessado em: 13/06/2022. Disponível em: <https://www.robocore.net/upload/attachments/regras_robotrekking_2018_05_06_172.pdf>.

ROBOCORE. **Motor com Caixa de Redução 12V 1600RPM**. [S.l.], 2022. Acessado em: 13/06/2022. Disponível em: <<https://www.robocore.net/motor-motoredutor/motor-com-caixa-de-reducao-12v-1600rpm>>.

ROS. **ROS: Robot operating system**. [S.l.], s.d. Acessado em: 13/06/2022. Disponível em: <<http://docs.ros.org/>>.

STMICROELECTRONICS. **VL53L0X: World smallest time-of-flight ranging and gesture detection sensor**. may 2016. Acessado em: 12/06/2022. Disponível em: <<https://pdf1.alldatasheet.com/datasheet-pdf/view/948120/STMICROELECTRONICS/VL53L0X.html>>.

STMICROELECTRONICS. **VNH3SP30-E: Automotive fully integrated h-bridge motor driver**. [S.l.], 2017. Acessado em: 12/06/2022. Disponível em: <<https://www.st.com/resource/en/datasheet/vnh3sp30-e.pdf>>.

TERABEE. **Time-of-Flight principle**. [S.l.], s.d. Acessado em: 12/06/2022. Disponível em: <<https://www.terabee.com/time-of-flight-principle/>>.

U-BLOX. **NEO-6**: u-blox 6 gps modules. [S.l.], 2011. Acessado em: 12/06/2022. Disponível em: <https://content.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf>.