

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

FELIPE NEGRELLI WOLTER
HADRYAN SALLES, LUAN CARLOS KLEIN

RESCUE BOT

OFICINA DE INTEGRAÇÃO 2 – RELATÓRIO FINAL

CURITIBA

2021

FELIPE NEGRELLI WOLTER
HADRYAN SALLES, LUAN CARLOS KLEIN

RESCUE BOT

Relatório Final da disciplina Oficina 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 obtenção da aprovação na disciplina.

Orientador: Prof. Dr. César Manuel Vargas Benítez
Prof. Dr. Heitor S. Lopes

CURITIBA

2021

Este trabalho é dedicado a todas as vítimas de desabamentos e também às pessoas que perderam a vida pelo Covid 19.

AGRADECIMENTOS

Agradecimento especial as nossas famílias que nos auxiliaram no desenvolvimento desse projeto, uma vez que ele foi desenvolvimento de maneira integral em nossas casas. Agradecimento aos professores Cesar M. V. Benitez e Heitor S. Lopes, por todo o apoio e ensinamentos durante a realização do projeto.

RESUMO

. RESCUE BOT. 51 f. Oficina de Integração 2 – Relatório Final – Curso de Engenharia de Computação, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2021.

O projeto tratado neste documento consiste no desenvolvimento de um robô que possa ser utilizado para o auxílio na localização e resgate de vítimas em desabamentos, sendo controlado por um controle remoto por uma pessoa na estação, tornando o processo mais seguro para os profissionais de resgate. Suas funções são a captação e transmissão de imagens em tempo real para a estação, permitir a comunicação via áudio com a estação, monitorar variáveis do ambiente como temperatura e fumaça e conseguir penetrar em espaços pequenos e irregulares. Para a construção de tal, foi realizado os projeto de hardware, software e mecânica, e posteriormente feita a integração de todas elas. Ao final, foi possível obter um robô que atendia os objetivos iniciais e suas especificações.

Palavras-chave: desabamentos, resgate, robô

ABSTRACT

. RESCUE BOT. 51 f. Oficina de Integração 2 – Relatório Final – Curso de Engenharia de Computação, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2021.

The project subject of this report consists of the development of a robot to be utilized for helping with locating and rescuing victims of building collapses, being controlled from afar by an expert through a remote controller, making the process safer for the professionals involved. Its functions are the capture and transmission of real time images to the station, provide audio communication between the station and the victim, monitor environment characteristics such as temperature and smoke and penetrate in small and irregular spaces. For its construction, hardware, software and mechanical projects were made and were later integrated. The end result was a robot that could execute all of the tasks that were initially specified.

Keywords: collapses, rescue, robot

LISTA DE FIGURAS

FIGURA 1	– GPIOs do Raspberri pi 3B+ (ETRONICWINGS, 2019)	16
FIGURA 2	– Comunicação UART (MENDONCA, 2015)	17
FIGURA 3	– Comunicação via soquetes (PANTUZA, 2017)	17
FIGURA 4	– Diagrama de blocos do projeto	22
FIGURA 5	– Modelo 3D do robô sem a cobertura	23
FIGURA 6	– Modelo 3D do robô de trás com a cobertura	23
FIGURA 7	– Modelo 3D do robô de frente com a cobertura	23
FIGURA 8	– Diagrama de blocos do hardware	24
FIGURA 9	– Esquemático do circuito	25
FIGURA 10	– Esquemático da placa desenvolvida	26
FIGURA 11	– Roteamento da placa	27
FIGURA 12	– Diagrama de casos de uso do projeto.	27
FIGURA 13	– Diagrama de sequências para o caso de uso UC001.	32
FIGURA 14	– Diagrama de sequências para o caso de uso UC002.	32
FIGURA 15	– Diagrama de sequências para o caso de uso UC003.	32
FIGURA 16	– Diagrama de estados e transições de alto nível para o <i>software</i> do robô.	33
FIGURA 17	– Diagrama de estados e transições de alto nível para o <i>software</i> da base.	33
FIGURA 18	– Diagrama de classes.	34
FIGURA 19	– Fluxograma da <i>thread</i> de recebimento dos dados do controle.	35
FIGURA 20	– Fluxograma da <i>thread</i> de envio dos dados do áudio.	35
FIGURA 21	– Fluxograma da <i>thread</i> de recebimento dos dados do áudio.	36
FIGURA 22	– Fluxograma da <i>thread</i> principal do <i>software</i> do robô.	36
FIGURA 23	– Fluxograma da <i>thread</i> principal do <i>software</i> da estação base.	37
FIGURA 24	– Imagem do robô físico construído	40
FIGURA 25	– Imagem do robô físico construído	40
FIGURA 26	– Imagem do robô superando obstáculos	41
FIGURA 27	– Interface do robô, que é exibida na base	42
FIGURA 28	– Tarefas planejadas para o entregável 1	43
FIGURA 29	– Tarefas planejadas para o entregável 2 - Parte 1	43
FIGURA 30	– Tarefas planejadas para o entregável 2 - Parte 2	44
FIGURA 31	– Tarefas planejadas para o entregável 3	44
FIGURA 32	– Tarefas planejadas para o entregável 4	44
FIGURA 33	– Tarefas planejadas para o entregável 5	44
FIGURA 34	– Diagrama de Gantt de todas as atividades - Parte 1	45
FIGURA 35	– Diagrama de Gantt de todas as atividades - Parte 2	46
FIGURA 36	– Gráfico de horas trabalhadas e estimadas semanalmente por integrante	47

LISTA DE TABELAS

TABELA 1	– Tabela de conexões	26
TABELA 2	– Diagrama de caso de uso 1: UC001	28
TABELA 3	– Especificação do caso de uso 1: UC001	29
TABELA 4	– Diagrama de caso de uso 2: UC002	29
TABELA 5	– Especificação do caso de uso 2: UC002	30
TABELA 6	– Diagrama de caso de uso 3: UC003	30
TABELA 7	– Especificação do caso de uso 3: UC003	31
TABELA 8	– Tabela de componentes e preços	48

SUMÁRIO

1	INTRODUÇÃO	13
1.1	MOTIVAÇÃO	13
1.2	OBJETIVOS	13
1.2.1	Objetivo geral	13
1.2.2	Objetivos específicos	13
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	PLANEJAMENTO	15
2.2	RASBERRY PI 3B+	15
2.3	COMUNICAÇÃO SERIAL UART	16
2.4	COMUNICAÇÃO POR SOQUETES DE REDE	17
2.5	THREADS	17
2.5.1	Amplificadores	18
2.6	MOTORES	18
2.6.1	Motor DC e drivers	18
2.6.2	Servomotores	19
3	METODOLOGIA	20
3.1	VISÃO GERAL	20
3.2	PROJETO MECÂNICO	22
3.3	PROJETO DE HARDWARE	23
3.4	PROJETO DE SOFTWARE	27
3.5	INTEGRAÇÃO	37
4	EXPERIMENTOS E RESULTADOS	39
5	CRONOGRAMA E CUSTOS DO PROJETO	43
5.1	CRONOGRAMA	43
5.2	CUSTOS	47
6	CONCLUSÕES	49
6.1	CONCLUSÕES	49
6.2	TRABALHOS FUTUROS	49
	REFERÊNCIAS	51

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Quando ocorrem catástrofes, sejam elas naturais ou causadas diretamente por seres humanos, as estruturas físicas frequentemente são afetadas, como casas, prédios ou galpões, deixando assim uma enorme pilha de escombros. Com isso, pessoas podem acabar sendo soterradas, ficando imobilizadas e sem poder de reação, com sua vida dependendo de serem encontradas pela equipe de resgate (normalmente bombeiros). Entretanto, por serem situações extremas, o resgate encontra muita dificuldade de locomoção, sendo um fator de alto risco de vida para os próprios membros da equipe. Visando apresentar uma alternativa a esse cenário, o projeto a ser desenvolvido busca auxiliar o processo de encontrar vítimas soterradas, de maneira mais rápida e segura.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

O principal objetivo do Rescue Bot é ser um robô para ser utilizado em situações de catástrofes (desmoronamentos e queda de construções físicas, como casas e prédios) para auxiliar a encontrar vítimas soterradas. Ele busca ser uma ferramenta de uso prático para as equipes de resgate, possibilitando entrar em lugares pequenos e perigosos, coletando informações do ambiente e transmitindo-as a equipe, tornando seu trabalho mais ágil, seguro e eficiente.

1.2.2 OBJETIVOS ESPECÍFICOS

O produto do projeto apresentado tem por objetivos específicos:

- Transmitir informações (tais como presença de gases inflamáveis e temperatura) do ambiente de maneira rápida à equipe de resgate

- Ser um robô pequeno e ágil, possibilitando passar por obstáculos e entrar em lugares com pouco espaço
- Mapear o ambiente pelos quais ele passar, identificando obstáculos
- Enviar e transmitir áudios, para no momento em que uma pessoa for encontrada, a equipe de socorro possa coletar informações da vítima e enviar orientações a ela

2 FUNDAMENTAÇÃO TEÓRICA

Para a elaboração desse projeto, foi necessária a aplicação de diversos conhecimentos adquiridos no decorrer do curso e fora dele. Nas seções abaixo serão detalhados os fundamentos teóricos que foram necessários e quais suas aplicações no projeto.

2.1 PLANEJAMENTO

Um planejamento detalhado e preciso é um componente essencial de um projeto e um dos principais critérios que determinam seu sucesso ou fracasso. Por esse motivo, para a elaboração do Rescue Bot, foram utilizados os conhecimentos e técnicas adquiridos nas disciplinas de Análise e Projeto de Sistemas bem como em Engenharia de Software, com destaque para a especificação de requisitos, elaboração de diagramas em UML e a elaboração do cronograma.

2.2 RASBERRY PI 3B+

O Raspberry Pi é um computador de tamanho reduzido utilizado no aprendizado e em aplicações relacionadas a área de computação. O modelo 3B+ possui um processador Cortex-A53 (ARMv8) 64-bit, 1GB de memória RAM, GPIO de GPIOs (muitos deles com funções alternativas), entrada ethernet, 4 entradas USB, uma saída de áudio P3, uma entrada micro USB entre outros (FILIFELOP, 2018). Na Figura 1, há uma representação da placa do Raspberry, com um detalhamento dos GPIOs disponíveis.

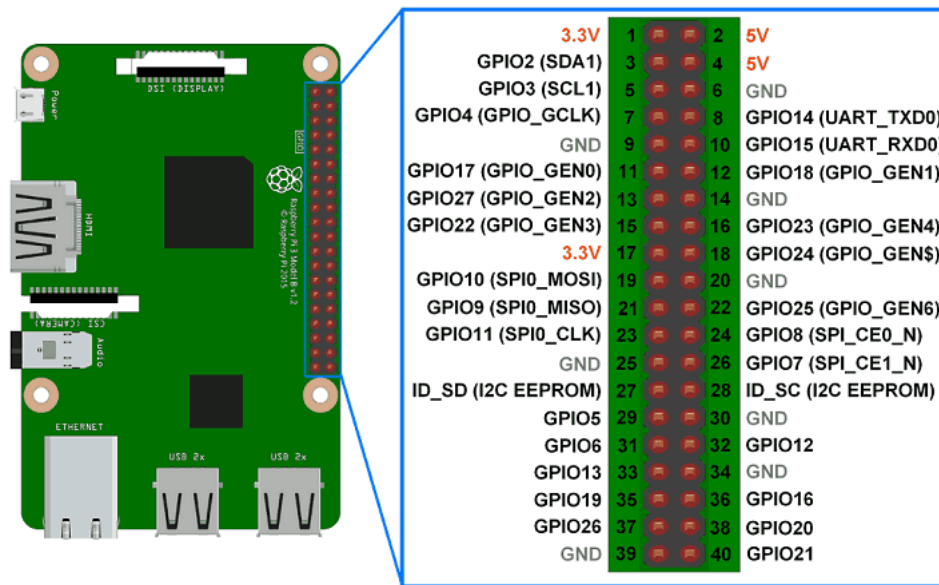


Figura 1: GPIOs do Raspberri pi 3B+ (ETRONICWINGS, 2019)

2.3 COMUNICAÇÃO SERIAL UART

A compreensão de princípios de comunicação UART, aprendida na disciplina de Sistemas Microcontrolados, foi necessária pois foram utilizados componentes que utilizam esse modelo de comunicação serial para a troca de informações com o microcontrolador. UART é um modelo de comunicação serial entre dois dispositivos que opera em modo *full-duplex* (MENDONCA, 2015), que significa que ambos os dispositivos podem enviar e receber dados simultaneamente (MENDONCA, 2015). Os canais utilizados para a comunicação são:

- TX: Envio dos dados
- RX: Recebimento dos dados

Ao utilizar esses dois canais, ressalta-se que o TX de um dispositivo deve ser conectado ao RX do outro, e vice-versa (MENDONCA, 2015).

A velocidade da comunicação deve ser a mesma em ambos os lados e é chamada de *baud rate*, dado em bits por segundo. Enquanto um dispositivo não estiver enviando dados, ele fica no chamado *idle state*, onde sua saída é em nível lógico alto (MENDONCA, 2015). Antes de iniciar o envio de uma mensagem (normalmente de até 8 ou 9 bits), o emissor envia um *start bit*, em nível lógico baixo. Ao encerrar a transmissão, o dispositivo envia o *stop bit*, em nível lógico alto (MENDONCA, 2015). A mensagem pode ou não conter um bit de paridade (MENDONCA, 2015). O processo é ilustrado na Figura 2

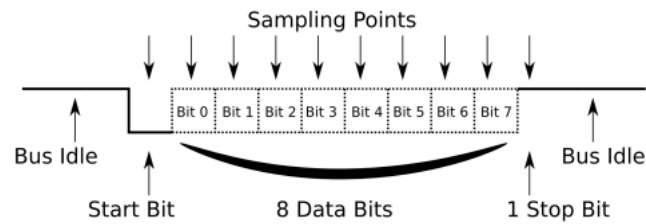


Figura 2: Comunicação UART (MENDONCA, 2015)

2.4 COMUNICAÇÃO POR SOQUETES DE REDE

Foi decidido que a comunicação entre o robô e a estação seriam feitos via soquetes de rede, conceito abordado na disciplina de Comunicação de Dados. Em uma comunicação via rede entre dois aplicativos, um soquete é uma API que faz uma abstração da camada de rede, facilitando a lógica de comunicação entre os dois terminais, tornando-a semelhante a uma comunicação direta, sem a passagem de informação via rede (PANTUZA, 2017). Em uma pilha de protocolos TCP/IP, os soquetes se localizam entre a camada de aplicação e de transporte (PANTUZA, 2017). O fluxo de uma comunicação via soquetes de rede é ilustrado na Figura 3.

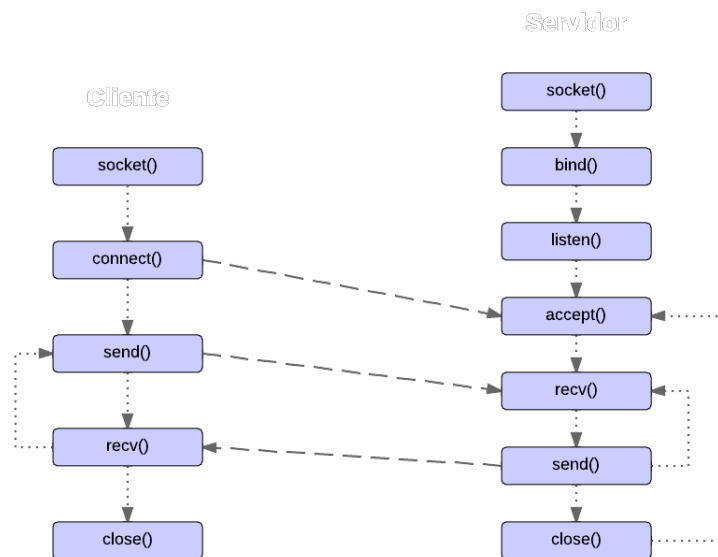


Figura 3: Comunicação via soquetes (PANTUZA, 2017)

2.5 THREADS

Uma vez que o projeto desenvolvido possui múltiplas partes independentes nas quais há a necessidade de espera por uma informação, foram utilizadas *threads* para evitar o desperdício de tempo que seria causado por esse período onde há a espera por uma entrada.

Threads foi um conteúdo abordado na disciplina de Sistemas operacionais e constituem um fluxo independente de execução de código, contendo um pequeno contexto (armazenamento do estado da tarefa) (MAZIERO, 2019). *Threads* de um mesmo processo compartilham o mesmo espaço de memória, o que facilita a comunicação entre elas, porém abre espaço para inconsistência nos dados caso não sejam utilizadas práticas como a utilização de semáforos em seções críticas, logo, exigem um maior cuidado para a sua implementação (MAZIERO, 2019).

2.5.1 AMPLIFICADORES

No projeto, foi utilizado um circuito amplificador para áudio. O tópico de amplificadores foi abordado na disciplina de Eletrônica Geral 1 e aprofundado na disciplina de Eletrônica Geral 2.

Um amplificador consiste, de maneira simplificada, em um componente que irá amplificar um sinal elétrico recebido (CODELLOS, 2013). O quociente entre a tensão de saída e de entrada é denominado ganho. Ao amplificar o sinal, o amplificador pode ocasionar uma distorção harmônica no sinal de saída. Por exemplo, ao receber uma onda senoidal perfeita, a saída pode apresentar discrepâncias em relação a entrada. Quanto menor a distorção harmônica obtida, maior a qualidade do amplificador. Ao utilizar amplificadores em áudio, para a qualidade de som ser boa, a distorção harmônica deve ser inferior a 1% (CODELLOS, 2013).

Existem diversas classes de amplificadores, cada uma com características distintas. Para o presente projeto, foi utilizado o amplificador PAM8403, que é da classe D. Nessa classe, os amplificadores tem uma eficiência energética muito boa (em torno de 90%) (CODELLOS, 2013).

2.6 MOTORES

2.6.1 MOTOR DC E DRIVERS

Motores DC e seu interfaceamento com microcontroladores foi um dos tópicos estudados na matéria de Sistemas Microcontrolados. Esses motores são atuadores, que transformam energia elétrica em energia mecânica. São cargas analógicas, ou seja, o torque e a velocidade variam conforme se varia a tensão e a corrente sobre ele (WIKIPEDIA, 2021). Para serem acionados e controlados por meio de um microcontrolador, há a necessidade de um driver de corrente, pois sua alta demanda de corrente e o fato de ser uma carga indutiva podem causar danos ao micro controlador (SANTOS, 2020).

Um driver ponte H é um circuito que, além de proteger o microcontrolador e proporcionar uma interface segura com o motor, permite a polarização do mesmo em ambos os sentidos (THOMSEN, 2013).

2.6.2 SERVOMOTORES

Servomotores, estudados na disciplina Sistemas Microcontrolados, são motores DC realimentados em malha fechada para os quais o posicionamento depende de um sinal de entrada, como por exemplo a porcentagem de tempo em nível alto de um PWM em uma frequência específica (WIKIPEDIA, 2020).

3 METODOLOGIA

3.1 VISÃO GERAL

Inicialmente, definiu-se os requisitos necessários para a construção do projeto, para que os objetivos iniciais pudessem ser cumpridos de maneira efetiva. Após análises e discussões, os requisitos definidos para o projeto puderam ser divididos em duas partes distintas: Requisitos da Estação (que é referente a parte na qual a equipe de resgate estará), enquanto que a outra consiste no Robô, que é o Robô propriamente dito. Assim, os requisitos definidos foram:

Requisitos Funcionais do Robô

- **RF01:** Se comunicar bilateralmente com a estação
- **RF02:** Conseguir se movimentar em terrenos irregulares
- **RF03:** Ter capacidade de iluminar o ambiente
- **RF04:** Ter o movimento controlado pela estação
- **RF05:** Ter a rotação da câmera controlada pela estação
- **RF06:** Transmitir as imagens da câmera para a estação
- **RF07:** Transmitir dados sobre a qualidade do ar (gases inflamáveis, fumaça, monóxido de hidrogênio) para a estação
- **RF08:** Captar e transmitir para a estação a temperatura do ambiente em que está
- **RF09:** Emitir os dados de áudio recebidos da estação
- **RF10:** Captar e transmitir para a estação os sons ao seu redor
- **RF11:** Transmitir os dados de aceleração e orientação para a estação
- **RF12:** Ter uma reserva de energia caso a alimentação principal seja interrompida

Requisitos Funcionais da Estação

- **RF13:** Se comunicar bilateralmente com o robô
- **RF14:** Captar os sons emitidos pelo usuário e transmiti-los para o robô
- **RF15:** Receber os comandos de controle de movimento do robô
- **RF16:** Receber os comandos de controle da câmera
- **RF17:** Mapear o percurso do robô
- **RF18:** Mapear os obstáculos identificados pelo robô
- **RF19:** Mostrar ao usuário os dados do ar e da temperatura
- **RF20:** Mostrar ao usuário o percurso percorrido pelo robô
- **RF21:** Mostrar ao usuário as imagens transmitidas pelo robô
- **RF22:** Captar áudio e enviá-lo ao robô
- **RF23:** Emitir os dados de áudio recebidos do robô
- **RF24:** Fornecer ao usuário a opção de iluminar as proximidades do robô
- **RF25:** Ter uma interface gráfica com as informações recebidas do robô (dashboard)

Requisitos Não Funcionais do Robô

- **RN01:** A comunicação com o notebook deve ser na rede local
- **RN02:** O robô deverá ser conectado à estação por um cabo de rede Ethernet
- **RN03:** O robô deverá carregar uma fonte de luz (LEDs)
- **RN04:** O robô deverá se locomover através de um chassi de esteiras
- **RN05:** O robô deverá ter um tamanho reduzido
- **RN06:** O robô deverá ser leve
- **RN07:** O robô deverá carregar uma bateria reserva
- **RN08:** Os sensores e demais componentes devem ser fixados sobre o chassi

- **RN09:** O robô será alimentado via cabo
- **RN10:** Os cabos devem ser fortemente conectados a estrutura, permitindo que o robô seja recuperado ao se puxar o cabo em caso problemas
- **RN11:** O script para a execução do robô será programada na linguagem python

Requisitos Não Funcionais da Estação

- **RN12:** O script da estação será programado em python
- **RN13:** A interface visual será criada utilizando a biblioteca Qt

A equipe também realizou a construção do diagrama de blocos, presente na Figura 4, o qual demonstra de maneira geral como será o projeto e a integração das partes envolvidas. Observa-se que o robô contém um Raspberry Pi, que controla os diversos atuadores e coleta informações provenientes dos sensores. Além disso, ele se comunica via cabo com a equipe de resgate, transmitindo por ele as informações coletadas, e recebe comandos da estação e a energia necessária para a alimentação.

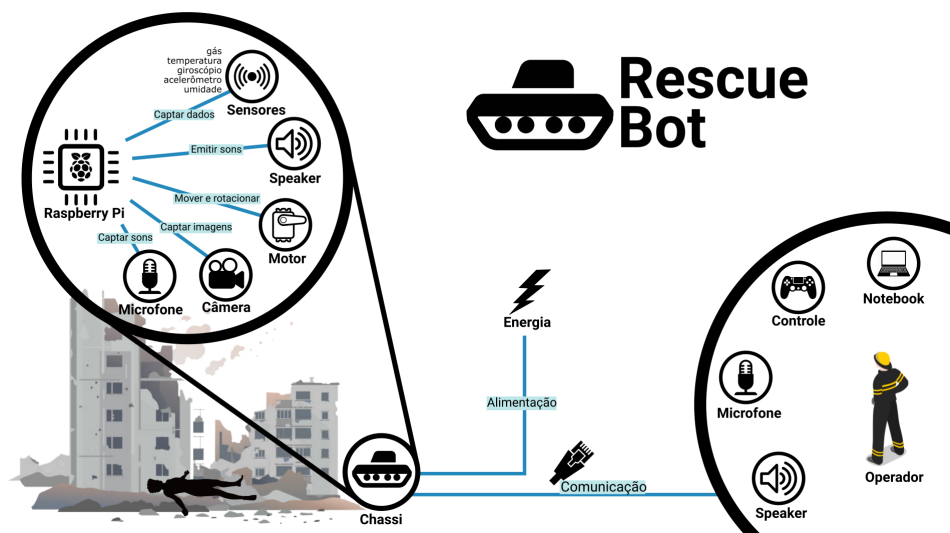


Figura 4: Diagrama de blocos do projeto

3.2 PROJETO MECÂNICO

Foi utilizado, como base para o robô um chassi tanque lagarta, pela sua capacidade de se locomover em terrenos irregulares e também por sua altura de 4 cm, o que deveria auxiliar no requisito RN05, que se refere a possuir um tamanho reduzido.

Para a proteção do robô e das conexões elétricas e componentes nele contidos, foi feita uma cobertura em papelão. Vale ressaltar que por se tratar de um protótipo, a equipe optou por utilizar o papelão que seria rápido e barato de se obter, além ser barato comparado a demais alternativas, como por exemplo, acrílico.

Foi feito um modelo em 3D da estrutura mecânica e de como seriam encaixados os componentes e módulos do robô, como ilustrado nas Figuras 5, 6, 7.

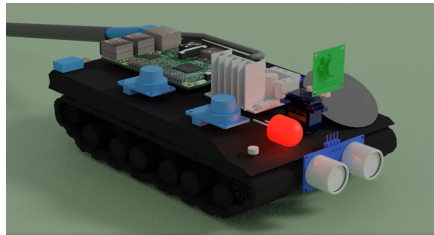


Figura 5: Modelo 3D do robô sem a cobertura

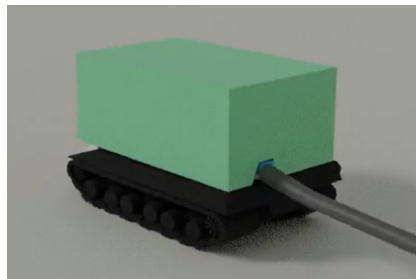


Figura 6: Modelo 3D do robô de trás com a cobertura

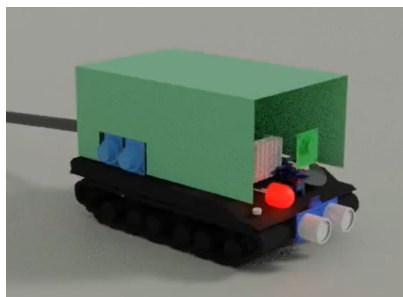


Figura 7: Modelo 3D do robô de frente com a cobertura

3.3 PROJETO DE HARDWARE

O circuito projetado e construído possui componentes específicos para cada funcionalidade do robô:

- Microfone USB, amplificador PAM8403 e alto-falante de 3W para a comunicação via áudio.
- Sensores de gás MQ2 (fumaça e gases inflamáveis) e MQ9 (monóxido de carbono) para detecção de gases.
- DHT-11 para detecção da temperatura e umidade.
- Câmera para captura de imagem.
- HC-04 para o mapeamento dos obstáculos.
- Servo motor SG-90 para controle do ângulo da câmera
- MPU6050 para captura de informações do estado atual do carrinho.
- Motores DC e Ponte H L298N para a locomoção do robô.

Os módulos e componentes utilizados no hardware do projeto são ilustrados na Figura 8.

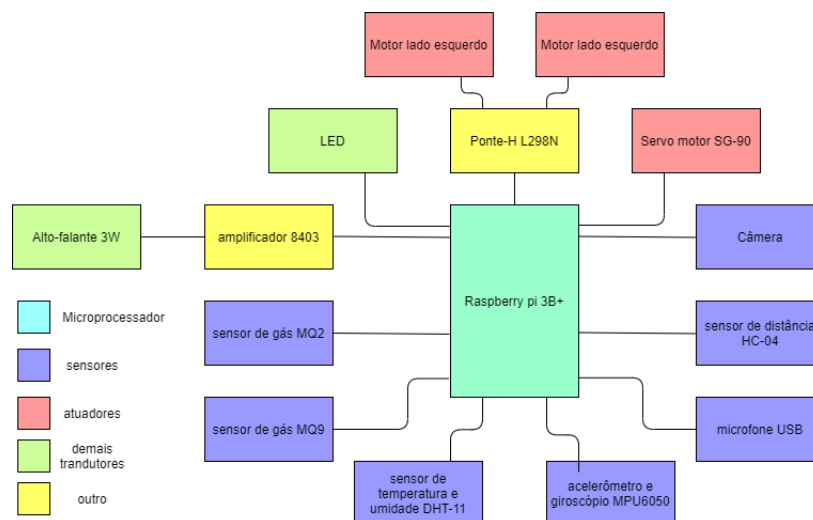


Figura 8: Diagrama de blocos do hardware

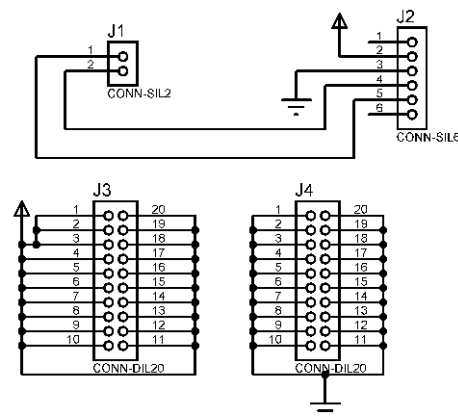
Para a alimentação dos sensores, do LED e do microcontrolador, foi utilizado um cabo de 10A de 10m, que foi soldado a um conector USB. Esse conector foi conectado a uma fonte de celular de 5V e 1,5A, que é responsável por alimentar essa parte do circuito. Para o servo motor, ponte H, motores CC, amplificador de áudio e alto-falante, foi utilizada uma bateria composta por 4 pilhas AA em série, inserida na parte inferior do robô.

A alimentação dessas duas partes do circuito foi separada para evitar que o ruído causado pelas cargas indutivas dos motores e do alto-falante exercesse influência sobre os

Tabela 1: Tabela de conexões

Sensor	Pino do Raspberry Pi
Giroscópio (SDA)	3 (GPIO2)
giroscópio (SDA)	5 (GPIO3)
HC-04 (TRIGGER)	16 (GPIO23)
HC-04 (ECHO)	18 (GPIO24)
Temperatura DHT-11	36 (GPIO36)
MQ-9 (D_O)	11 (GPIO17)
MQ-2 (D_O)	13 (GPIO27)
Mini-Microfone USB	USB
Atuador	Pino do Raspberry Pi
Motor 1 +	21 (GPIO9)
Motor 1 -	23 (GPIO11)
Motor 2 +	22 (GPIO9)
Motor 2 -	24 (GPIO11)
Servo	12 (GPIO18)
Outros	Pino do Raspberry Pi
LED	7 (GPIO23)
PAM8403 (RIGHT)	P3 Audio Jack

Foi elaborada e confeccionada uma placa de circuito impresso, com o principal objetivo de facilitar a alimentação dos módulos, aumentando o número de conexões para as fontes de alimentação. O esquemático está representado na Figura 10 e o roteamento na Figura 11.

**Figura 10: Esquemático da placa desenvolvida**

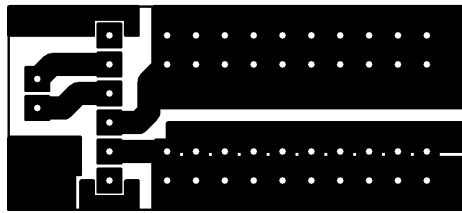


Figura 11: Roteamento da placa

3.4 PROJETO DE SOFTWARE

O projeto de *software* teve início com a modelagem UML. Primeiro, foi feito o diagrama de casos de uso, identificando dois usuários para o sistema: o membro da equipe de resgate operador do robô e a vítima soterrada, que necessita ser resgatada. O diagrama de casos de uso pode ser visto na Figura 12.

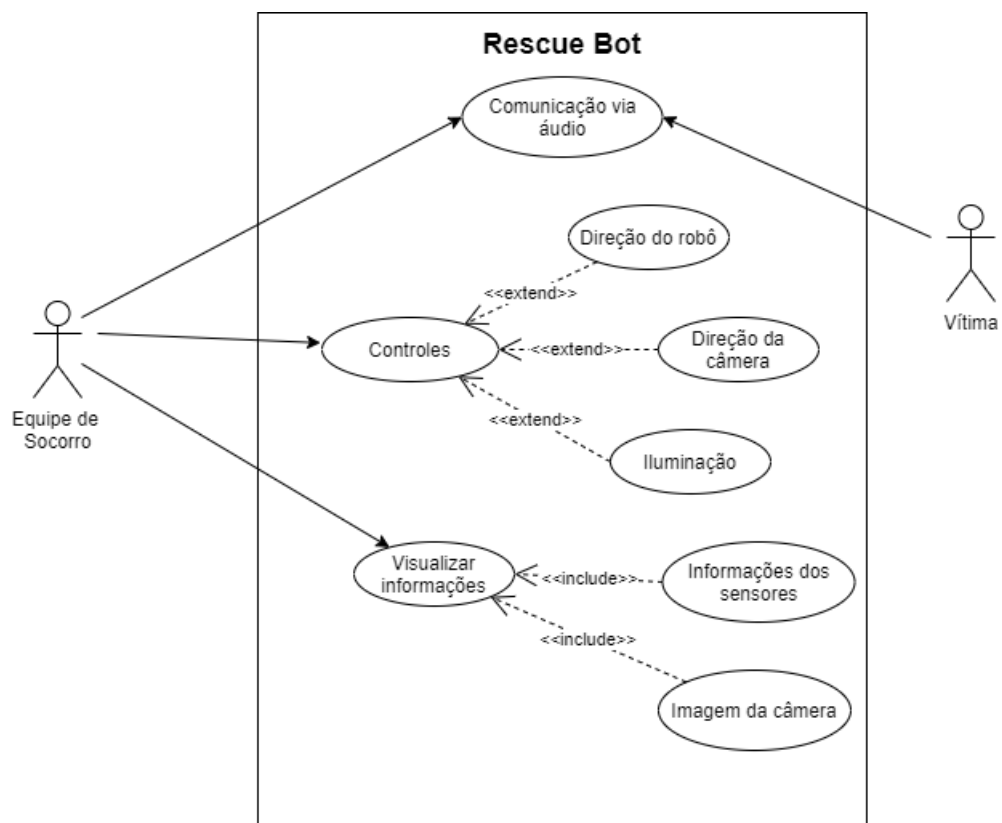


Figura 12: Diagrama de casos de uso do projeto.

A partir deste diagrama foram identificados os requisitos funcionais referentes a cada caso de uso:

- Comunicação via áudio: **“Se comunicar bilateralmente com a estação - RF1, Emitir**

os dados de áudio recebidos da estação - RF9, Captar e transmitir para a estação os sons ao seu redor - RF10, Se comunicar bilateralmente com o robô - RF13, Captar os sons emitidos pelo usuário e transmiti-los para o robô - RF14, Captar áudio e enviá-lo ao robô - RF22, Emitir os dados de áudio recebidos do robô - RF23”

- Controles:
 - Direção do robô: Ter o movimento controlado pela estação - RF4, Receber os comandos de controle de movimento do robô - RF15
 - Direção da câmera: Ter a rotação da câmera controlada pela estação - RF5, Receber os comandos de controle da câmera - RF16
 - Iluminação: Ter capacidade de iluminar o ambiente - RF3, Fornecer ao usuário a opção de iluminar as proximidades do robô - RF24
- Visualizar Informações:
 - Mapear o percurso do robô - RF17, Mapear os obstáculos identificados pelo robô - RF18, Mostrar ao usuário o percurso percorrido pelo robô - RF20
 - Informações dos sensores: Transmitir dados sobre a qualidade do ar (gases inflamáveis, fumaça, monóxido de hidrogênio) para a estação - RF7, Captar e transmitir para a estação a temperatura do ambiente em que está - RF8, Transmitir os dados de aceleração e orientação para a estação - RF11, Mostrar ao usuário os dados do ar e da temperatura - RF19, Ter uma interface gráfica com as informações recebidas do robô (dashboard) - RF25
 - Imagem da câmera: Transmitir as imagens da câmera para a estação - RF6, Mostrar ao usuário as imagens transmitidas pelo robô - RF21

Cada caso de uso foi devidamente identificado e especificado nas Tabelas 2, 3, 4, 5, 6 e 7.

Tabela 2: Diagrama de caso de uso 1: UC001

Nome	UC001: Comunicação via áudio
Atores	Atores principais: Equipe de Socorro, Vítima
Descrição	Caso de uso executado quando é feita qualquer comunicação via áudio entre o Robô e a base
Pré-condições	Robô deve estar ligado e conectado a base
Pós-condições	Informações transmitidas da base para o robô, ou vice-versa.

Tabela 3: Especificação do caso de uso 1: UC001

Fluxo Básico	
Ações dos atores	Ações do sistema
1 - Equipe de resgate fala no microfone	
	2 - O áudio é captado pelo sistema, e enviado para o robô, e emitido por ele.
3 - A vítima escuta o áudio, e responde a mensagem falando	
	4 - O sistema capta esse áudio, envia para a base, e ela emite o som para a equipe de resgate.
Regras de Negócio	
Não há neste caso de uso.	
Fluxo alternativo 1	
Ações dos atores	Ações do sistema
1 - A qualquer momento a vítima pode falar	
	2 - O robô irá captar a informação, e transmiti-la à base. Ela, por sua vez, irá reproduzir o som captado.

Tabela 4: Diagrama de caso de uso 2: UC002

Nome	UC002: Controles
Atores	Ator principal: Equipe de Socorro
Descrição	Caso de uso executado quando é realizado alguma ação de controle do robô, pela equipe de socorro
Pré-condições	Robô deve estar ligado e conectado a base
Pós-condições	Ação é executada no robô

Tabela 5: Especificação do caso de uso 2: UC002

Fluxo Básico	
Ações dos atores	Ações do sistema
1 - Equipe de resgate realiza uma ação de movimentação do robô no joystick	
	2 - O sistema interpreta esse movimento, e envia-o para o robô.
	3 - O robô recebe essa informação, e realiza o movimento recebido (nos motores)
4 - Equipe de resgate realiza uma ação de movimentação da câmera no joystick	
	5 - O sistema interpreta esse movimento, e envia-o para o robô.
	6 - O robô recebe essa informação, e realiza o movimento recebido (no motor da câmera)
7 - Equipe de resgate clica no botão de controle da iluminação	
	8 - O sistema detecta que foi apertado o botão para ativar a iluminação, e envia-a essa informação para o robô.
	9 - O robô recebe essa informação, e ativa a iluminação
Regras de Negócio	
Não há neste caso de uso.	
Fluxo alternativo 1	
Ações dos atores	Ações do sistema
1 - A qualquer momento a equipe de socorro pode realizar uma ação de controle	
	2 - O sistema irá captar a informação, e transmiti-la ao robô. Ela, por sua vez, irá reproduzir a ação de acordo com a especificação.

Tabela 6: Diagrama de caso de uso 3: UC003

Nome	UC003: Visualizar informações
Atores	Ator principal: Equipe de Socorro
Descrição	Caso de uso executado quando é realizado quando uma nova informação deve ser capturada e exibida na tela
Pré-condições	Robô deve estar ligado e conectado a base
Pós-condições	Informações disponíveis para visualização da equipe de resgate

Tabela 7: Especificação do caso de uso 3: UC003

Fluxo Básico	
Ações dos atores	Ações do sistema
	1 - O robô coleta as informações de distância frontal, de aceleração, do giroscópio, de temperatura, umidade e de qualidade do ar, e envia para a base.
	2 - A base recebe essas informações, e as exibe na tela;
	3 - O sistema utiliza as informações de distância frontal e de movimentação para construir a trajetória do robô, e identificar obstáculos. O Sistema exibe essas informações na tela.
4 - A equipe de resgate pode visualizar essas informações, e interagir com a trajetória mapeada, observando-a por vários ângulos.	
	5 - O robô captura a imagem da câmera, e a envia para a base.
	6 - A base recebe a imagem, e a exibe para a equipe de resgate.
7 - A equipe de resgate pode visualizar a imagem capturada pelo robô.	
Regras de Negócio	
Não há neste caso de uso.	
Fluxo alternativo 1	
Ações dos atores	Ações do sistema
Não há neste caso de uso.	

Após identificar e especificar os casos de uso para o *software*, foram feitos os diagramas de sequência para cada caso de uso, que podem ser vistos nas Figuras 13, 14 e 15.

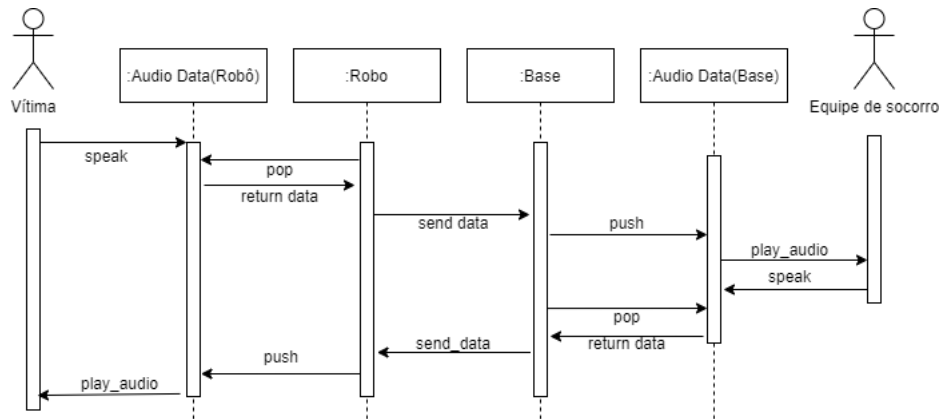


Figura 13: Diagrama de seqüências para o caso de uso UC001.

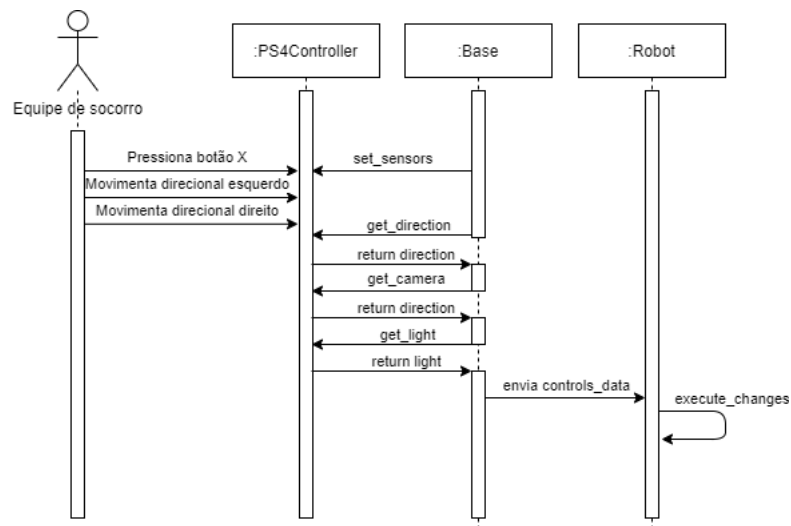


Figura 14: Diagrama de seqüências para o caso de uso UC002.

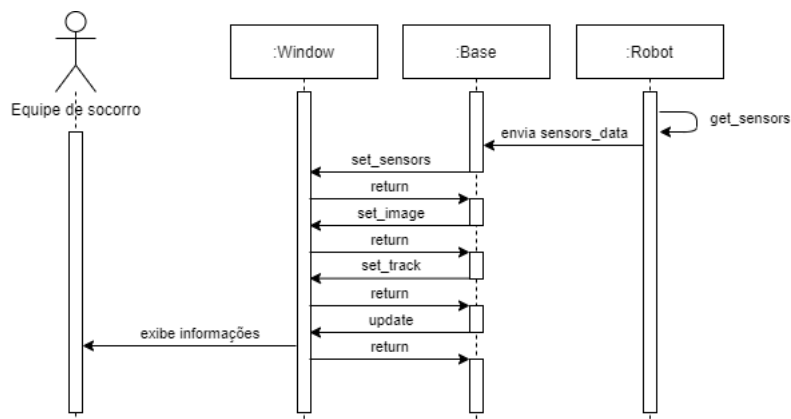


Figura 15: Diagrama de seqüências para o caso de uso UC003.

Com isso, foi identificado a necessidade de duas partes principais de *software*: O programa para ser executado no Raspberry (robô) e o programa para ser executado no computador principal (base). Para ambos esses programas foram feitos diagramas de estados e transições de alto nível, mostrando a rotina de execução que deveria ser implementada por cada um deles. Esses diagramas de estados e transições podem ser vistos nas Figuras 16 e 17

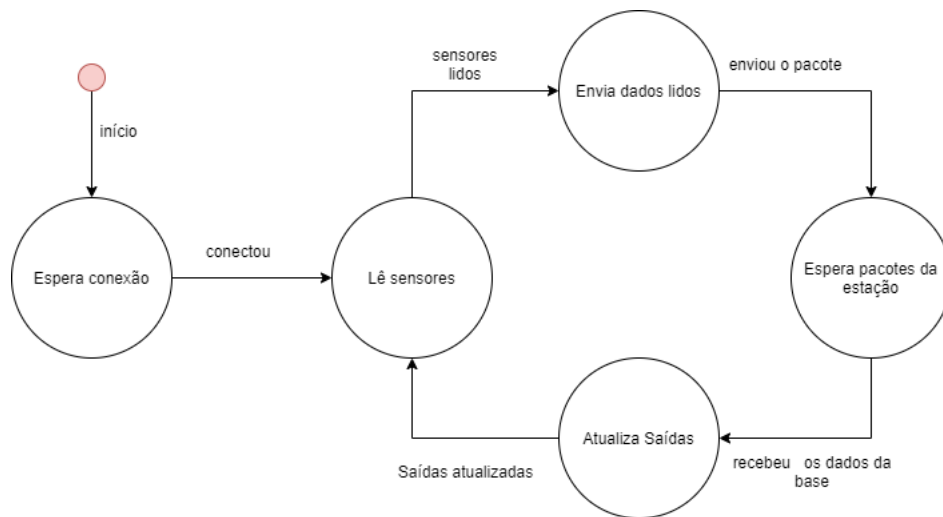


Figura 16: Diagrama de estados e transições de alto nível para o *software* do robô.

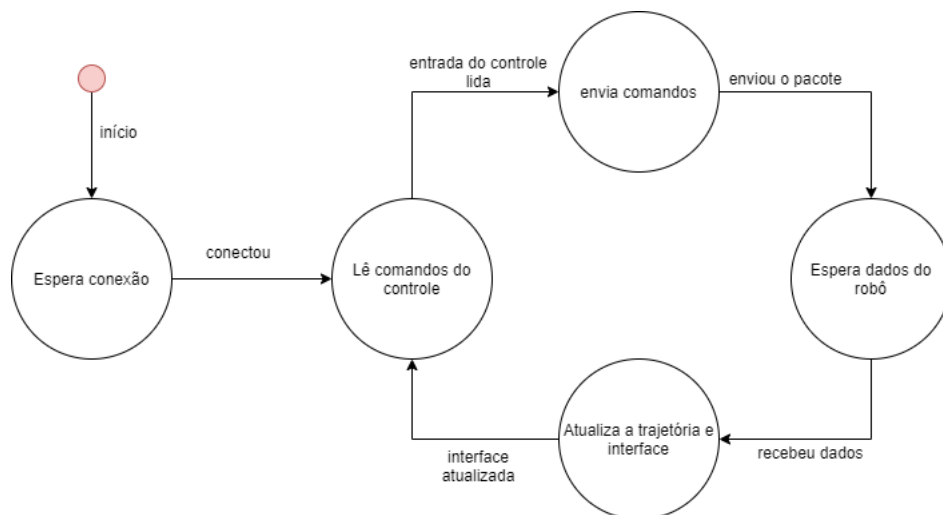


Figura 17: Diagrama de estados e transições de alto nível para o *software* da base.

Então, utilizando os diagramas de estados e transições, de sequência e de casos de uso, foi possível modelar o diagrama de classes para o *software*, que pode ser visto na Figura 18.

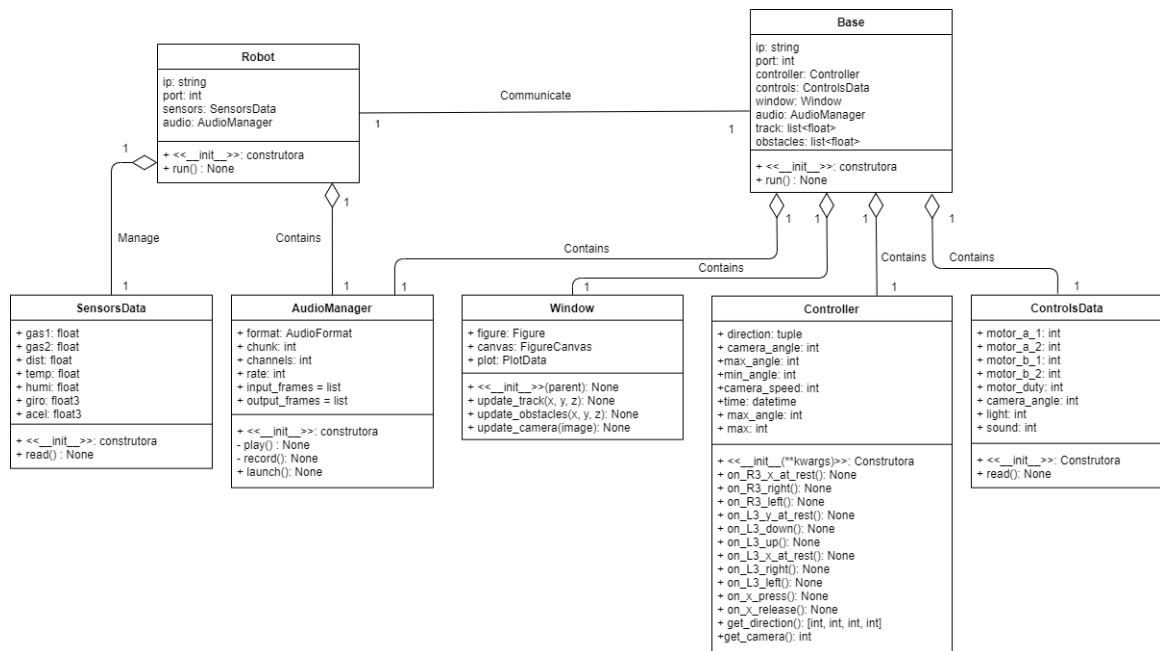


Figura 18: Diagrama de classes.

Pela quantidade de sensores e atuadores envolvidos no projeto, além da necessidade de manter um sistema em tempo real com uma taxa de atualização constante, a implementação do *software* foi separada em múltiplas *threads* para cada parte importante do *software*. Os fluxogramas para cada uma dessas *threads* foram descritos nas Figuras 19, 20, 21, 22 e 23 permitindo uma visão mais específica da rotina de execução para cada parte do *software*.

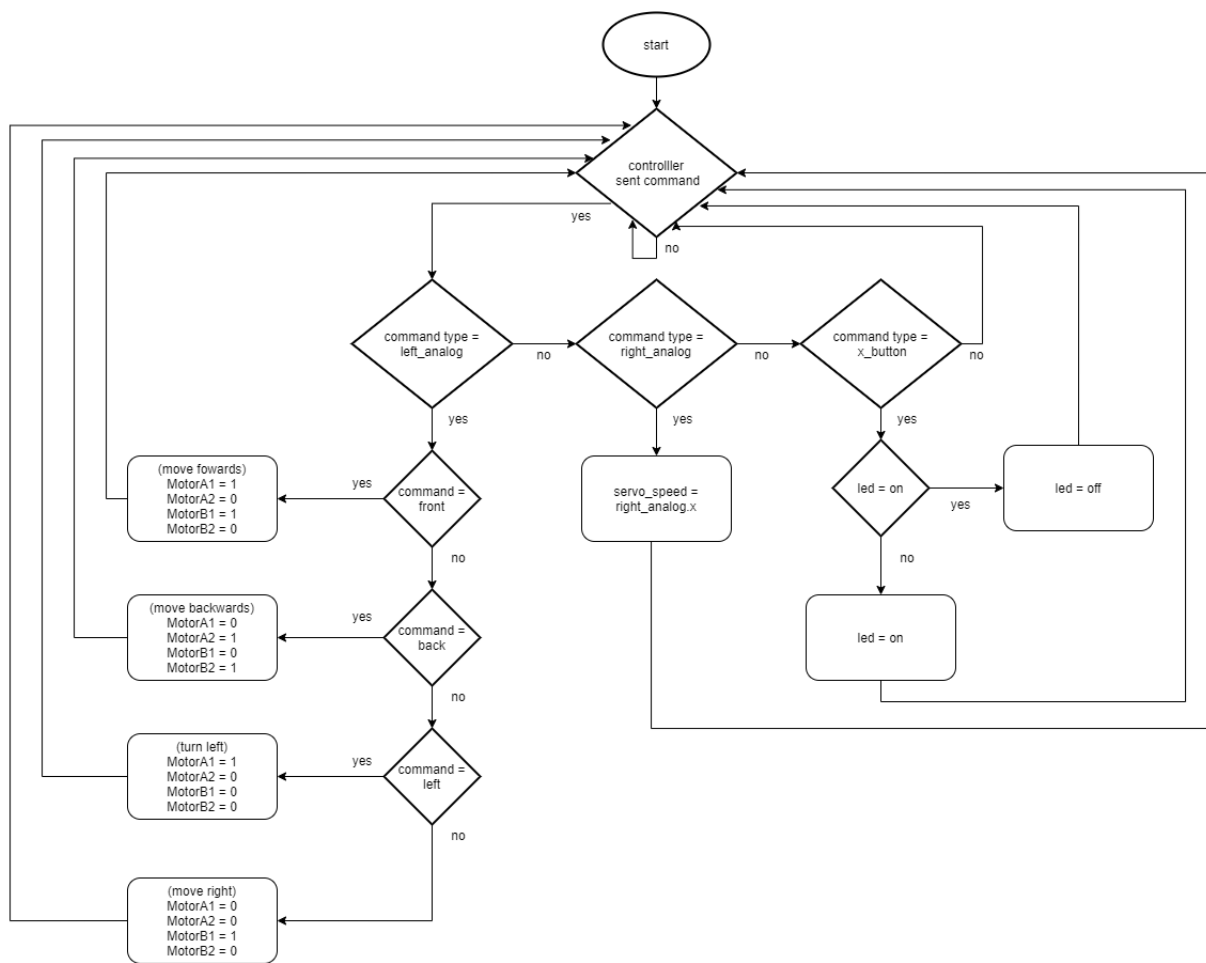


Figura 19: Fluxograma da *thread* de recebimento dos dados do controle.

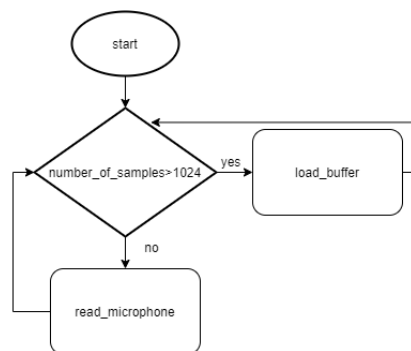


Figura 20: Fluxograma da *thread* de envio dos dados do áudio.

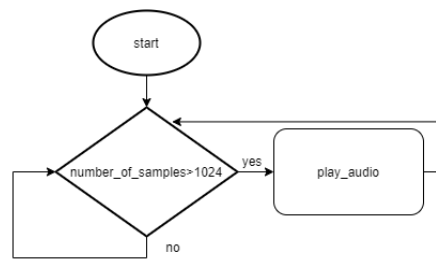


Figura 21: Fluxograma da *thread* de recebimento dos dados do áudio.

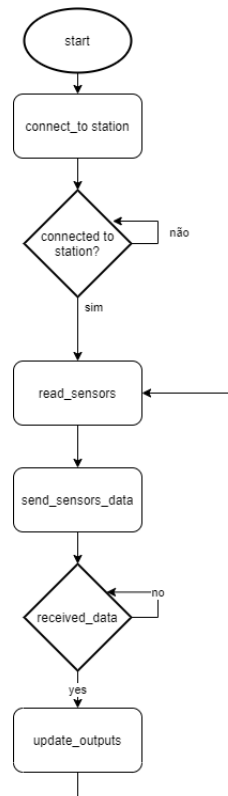


Figura 22: Fluxograma da *thread* principal do *software* do robô.

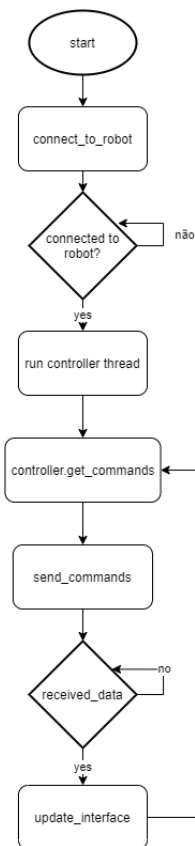


Figura 23: Fluxograma da *thread* principal do *software* da estação base.

Após realizar o projeto do *software* foi dado início à sua implementação utilizando a linguagem de programação Python e a biblioteca para interface gráfica Qt. Para controle e versionamento do código foi utilizado o sistema Git em conjunto com o sistema de hospedagem de repositórios Github, onde foi possível gerenciar todo o código do projeto em grupo.

3.5 INTEGRAÇÃO

Devido a grande quantidade de funcionalidades, a parte de integração recebeu uma atenção especial. Durante todo o projeto, buscou-se realizar validações ao final de cada etapa, sendo ela de hardware, software ou mecânica, tendo essas validações o objetivo de evitar falhas inesperadas na hora das integrações com as demais partes. Com isso, buscou-se consolidar de maneira efetiva cada parte a cada etapa. A integração do projeto final foi realizada por partes. Primeiro, realizou-se a integração da parte mecânica com o hardware, fazendo com que todos os componentes do projeto (Raspberry PI, sensores, atuadores...) fossem acoplados e encaixados de maneira adequada no chassi de esteira. Após alguns testes, constatou-se que alguns componentes poderiam ser encaixados de maneira melhor, otimizando melhor o espaço, e assim, pequenas mudanças foram realizadas.

A segunda integração foi entre parte do hardware e do software. Nesse momento, integrou-se a câmera e seu servo motor, já dando início a comunicação entre o Raspberry e o computador, possibilitando a troca de informações entre eles, como por exemplo, envio da imagem da câmera para o computador e o envio de comandos de rotação do computador para o Raspberry. Essa integração também foi bem sucedida, e não ocorreram problemas maiores devido as validações realizadas entre cada parte anteriormente.

Por fim, a ultima integração foi a total, de todos os componentes do projeto. Como grande parte já havia sido feito, a maior parte dessa integração não apresentou muitos problemas. Porém, no momento de integrar a bateria reserva, constatou-se o problema de que o Raspberry não realizava o *boot* de maneira adequada, pois a bateria utilizada não possui capacidade de fornecimento de corrente suficiente para alimentar todos os componentes do sistema. Diversas alternativas foram buscadas porém nenhuma delas resolveu o problema. Devido a isso, o grupo reconsiderou o requisito funcional RF12 de ter uma bateria reserva, sem afetar de maneira significativa o projeto.

Uma vez que a comunicação e a alimentação são realizadas por cabo, se a alimentação sofresse algum problema (como por exemplo, rompimento do cabo), a comunicação também seria perdida, pois os cabos são unidos de maneira a serem apenas um. Assim, mesmo que o raspberry continuasse alimentando, a comunicação seria perdida, não sendo mais possível controlar o robô. Nesse caso, para recuperar o robô, seria necessário puxar o robô pelo cabo. Devido a esses motivos, conversou-se com os professores orientadores, e o requisito da bateria reservada foi desconsiderado e não implementado.

Com exceção da bateria reserva, a integração das demais partes do projeto ocorreu de maneira satisfatória e como o esperado.

4 EXPERIMENTOS E RESULTADOS

Diversos experimentos foram realizados durante o projeto. Dentre eles, estão os testes de todos os sensores e atuadores de maneira individualizada, e posteriormente, de maneira coletiva e utilizando o software para controle e verificação. Uma vez que a metodologia seguida pela equipe foi a de validar cada pequena etapa do projeto, foram realizados experimentos contínuos para garantir o funcionamento e facilitar integração das partes posteriormente.

Para a construção do mapeamento da trajetória, havia sido planejado utilizar os dados coletados do acelerômetro. Entretanto, após pesquisas e testes, observou-se que seria necessário realizar uma integral dupla nos dados coletados, para que fosse possível obter os dados desejados. Entretanto, ao realizar essa operação, o erro da posição escalava de maneira exponencial, perdendo assim a sua aplicação prática devido ao grande erro em relação a posição real. Devido a esse motivo, utilizou-se os dados provenientes do controle de video game, o qual fornecia os comandos para a movimentação do robô. Realizando a calibragem com a velocidade real do robô, para o mapeamento ser coerente com a movimentação, obteve-se um resultado satisfatório, como mostrado na Figura 27. Cientes das limitações da solução implementada, a equipe acreditou que o resultado estava suficientemente adequado para o projeto.

Ao final do projeto, após todas as integrações terem sido feitas, obteve-se diversos resultados, os quais serão relatados de maneira detalhada a seguir. Da parte do robô, obteve-se o robô físico presente na Figura 24, que tem dimensões de 19,5cm x 10cm x 9,3cm de comprimento, largura e altura, respectivamente. Com essas dimensões, o robô cumpre de maneira satisfatória o requisito de ser pequeno o suficiente para passar por pequenas frestas. Também é possível visualizar sua cobertura, construída para proteger os componentes de hardware. É possível observar o robô internamente sem a cobertura de proteção na Figura 25.

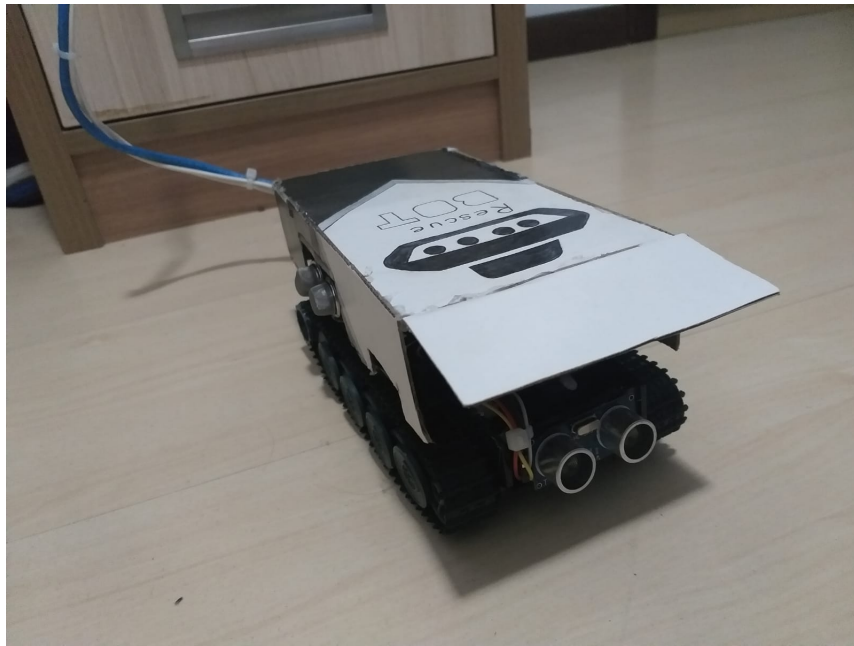


Figura 24: Imagem do robô físico construído

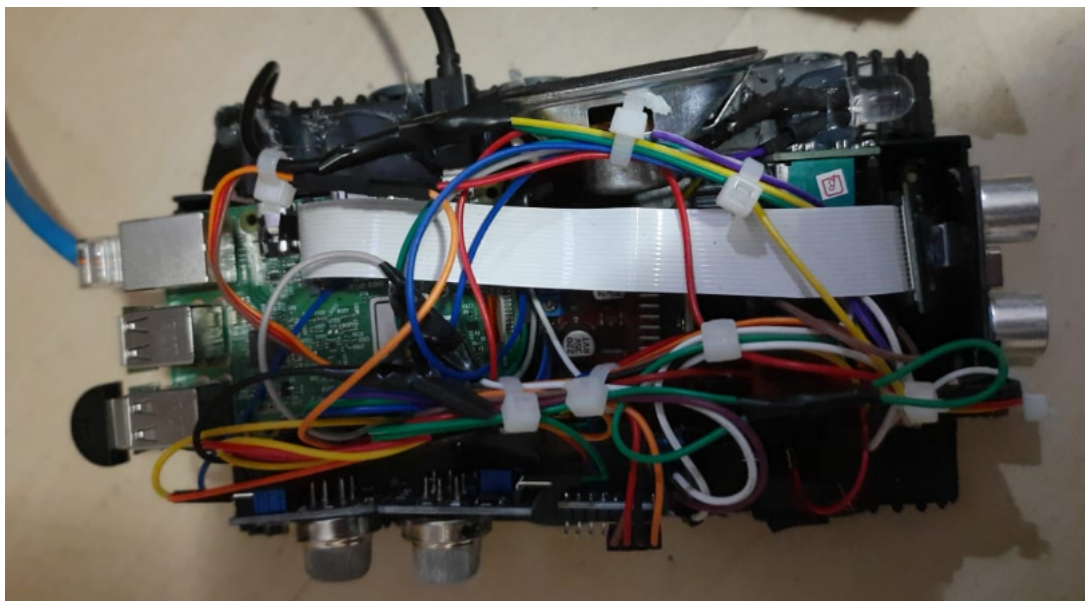


Figura 25: Imagem do robô físico construído

Além disso, o robô construído também foi capaz de superar obstáculos, sem apresentar instabilidade e correr o risco de tombos. Isso ocorre pois as esteiras auxiliam no movimento de ultrapassar obstáculos, e a configuração dos componentes no chassi favorece para que ele seja bastante estável. Um exemplo disso está no uso de pilhas de alimentação no compartimento inferior, que auxiliam a manter o centro de gravidade no centro do robô, fornecendo uma maior estabilidade. Na Figura 26 é possível visualizá-lo superando dois obstáculos ao mesmo tempo.

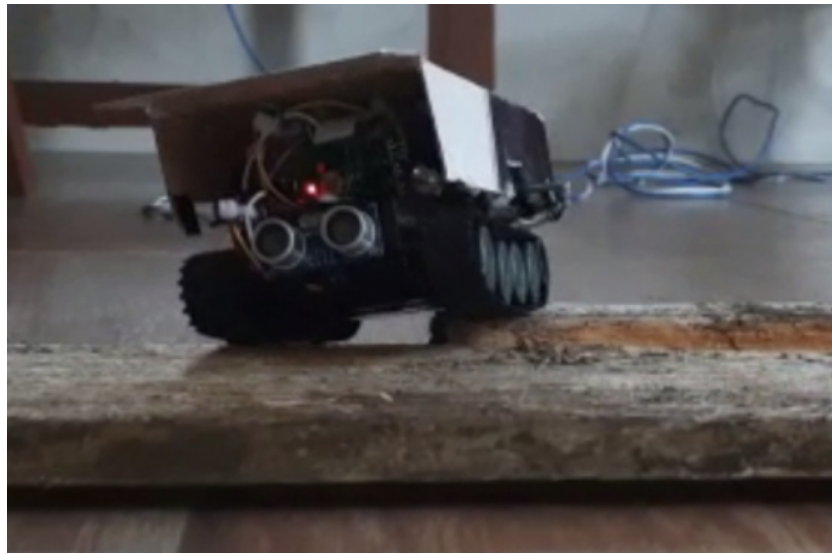


Figura 26: Imagem do robô superando obstáculos

Outro resultado bastante significativo do projeto foi a interface gráfica desenvolvida, presente na Figura 27. Nela, é possível observar, a esquerda, a imagem da câmera, que é transmitida com bastante fluidez. Ao lado, estão presentes as informações do ambiente coletadas pelos sensores. São elas:

- Presença de gás inflamável e fumaça
- Presença de monóxido de carbono
- Temperatura em graus Celsius
- Informações do giroscópio, nos eixos x, y e z
- Informações de aceleração, também nos eixos x, y e z
- Distância frontal livre, sem obstáculos
- Umidade relativa do ar

Na parte inferior direita, está presente o mapeamento da trajetória realizada pelo robô. O triângulo em azul representa o robô na posição atual, as curvas vermelhas representam a trajetória realizada pelo robô e os pontos verdes os obstáculos mapeados por ele.

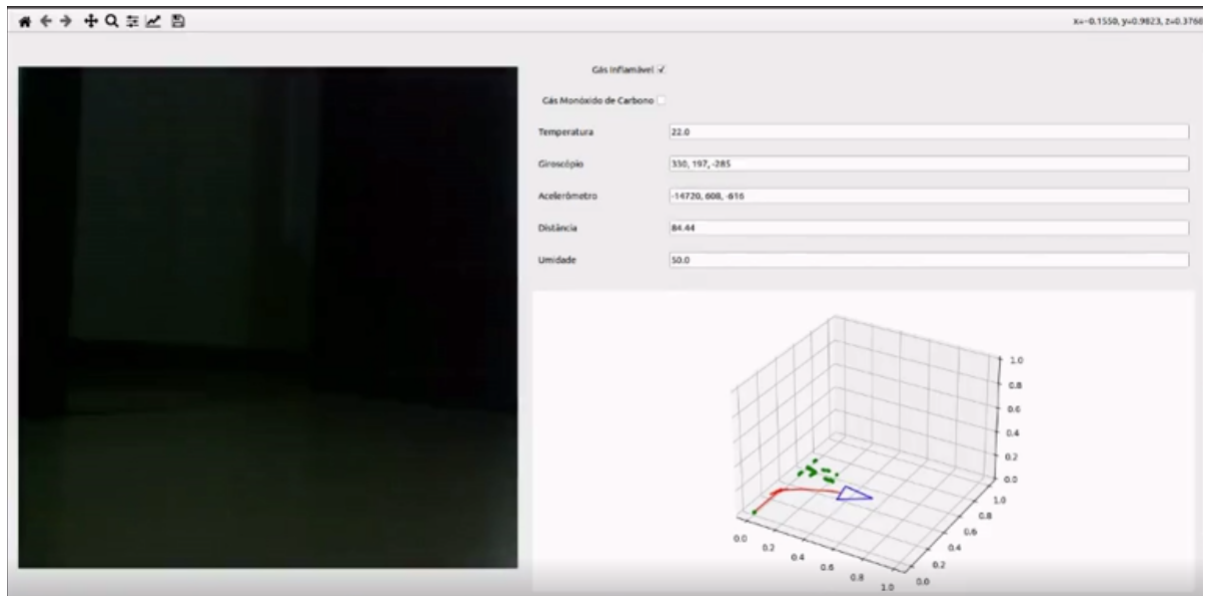


Figura 27: Interface do robô, que é exibida na base

Além dessas funcionalidades, foi possível controlar a movimentação do robô, a posição da câmera e o ativação da luz por meio de um controle de video game, facilitando assim o seu uso. Outro ponto de destaque também foi a comunicação *full-duplex* implementada, onde é possível receber e enviar áudio do robô para base (e vice versa), com uma qualidade relativamente boa.

Outros resultados obtidos também foram o forte acoplamento dos cabos de alimentação e comunicação no chassi do robô, possibilitando que ele fosse puxado e recuperado em casos de perda de comunicação. Além disso, também implementou um sistema de listagem das informações do robô ao longo do seu funcionamento, salvando-as em um arquivo de texto. Essa última não fazia parte dos requisitos, mas a equipe optou por realizá-la após sugestão dos professores.

5 CRONOGRAMA E CUSTOS DO PROJETO

5.1 CRONOGRAMA

Para a construção do cronograma, foram definidas todas as atividades necessárias para a realização do projeto. Com essa definição, foram estipuladas datas visando uma melhor organização do tempo. O cronograma pode ser dividido em 5 partes distintas, sendo cada uma delas voltada mais para cada um dos respectivos entregáveis, como está presente nas Figuras 28, 29, 30, 31, 32, e 33. O Diagrama de Gantt está presente nas imagens 34 e 35.

Descrição	Atividades	Responsáveis		Status da atividade	Situação da conclusão	PREVISÃO		HORAS		
		Principal	Backup/auxílio			INÍCIO	FIM	PREVISTAS	EXTRA	TRABALHADAS
AT01	Apresentação de propostas e discussão	Todos	Todos	Concluído	No prazo	16-jun.	22-jun.	8	2	8
Entregável1: Plano de projeto detalhado										
AT02	Construção do Diagrama de blocos	Todos	Todos	Concluído	No prazo	22-jun.	22-jun.	3	1	3
AT03	Levantamento de requisitos	Todos	Todos	Concluído	No prazo	25-jun.	29-jun.	3	2	3
AT04	Definição da lista de Componentes	Todos	Todos	Concluído	No prazo	25-jun.	25-jun.	3	1	2
AT05	Compra dos componentes	Todos	Todos	Concluído	No prazo	26-jun.	26-jun.	1	1	1
AT06	Análise de Riscos	Todos	Todos	Concluído	No prazo	4-jul.	5-jul.	2	1	2
AT07	Construção do Plano de projeto	Todos	Todos	Concluído	No prazo	4-jul.	6-jul.	20	7	18
AT08	Criação do Blog	Todos	Todos	Concluído	No prazo	29-jun.	4-jul.	2	1	2

Figura 28: Tarefas planejadas para o entregável 1

Descrição	Atividades	Responsáveis			Status da atividade	Situação da conclusão	PREVISÃO		HORAS		
		Principal	Backup/auxílio				INÍCIO	FIM	PREVISTAS	EXTRA	TRABALHADAS
	Entregável2: Site/blog de acompanhamento										
AT09	Inserção das informações do Plano no Blog	Luan	Hadryan	Concluído	No prazo	6-jul.	8-jul.	1	1	3	
AT10	Planejamento da parte mecânica	Hadryan, Luan	Felipe	Concluído	No prazo	7-jul.	14-jul.	8	4	9	
AT11	Configuração do ambiente no Raspberry	Felipe	Luan	Concluído	No prazo	2-jul.	2-jul.	2	1	3	
AT12	Implementação da comunicação entre Controle e Notebook	Hadryan	Felipe	Concluído	Com atraso	2-jul.	2-jul.	2	1	2	
AT13	Teste dos motores	Felipe	Luan	Concluído	Com atraso	7-jun.	7-jul.	2	1	3	
AT14	Implementação de controle dos motores	Felipe, Luan	Hadryan	Concluído	No prazo	8-jul.	10-jul.	5	3	1	
AT15	Teste da câmera	Felipe	Hadryan	Concluído	No prazo	2-jul.	2-jul.	1	1	1	
AT16	Teste Giroscópio e acelerômetro	Felipe	Hadryan	Concluído	Com atraso	6-jul.	7-jul.	1	1	5	
AT17	Teste dos sensores de gás	Felipe	Hadryan	Concluído	Com atraso	6-jul.	6-jul.	1	1	1	
AT18	Pesquisar métodos de comunicação Computador e Raspberry utilizando cabo	Hadryan, Luan	Felipe	Concluído	No prazo	6-jul.	13-jul.	6	3	2	
AT19	Teste sensores de distância	Felipe	Luan	Concluído	Com atraso	7-jul.	7-jul.	1	1	3	
AT20	Acender e apagar a luz do Robô	Felipe	Hadryan	Concluído	No prazo	6-jul.	6-jul.	1	1	1	
AT21	Confeccionamento da placa de alimentação	Felipe, Luan	Hadryan	Concluído	No prazo	2-jul.	3-jul.	3	1	4	
AT22	Validação I - Todas as atividades anteriores*	Todos	Todos	Concluído	Com atraso	16-jul.	17-jul.	15	5	3	
AT23	Inserção da evolução do projeto no blog	Luan	Hadryan	Concluído	No prazo	12-jul.	13-jul.	1	1	1	

Figura 29: Tarefas planejadas para o entregável 2 - Parte 1

Descritor	Atividades	Responsáveis		Status da atividade	Situação da conclusão	PREVISÃO		HORAS		
		Principal	Backup/auxílio			INÍCIO	FIM	PREVISTAS	EXTRA	TRABALHADAS
	Projeto/montagem da estrutura mecânica									
AT24	Teste do microfone	Felipe	Luan	Concluído	Com atraso	9-jul.	12-jul.	4	2	6
AT25	Teste do auto-falante	Felipe	Hadryan	Concluído	No prazo	9-jul.	12-jul.	4	2	6
AT26	Roteamento da pcb do autôfalante	Felipe	Luan	Não necessário		10-jul.	13-jul.	2	1	0
AT27	Roteamento da pcb do microfone	Felipe	Hadryan	Não necessário		11-jul.	14-jul.	2	1	0
AT28	Comunicação entre o computador e raspberry	Hadryan, Luan	Felipe	Concluído	Com atraso	16-jul.	17-jul.	4	2	4
AT29	Emissão de som no Computador	Hadryan, Luan	Felipe	Concluído	No prazo	9-jul.	9-jul.	4	2	2
AT30	Emissão de som no Raspberry	Felipe	Luan	Concluído	No prazo	13-jul.	15-jul.	5	3	1
AT31	Captura de voz no computador	Hadryan	Luan	Concluído	No prazo	13-jul.	15-jul.	4	2	2
AT32	Captura de voz no raspberry	Felipe	Luan	Concluído*	Com atraso	16-jul.	17-jul.	4	2	3
AT33	Envio de voz Computador -> Raspberry	Hadryan	Felipe	Concluído*	No prazo	16-jul.	17-jul.	3	2	3
AT34	Envio de voz Raspberry -> Computador	Luan	Felipe	Concluído*	Com atraso	16-jul.	17-jul.	3	2	2
AT35	Teste da envios de vozes	Hadryan	Felipe	Concluído*	Com atraso	16-jul.	17-jul.	2	1	1
AT36	Envio da imagem do robô para o computador	Felipe	Hadryan	Concluído*	No prazo	16-jul.	17-jul.	6	2	3
AT37	Montagem do cabo	Hadryan, Luan	Felipe	Concluído*	No prazo	12-jul.	17-jul.	2	1	1
AT38	Montagem da estrutura de proteção	Luan	Felipe	Concluído	No prazo	17-jul.	17-jul.	4	2	3
AT39	Validação II - Todas as atividades anteriores	Todos	Todos	Concluído	Com atraso	16-jul.	17-jul.	4	2	7
AT40	Inserção da evolução do projeto no blog	Luan	Hadryan	Concluído	No prazo	18-jul.	18-jul.	1	1	1

Figura 30: Tarefas planejadas para o entregável 2 - Parte 2

		Responsáveis				PREVISÃO		HORAS		
Descritor	Atividades	Principal	Backup/auxílio	Status da atividade	Situação da conclusão	INÍCIO	FIM	PREVISTAS	EXTRA	TRABALHADAS
	Entregável#3: Projeto/testes do hardware									
AT41	Modelagem da interface gráfica	Hadryan, Luan	Felipe	Concluído	No prazo	17-jul.	22-jul.	4	2	3
AT42	Construção do UML	Luan	Hadryan	Concluído	No prazo	12-jul.	23-jul.	6	3	5
AT43	Construção do diagrama de caso de uso	Luan	Felipe	Concluído	No prazo	12-jul.	23-jul.	4	2	3
AT44	Construção do diagrama de sequência	Felipe, Luan	Hadryan	Concluído *	Com atraso	16-jul.	30-jul.	4	2	2,5
AT45	Construção do diagrama de estados e transições	Felipe, Luan	Hadryan	Concluído *	Com atraso	17-jul.	30-jul.	3	1	1
AT46	Processamento dos dados de movimentação do robô	Hadryan, Luan	Felipe	Não necessário		15-jul.	22-jul.	20	7	1
AT47	Construção do mapeamento da trajetória	Hadryan	Luan	Concluído *	No prazo	18-jul.	30-jul.	5	3	4
AT48	Implementação da interface gráfica	Hadryan, Luan	Felipe	Concluído	No prazo	19-jul.	24-jul.	6	3	3
AT49	Construção da representação visual da trajetória	Hadryan, Luan	Felipe	Concluído	No prazo	19-jul.	24-jul.	4	2	2
AT50	Construção da placa para alimentação reserva	Felipe	Luan	Não necessário		19-jul.	21-jul.	8	4	0
AT51	Inserção da bateria reserva e fixação	Felipe	Luan	Não concluído	Não concluído	21-jul.	21-jul.	2	1	8
AT52	Validação III - Todas as atividades anteriores	Todos	Todos	Concluído	Com atraso	22-jul.	22-jul.	8	4	6
AT53	Documentação do software	Luan	Hadryan	Concluído	No prazo	18-jul.	30-jul.	6	3	2
AT54	Inserção da evolução do projeto no blog	Felipe	Luan	Concluído	No prazo	25-jul.	25-jul.	2	1	2

Figura 31: Tarefas planejadas para o entregável 3

Descritor	Atividades	Responsáveis			Situação da conclusão	PREVISÃO		HORAS		
		Principal	Backup/auxílio	Status da atividade		INÍCIO	FIM	PREVISTAS	EXTRA	TRABALHADAS
	Entregável#4: Projeto do software									
AT55	Integração de hardware e software	Todos	Todos	Concluído *	No prazo	21-jul.	31-jul.	40	20	45
AT56	Validação IV: Bateria de testes	Luan	Felipe	Concluído *	No prazo	24-jul.	31-jul.	5	3	5
AT57	Ajustes necessários da validação IV	Hadryan	Felipe	Concluído *	No prazo	24-jul.	2-ago.	10	3	12
AT58	Validação V: Bateria de testes	Felipe	Luan	Concluído *	No prazo	25-jul.	2-ago.	5	3	2
AT59	Ajustes necessários da validação V	Hadryan	Luan	Concluído *	No prazo	26-jul.	2-ago.	10	5	9
AT60	Inserção da evolução do projeto no blog	Felipe	Hadryan	Concluído *	No prazo	28-jul.	2-ago.	1	1	0,5

Figura 32: Tarefas planejadas para o entregável 4

Descritor	Atividades	Responsáveis		Status da atividade	Situação da conclusão	PREVISÃO		HORAS		
		Principal	Backup/auxílio			INÍCIO	FIM	PREVISTAS	EXTRA	TRABALHADAS
Entregável#5: Integração mecânica/hardware/software										
AT61	Realização do relatório técnico final	Todos	Todos	Concluído *	Com atraso	29-jul.	7-ago.	30	10	25
AT62	Planejamento e gravação do vídeo de apresentação	Todos	Todos	Concluído	No prazo	5-ago.	7-ago.	10	3	8
AT63	Edição do vídeo final	Hadryan, Felipe	Luan	Concluído	No prazo	8-ago.	11-ago.	10	3	10

Figura 33: Tarefas planejadas para o entregável 5

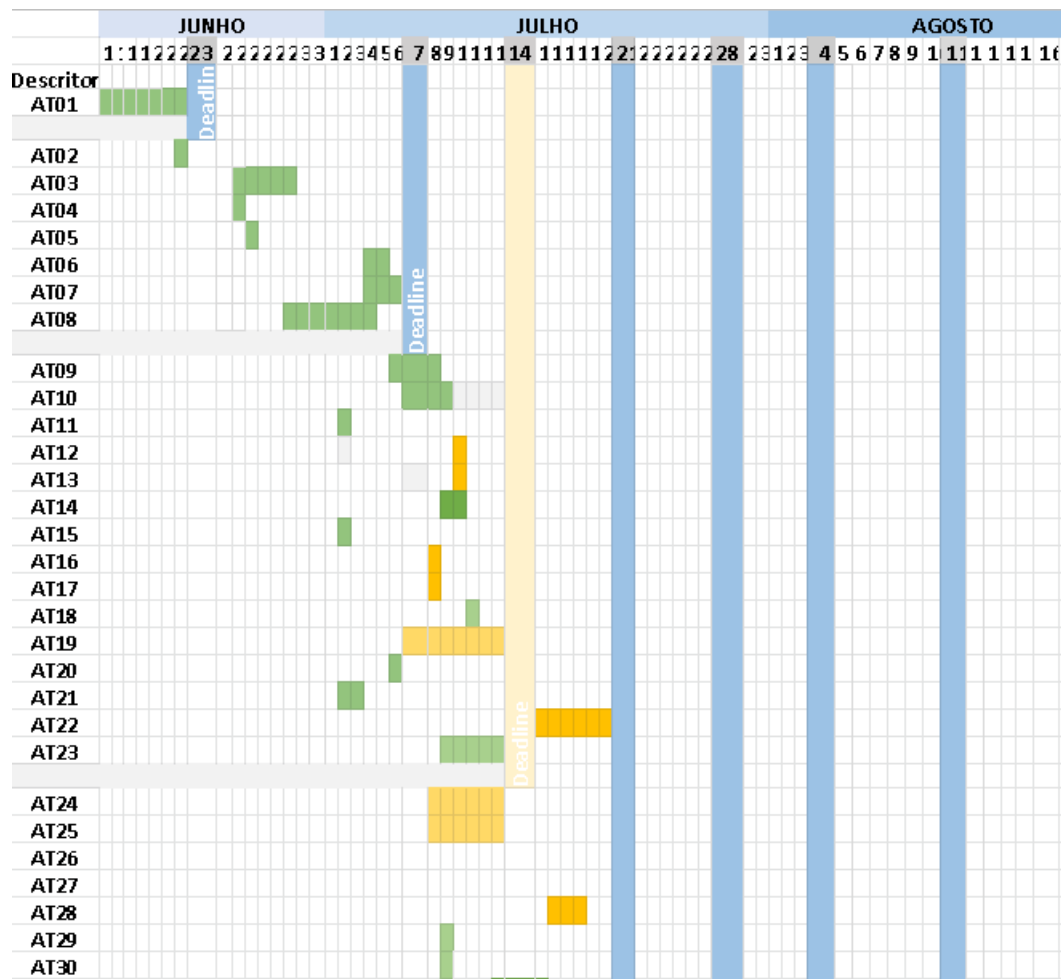


Figura 34: Diagrama de Gantt de todas as atividades - Parte 1

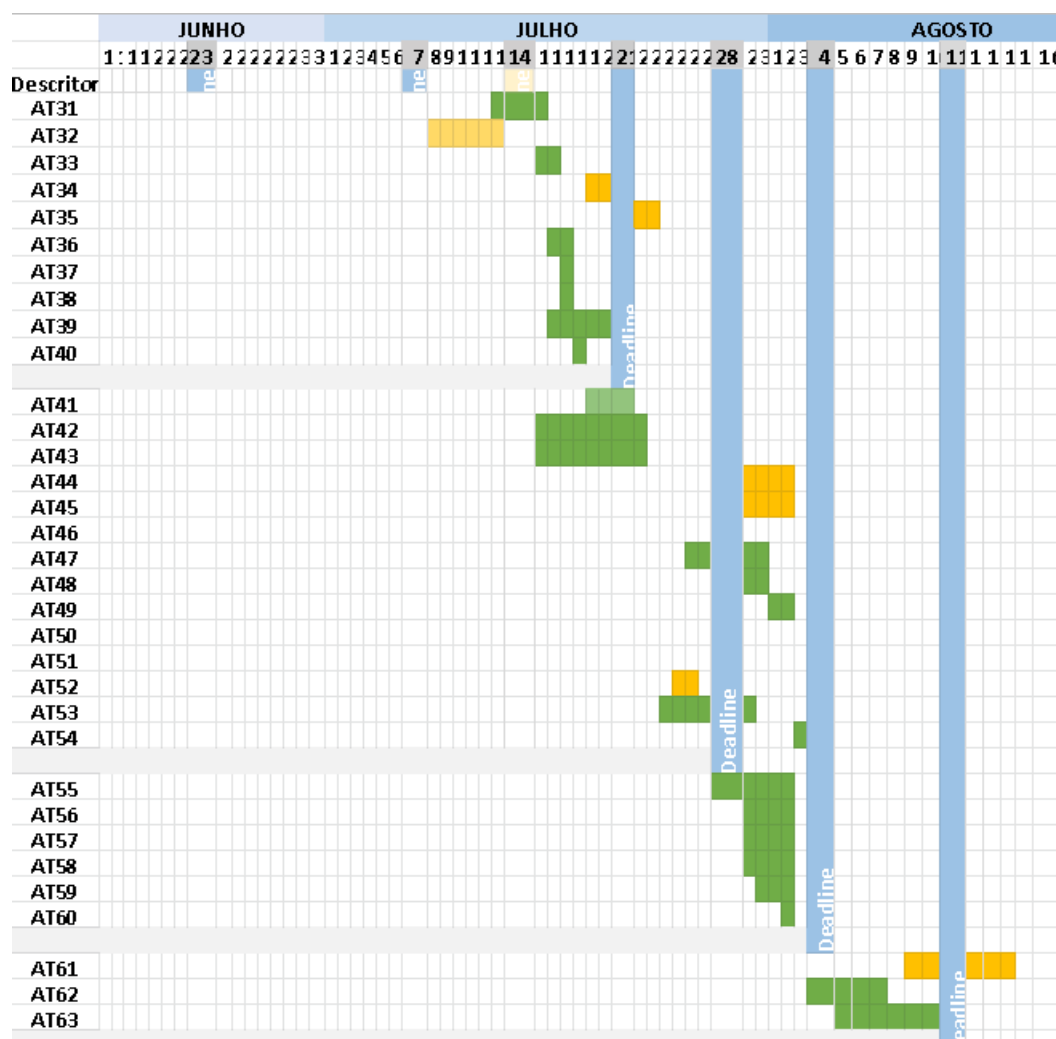


Figura 35: Diagrama de Gantt de todas as atividades - Parte 2

Por fim, no gráfico da Figura 36, tem-se a divisão de horas trabalhadas (em azul), previstas (laranja) e previstas com tempo extra (cinza) de maneira semanal para cada integrante do grupo (em média). A semana 1, que foi a semana de início do, se deu entre 13 de Junho (13/06) e 19 de Junho (19/06), e a semana da finalização do projeto foi a semana 9, que foi entre 8 de Agosto (08/08) e 14 de agosto (14/08). Ao total, foram estimadas 353 horas de trabalho para a conclusão do projeto. Entretanto, foram gastas apenas 285 horas. Isso ocorrer principalmente porque algumas tarefas que haviam sido previstas inicialmente não precisaram ser realizadas. Todo o cronograma está disponível online¹.

¹Disponível em: <https://docs.google.com/spreadsheets/d/1SBVyrjZVA9aAN5G0pka34jIzXBNRVLy/edit?usp=sharing&ouid=113599724903595486487&rtopof=true&sd=true>

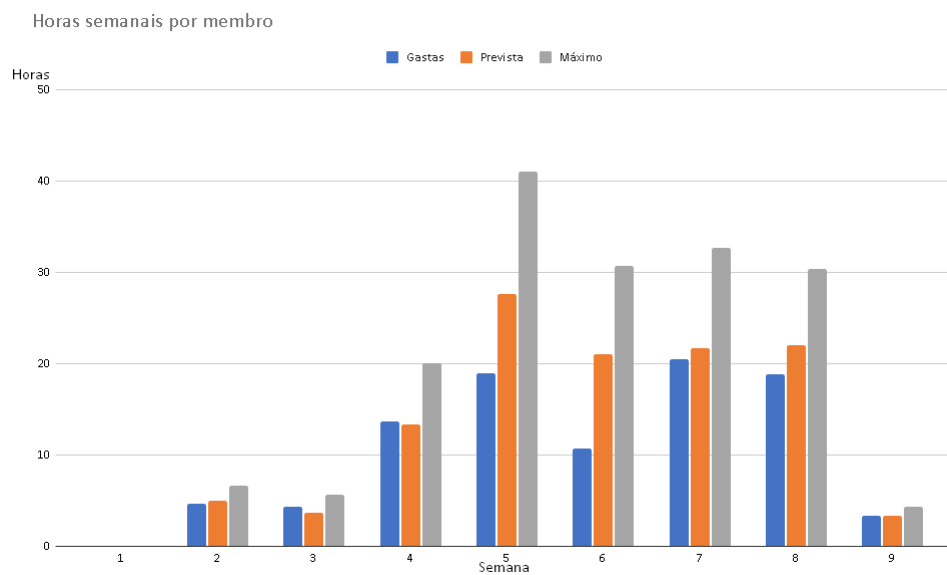


Figura 36: Gráfico de horas trabalhadas e estimadas semanalmente por integrante

5.2 CUSTOS

Uma vez definidos todos os componentes necessários, foi possível buscar informações precisas de qual seria o valor total do projeto. A lista de componentes e seus respectivos valores está presente na Tabela 8. O projeto apresentou um custo total final de R\$1003,54. Entretanto, vale destacar que como os integrantes do grupo já tinham diversos desses componentes e, por esse motivo, os custos foram significativamente inferiores aos apresentados na tabela.

Tabela 8: Tabela de componentes e preços

Material	Quantidade	Preço
Raspberry pi 3 Model B	1	R\$299,90
Módulo Câmera	1	R\$ 49,90
Giroscópio e acelerômetro	1	R\$22,99
Sensor ultrassônico HC-SR04	1	R\$13,00
Microfone	1	R\$24,00
Speaker	1	R\$10,00
Led alta brilho	1	R\$5,00
Ponte H - L298n	1	R\$21,90
Chassi tipo tanque	1	R\$129,90
Sensor de Gás inflamável - MQ-2	1	R\$28,40
Sensor monóxido de carbono - MQ-9	1	R\$33,95
Sensor de temperatura DHT11	1	R\$14,70
Servo rotação da câmera	1	R\$15,00
Cabo ethernet 10m	1	R\$18,90
Cabos alimentação 1 M	10	R\$ 10,00
Controle PS4	1	R\$306,00
Total		R\$1003,54

6 CONCLUSÕES

6.1 CONCLUSÕES

Diante dos resultados citados anteriormente, é possível concluir que o desenvolvimento do projeto obteve sucesso. Os requisitos funcionais e não funcionais definidos inicialmente foram cumpridos, com exceção da bateria reserva, por motivos explicados anteriormente. A qualidade do projeto final foi satisfatória, uma vez que os componentes utilizados não apresentam a qualidade e confiabilidade necessária para uma aplicação real, porém, para fins de um protótipo, são suficientes.

Outro ponto de destaque é o aprendizado obtido com o projeto. Além de aplicar na prática os conhecimentos adquiridos em sala de aula, foi possível exercitar a atividade de planejamento de projetos, da criação de um cronograma de maneira detalhada, com um número considerável de atividades a serem realizadas, e no objetivo de tomar esse cronograma como caminho a seguir, porém podendo adaptá-lo em caso de necessidade. Além disso, também exercitou-se a resolução de problemas de escopo aberto, no qual não há uma única solução, mas sim uma diversidade delas, e cabe ao grupo encontra-las e escolher o caminho a trilhar afim de resolver os problemas. Esse tipo de problema se aproxima daqueles presentes no mercado de trabalho, sendo esse um bom aprendizado e ganho de experiência.

6.2 TRABALHOS FUTUROS

Como trabalho futuro, diversas melhorias poderiam ser aplicadas sobre o projeto atual. Dentre elas estariam em melhorar a qualidade da cobertura de proteção, construindo-a com um material mais resistente, tal como acrílico. Além disso, também realizar o mapeamento das informações coletadas junto com a trajetória, tal como gases e temperatura. Poderia ser efetuada uma melhoria da precisão do mapeamento da trajetória do robô, utilizando componentes e métodos mais precisos. Foi observado que a utilização de um microfone com sensibilidade mais elevada auxiliaria no funcionamento do robô, sendo essa mais uma possível melhoria futura.

Outra evolução do projeto seria possibilitar que o robô explore de maneira autônoma o ambiente, sem a necessidade de um operador humano diretamente envolvido. Assim, seria possível utilizar mais robôs realizando o mapeamento do local, e, através do compartilhamento de informações entre eles, caracterizando assim um sistema multiagente cooperativo. Com isso, é possível obter maior precisão, área de abrangência e velocidade no mapeamento. Com um mapeamento mais detalhado e preciso, também seria possível construir um grafo com os caminhos existentes, no qual os caminhos seriam as arestas, e os cruzamentos de dois ou mais caminhos seriam os vértices. Com isso, seria possível aplicar algoritmos de descobrimento de caminhos ótimos, tais como o A^* e o $LRTA^*$, a fim de permitir a equipe de socorro encontrar a vítima soterrada de maneira mais rápida e fácil.

REFERÊNCIAS

- CODELLOS, F. **Amplificadores, suas funções e tipos**. 2013. Disponível em: <<https://revistaautomotivo.com.br/amplificadores-suas-funcoes-e-tipos/>>. Acesso em: 15 de agosto de 2021.
- ETRONICWINGS. **Raspberry Pi GPIO Access**. 2019. Disponível em: <<https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access>>. Acesso em: 15 de agosto de 2021.
- FILIFELOP. **Raspberry Pi 3 Model B+**. 2018. Disponível em: <<https://www.filipeflop.com/produto/raspberry-pi-3-model-b/>>. Acesso em: 15 de agosto de 2021.
- MAZIERO, C. A. **Sistemas operacionais: conceitos e mecanismos**. [S.l.]: DINF-UFPR, 2019.
- MENDONCA, H. S. **UART**. 2015. Disponível em: <<https://paginas.fe.up.pt/hsm/docencia/comp/uart/>>. Acesso em: 15 de agosto de 2021.
- PANTUZA, G. **O que são e como funcionam os Sockets**. 2017. Disponível em: <<https://blog.pantuza.com/artigos/o-que-sao-e-como-funcionam-os-sockets>>. Acesso em: 15 de agosto de 2021.
- SANTOS, F. **Controle a velocidade de um motor DC com o Arduino utilizando transistor**. 2020. Disponível em: <<https://www.filipeflop.com/blog/controle-a-velocidade-de-um-motor-dc-com-o-arduino-utilizando-transistor/>>. Acesso em: 15 de agosto de 2021.
- THOMSEN, A. **Motor DC com Driver Ponte H L298N**. 2013. Disponível em: <<https://www.filipeflop.com/blog/motor-dc-arduino-ponte-h-l298n/>>. Acesso em: 15 de agosto de 2021.
- WIKIPEDIA. **Servomotor**. 2020. Disponível em: <<https://pt.wikipedia.org/wiki/Servomotor>>. Acesso em: 15 de agosto de 2021.
- WIKIPEDIA. **DC Motor**. 2021. Disponível em: <https://en.wikipedia.org/wiki/DC_motor>. Acesso em: 15 de agosto de 2021.