

## Relatório Técnico

# Trayker - Um sistema automatizado de coleta de bandejas para lanchonetes e restaurantes

Andreas Anael Pereira Gomes andreas.p.gomes@gmail.com

Douglas Soares Vicente douglasvicente@alunos.utfpr.edu.br

Everson de Souza Silva everson@alunos.utfpr.edu.br

Giovane Negrini Marques Costa gcosta@alunos.utfpr.edu.br

Henrique Alves Batochi batochi@alunos.utfpr.edu.br

Abril de 2019

### Resumo

O presente relatório apresenta o desenvolvimento do Trayker, um sistema automatizado de retirada de bandejas em restaurantes. Este projeto surgiu com o intuito de facilitar o recolhimento das bandejas que muitas vezes são deixadas em cima da mesa, impossibilitando o uso por outros clientes do estabelecimento. Para isso, foi desenvolvido: um Robô seguidor de linha capaz de realizar a ação de recolhimento; uma Mesa com sensores capazes de determinar presença de usuários e de bandejas; assim como uma Base para se comunicar com os agentes e coordenar o processo.

## 1 Introdução

O projeto Trayker consiste no desenvolvimento e implementação de um sistema de retirada de bandejas automatizado para lanchonetes e restaurantes. O sistema é capaz de gerenciar a presença de bandejas deixadas sobre as mesas e atuar no recolhimento delas de forma a possibilitar a reutilização do espaço sem causar incômodo aos novos usuários.

O projeto é composto por três partes principais: um **robô**, uma **base** e as **mesas**. As mesas conectam-se à base fornecendo leituras de seus sensores acoplados, sendo eles: ultrassônico, balança, leitor RFID e uma interface de interação através de um botão. Em cima das mesas são posicionadas bandejas adaptadas com *tags* RFID.

A base fornece a interface de comunicação entre as leituras dos sensores recebida das mesas e monta uma fila de atendimento que é então encaminhada para execução pelo robô, além de apresentar uma interface gráfica ao usuário que monitorar o sistema com essas informações. A comunicação entre as mesas é estabelecidas através de um protocolo de comunicação pela rede *Wi-Fi*

e com o robô é utilizada uma comunicação *Bluetooth*. Por medida de precaução, o peso da bandeja deve estar dentro dos limites estabelecidos para o correto funcionamento do sistema. Além disso, há um botão que pode ser usado para impedir o robô de coletar uma bandeja, seja porque a pessoa deseja se ausentar temporariamente (ex: pegar um guardanapo) ou porque houve um erro no sistema.

Uma vez recebida a fila de atendimento pela base, o robô passa a atuar, sendo o responsável por seguir um caminho guiado até a origem da solicitação, percorrendo o trajeto guiado pelas leituras oriundas do sensor de infravermelho responsável por mantê-lo na demarcação do trajeto. Assim, quando o cliente se levanta mas deixa a bandeja, a base envia a instrução para o robô coletar a bandeja daquele lugar e levá-la à cozinha, para ser removida de sua estrutura pelo funcionário monitor do sistema.

Uma visão geral do projeto é apresentada na Figura 1.

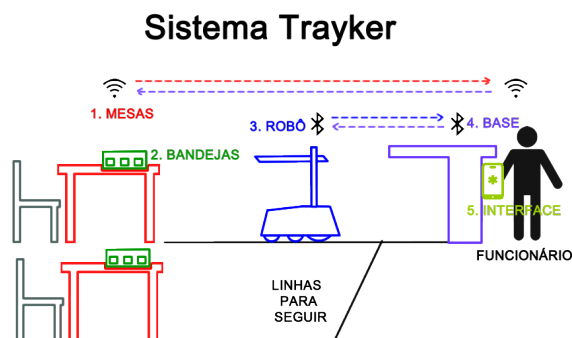


Figura 1: Visão geral do sistema Trayker.

## 1.1 Requisitos

Para delimitar o escopo deste projeto, foram estabelecidos requisitos funcionais. A tabela 1 apresenta os principais requisitos levantados para o projeto.

Tabela 1: Requisitos funcionais do projeto

|   |
|---|
| <b>Mesa</b>   |
| RF_M01 - A mesa deve detectar a presença de pessoas.  |
| RF_M02 - A mesa deve detectar a presença de bandeja e seu peso.   |
| RF_M03 - A mesa deve enviar suas informações para a base.   |
| RF_M04 - A mesa deve permitir ao usuário o cancelamento da retirada da bandeja.   |
| <b>Base</b>   |
| RF_B01 - A base deve receber as informações da mesas.   |
| RF_B02 - A base deve identificar onde há bandejas a serem retiradas.  |
| RF_B03 - A base deve gerenciar uma fila de coleta.  |
| RF_B04 - A base deve ser capaz de enviar ao robô a localização da bandeja a ser retirada, assim como a ordem ou cancelamento da ação. |
| <b>Robô</b>   |
| RF_R01 - O robô deve receber ordens da base.  |
| RF_R02 - O robô deve ser capaz de se locomover até as mesas, retirar e transportar uma bandeja.                                       |
| RF_R03 - O robô deve levar as bandejas coletadas até a área denominada cozinha.   |
| RF_R04 - A capacidade de carga do robô é de bandejas com peso total de no máximo de 300 gramas.                                       |
| RF_R05 - O robô deve permanecer na cozinha até ter a bandeja retirada por um funcionário e receber uma nova ordem.                    |

## 2 Componentes e Tecnologias

Esta seção detalha as diversas tecnologia utilizadas para garantir o funcionamento do sistema.

### 2.1 Mesa

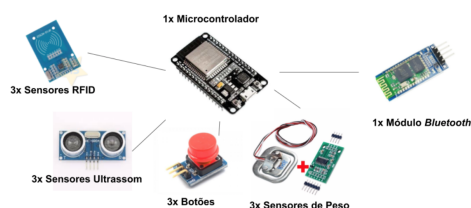


Figura 2: Conjunto eletrônico da mesa

#### 2.1.1 Microcontrolador ESP32

O ESP32[1] é um microcontrolador de alta performance para aplicações embarcadas, contando com um baixíssimo consumo de energia e módulos de WiFi e Bluetooth integrados. Pode ser programado tanto por Python através do MicroPython como pela Arduino IDE. Além disso, possui um baixo custo. Foi empregado na mesa para interfacear e processar os dados obtidos dos sensores embarcados, responsável também pela comunicação com a base para solicitações

de retiradas e interrupções de ordens oriundas de leituras errôneas.

### 2.1.2 Sensor Ultrassônico

O princípio de funcionamento dos sensores ultrassônicos consiste na emissão de sinais ultrassônicos e na leitura do retorno (reflexo/eco) desse mesmo sinal emitido. A distância entre o sensor e o objeto que refletiu o sinal é calculada com base no tempo entre o envio e leitura de retorno. O sensor utilizado é o HC-SR04[2], que é capaz de medir distâncias de 2cm a 4m com boa precisão. Este módulo possui um circuito pronto com emissor e receptor acoplados.

### 2.1.3 Célula de Carga

A resistência da célula de carga[3] varia de acordo com a força exercida em sua superfície, sendo utilizada como sensor de peso. Ela é conectada a um decodificador HX711[4], que alimenta a célula com um nível de tensão, lê a tensão resultante na saída e então converte o sinal analógico recebido para um número digital, representado em 24 *bits*.

### 2.1.4 Sensor RFID MFRC522

O módulo Mfrc522 Mifare[5] é um leitor capaz de realizar a leitura e escrita de *chips* RFID. Um *chip* possui um código único, identificado pelo leitor RFID quando estiver dentro do alcance de leitura, normalmente em torno de 10cm. Neste projeto, este sensor é utilizado tanto para detectar a presença da bandeja na mesa, quanto pelo robô para identificar os cruzamentos e pontos de coleta (nas mesas) e retorno das bandejas (na cozinha).

### 2.1.5 MicroPython

A linguagem de desenvolvimento do software controlador do hardware embarcado na mesa foi o *MicroPython*[6], uma implementação da linguagem de programação *Python 3* otimizada para operar em microcontroladores.

## 2.2 Robô

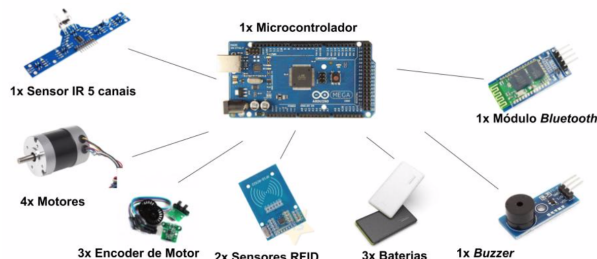


Figura 3: Conjunto eletrônico do robô

### 2.2.1 Arduino

A Placa Arduino Mega 2560[7] é uma placa de microcontrolador baseada no ATmega2560. Ele possui um grande número de portas GPIO, é alimentado a 5V e trabalha com um clock de 16 MHz.

### 2.2.2 Sensor Infravermelho

Sensores infravermelhos funcionam com um emissor e um receptor de sinais infravermelhos. Enquanto o primeiro emite um sinal base, o segundo retorna uma leitura da intensidade do sinal. Essa intensidade varia com a distância e a superfície que refletiu o sinal. Neste projeto, é utilizado o sensor BFD-1000[8] que acopla cinco sensores em uma barra de leitura.

### 2.2.3 Carregador portátil

Carregadores portáteis são utilizados para alimentação do sistema do robô. Possuindo capacidade de 10.000 mAh, possuindo duas saídas USB de 1A e 2A.

### 2.2.4 Buzzer

Utilizado para emissão sonora do robô.

## 2.3 Base

### 2.3.1 Raspberry Pi

Este dispositivo é capaz de realizar a maioria das funções esperadas de um computador *desktop* comum, operando um sistema operacional baseado em Debian[9]. Uma fotografia é apresentada na Figura 4.

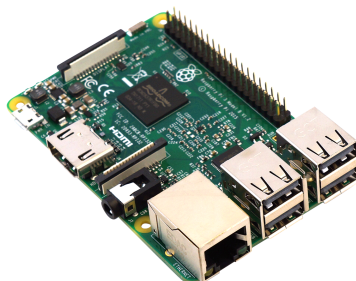


Figura 4: Único dispositivo da base: a Raspberry Pi.

### 2.3.2 MQTT

MQTT [10] significa “*Message Queuing Telemetry Transport*” [10] e é um protocolo projetado para troca confiável de mensagens entre pequenos dispositivos em baixa largura de banda. Seu funcionamento consiste de mensagens enviadas sob cabeçalhos para um servidor por um "publicador". Qualquer dispositivo que for "inscrito" em um dado cabeçalho recebe as mensagens pertinentes do servidor. Opera normalmente, mas não obrigatoriamente, sobre o TCP-IP, podendo utilizar outros protocolos na camada inferior.

### 2.3.3 Protocolo de Comunicação Bluetooth

*Bluetooth* [11] é um padrão de comunicação sem fio de curto alcance, baixo custo e baixo consumo de energia que utiliza tecnologia de rádio. Sua especificação é aberta e está publicamente disponível. Dois módulos Bluetooth são utilizados: o módulo interno da Raspberry Pi (BCM43438) e um módulo Bluetooth HT-06 na Arduino Mega, sendo este último o com o menor alcance: cerca de 10m.

## 3 Desenvolvimento

Nesta seção são abordados os processos de desenvolvimento do *Trayker*, sendo eles: o desenvolvimento do *hardware*, do *software* e da estrutura física.

### 3.1 Hardware - Mesa

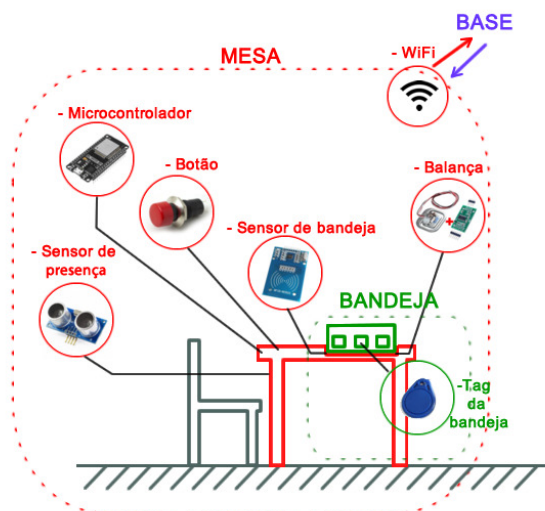


Figura 5: Esquemático da construção da mesa.

Como plataforma do sistema embarcado das mesas, foi escolhido a ESP32 (subseção 2.1.1), por possuir quantidade suficiente de pinos GPIO, conectividade *Wi-fi*, um baixo consumo energético e dimensão física adequada para sua empregabilidade no projeto. Dois sensores ultrassônicos HC-SR04 (subseção 2.1.2), realizam a detecção de usuários na mesa, que possui dois lugares. Duas células de cargas (seção 2.1.3), conectadas cada uma em um decodificador HX711 (subseção 2.1.3), realizam a medição do peso das bandejas. Dois botões servem para interfacear as interrupções das leituras dos sensores e cancelamento de solicitações de retiradas. Para interfacear todos os sensores, foi confeccionada uma placa de circuito impressa, presente na figura 6.

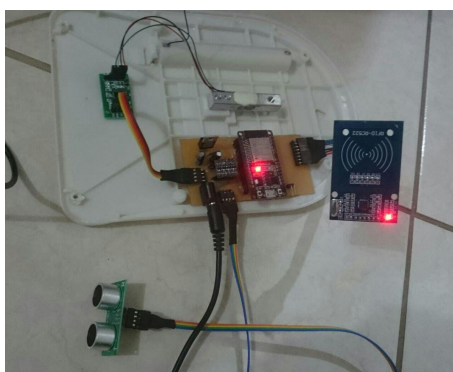


Figura 6: Placa de circuito confeccionada para o projeto

Os sensores ultrassônicos HC-SR04 (subseção 2.1.2) foram encaixados no beiral da mesa, visando a posição do usuário sentado a sua frente. As células de cargas (subseção 2.1.3) foram posicionadas na viga da mesa, sendo esta perfurada para permitir a passagem do prato de medição apenas, ficando a estrutura física das células, ESP32 (subseção 2.1.1) e fiação na parte inferior da viga. Os botões foram posicionados na parte superior direita de cada espaço de alimentação na mesa. Nas figuras 6 e 7 podem ser vistas

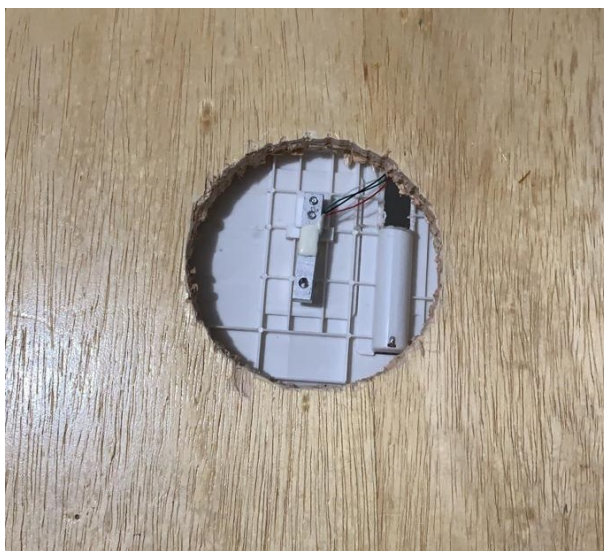


Figura 7: Estrutura embarcada na mesa.



Figura 8: Célula de carga exposta na parte superior da viga.

A conexão dos sensores ultrassônicos (subseção 2.1.2), célula de carga (subseção 2.1.3) e botões foi feita através de uma placa de interfaceamento, visando maior facilidade na conexão.

A comunicação com a estação base e o sistema embarcado na mesa se dá através de um protocolo MQTT (subseção 2.3.2) utilizando a rede *Wi-Fi* fornecida pela base.



### 3.2 Hardware - Robô

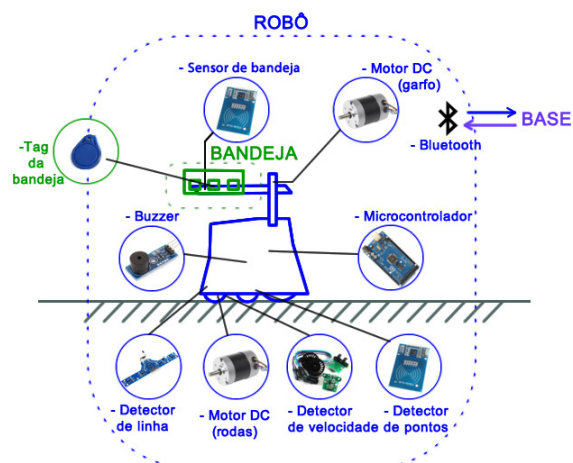


Figura 9: Esquemático da construção do robô.

O hardware do robô é centralizado no *Arduino* que é alimentado com um *power bank*. Os sensores de fim de curso, o *buzzer* e o infravermelho são conectados diretamente nele. Já para os sensores de RFID (sub-subseção ??sub-subseção 2.1.4), o *buzzer* e o módulo *Bluetooth* (sub-subseção 2.3.3), foi fabricada uma placa PCB para realizar a integração e gerar a tensão de 3.3 V utilizada por esses sensores. Essa tensão é gerada por um CI LM1117-3.3 na placa, que é conectado à mesma bateria utilizada para o *Arduino* (sub-subseção 2.2.1).

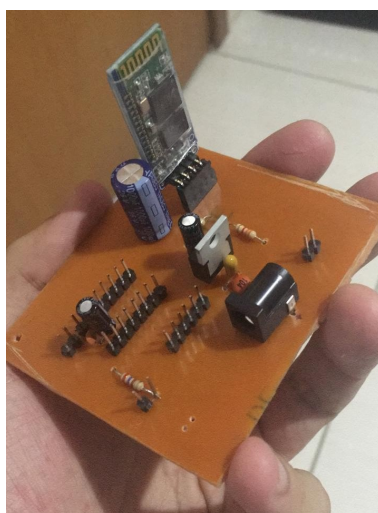


Figura 10: Placa PCB fabricada para o robô.

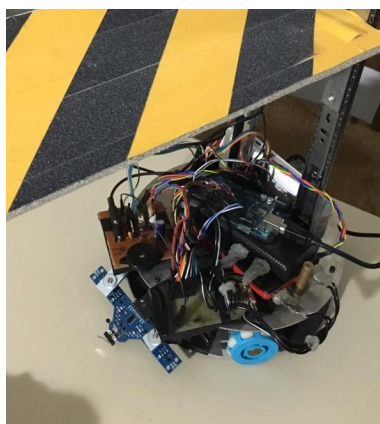


Figura 11: Visão geral da construção do robô.

Esta placa também incorpora diferentes capacitores para estabilizar a alimentação dos componentes, uma vez que o módulo *Bluetooth* (subseção 2.3.3) apresenta um consumo alto e irregular de corrente até que esteja pareado com algum dispositivo. Já para a alimentação dos motores, foi utilizada a mesma placa confeccionada pela equipe do semestre anterior (do projeto Robô Robçom[12]), que integra dois *power banks* para fornecer a corrente e a tensão necessária. O controle dos motores é feito por dois L298N, que agem como *driver* de corrente e ponte H – um para cada par de motores.

### 3.3 Software - Mesa

O sistema embarcado da mesa é responsável pela aquisição dos dados oriundos da leitura dos sensores, seu processamento e estabelece a comunicação com a estação base.

A aquisição dos dados dos sensores é realizada através de bibliotecas individuais desenvolvidas em *MicroPython* (subseção 2.1.5) para o conjunto de sensores presente na mesa, sendo eles: o sensor ultrassônico (subseção ??), a célula de carga e o sensor RFID. Esse conjunto de leituras fornece parâmetros para definição do estado da mesa, tendo como possibilidade: pronta e não pronta.

O *software* analisa os dados e quando eles se enquadram no cenário em que o peso aferido é de no máximo 300 gramas que é a capacidade de carga do robô especificada nos requisitos, podendo ter uma variação de 20% para mais (totalizando 315 gramas) e a leitura deste peso está atrelado a uma RFID que garanta a presença de uma bandeja e por fim não acuse a presença de usuários, aferido pelo sensor ultrassônico (sub-subseção 2.1.2), a mesa é sinalizada como **pronta** e encaminha a estação base uma requisição de retirada.

O estado **não pronta** é atribuído sempre que o cenário obtido pelas leituras descrito anteriormente não é obtido. O botão sempre que pressionado atribui o estado como **não pronta** e permanece assim por um tempo pré-determinado,

sendo definido atualmente em um minuto. Após este tempo, o estado da mesa é liberado para mudança.

A comunicação com a base é realizada através do protocolo MQTT (sub-seção 2.3.2) na rede *Wi-Fi* fornecida pela base.

### 3.4 Software - Robô

O *software* do Robô é responsável pela integração dos sensores, comunicação do robô com a base, acionamento da empilhadeira e navegação pela pista. O robô permanece na cozinha até que receba um *byte char* pela porta serial *Bluetooth*. Esse *byte* é então interpretado como um local que o robô deve buscar a bandeja.

Para a movimentação, foi desenvolvido um algoritmo PID que utiliza a leitura dos sensores infravermelho. A saída do algoritmo controla o valor de potência necessária em cada motor para seguir a linha da pista. Além disso, o sensor RFID na base do robô lê as *tags* que estão presentes em cada cruzamento. Através dos IDs dessas *tags*, o robô se localiza e sabe onde virar para chegar na bandeja desejada.

Na penúltima *tag* do percurso, o robô pára, emite um alerta sonoro com o Buzzer e checa o *buffer* do *Bluetooth* para ver se a base mandou um sinal de cancelamento. Caso tenha mandado, ele retorna para cozinha. Se não, ele se aproxima da mesa com velocidade reduzida até o fim da linha preta de forma a encaixar o garfo na bandeja.

Um outro sensor RFID presente no garfo faz a leitura da *tag* presente na bandeja para verificar se o robô está alinhado com ela. Quando o robô está alinhado, o software aciona o motor da empilhadeira até o fim de curso ser atingido. Depois, ele se afasta de ré, vira e retorna para a cozinha. Caso o robô não esteja alinhado, ele se afasta e tenta novas aproximações.

Ao chegar na cozinha com a bandeja, ele emite um alerta sonoro até que alguém pressione o botão de fim de curso, de forma a informar que a bandeja foi retirada. Por fim, ele envia um *char* de volta para a base sinalizando que está livre para a próxima tarefa.

### 3.5 Software - Base

A estação base é responsável por disponibilizar as interfaces de conexão com a mesa e com o robô. Para isso, a estação base (Raspberry Pi) fornece uma rede *Wi-fi*, que através de um protocolo MQTT (subseção 2.3.2) se conecta com as mesas. A comunicação com o robô é realizada através do seu módulo *bluetooth* interno, BCM4343. A interface gráfica fica disponível numa página web que pode ser acessada por qualquer dispositivo desde que esteja conectada na mesma estação base.

O primeiro software a ser desenvolvido foi o *backend*, um servidor REST[13] essencial para a alimentação de dados da interface gráfica de monitoração. Antes de iniciar o desenvolvimento, no entanto, foi estudada e projetada a solu-

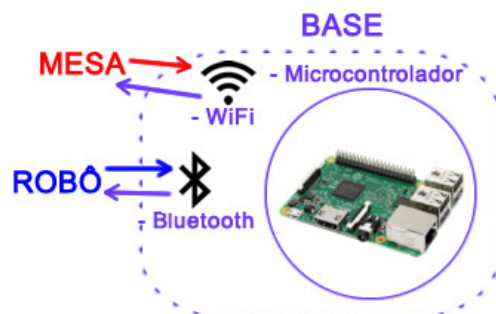


Figura 12: Visão geral da comunicação entre a base e as outras partes do sistema.

ção para fornecimento de dados visando as limitações de rede encontrada pela equipe. Foi decidido utilizar e trafegar todas as informações do servidor REST através da rede local fornecida pela estação base, assim assegurando a publicação das informações nos dispositivos conectados na rede.

O servidor REST foi desenvolvido em *FLASK* [14], um web framework de código aberto amplamente utilizado para *Python*. Essa escolha foi feita levando em consideração a sua velocidade, além de possuir um servidor *web* embutido e o rápido tempo de desenvolvimento da aplicação.

O protocolo MQTT (subseção 2.3.2) é utilizado em conjunto da rede *Wi-Fi* criada pela estação base, fazendo uso dos tópicos de publicação do protocolo para trafegar as informações originadas da mesa, estas informações são disponibilizadas para o servidor REST atualizar a interface gráfica.

As solicitações de retiradas são organizadas em uma fila de atendimento baseado no estado informado das mesas. A sequência de atendimento é montada conforme a ordem em que as solicitações chegam. Quando o botão é pressionado na mesa e existe uma solicitação de retirada para esta mesa, ela então é retirada/cancelada.

A base comunica, através de sua rede *Bluetooth*, o robô sobre qual mesa ele deve atender.

### 3.6 Testes de Avaliação



Figura 13: Visão geral do sistema montado.

A figura apresentada acima demonstra o sistema Trayker devidamente montado em seu ambiente de prova de conceito. As linhas demarcadas no chão são utilizadas para movimentação do robô e as mesas posicionadas no final de cada percurso. O robô realiza de forma satisfatória a locomoção pelos caminhos demarcados e o atendimento as solicitações de retiradas. Como a mudança da localidade onde o ambiente da prova de conceito é montada pode apresentar novas configurações de luminosidade o robô pode realizar algumas leituras errôneas e sair do trajeto. Contudo em testes realizados a ocorrência tem sido considerada baixa pela equipe.

### 3.7 Custos de Projeto

A tabela 2 apresenta o valor necessário para aquisição de todas as peças para elaboração do projeto do zero. O valor real gasto pela equipe, no entanto, foi inferior por emprestar e já possuir vários itens necessários.

Tabela 2: Custos do Projeto

|                             | Robô         | Mesa       | Bandeja   | Base       |
|-----------------------------|--------------|------------|-----------|------------|
| <b>Estrutura e Mecânica</b> | R\$ 623,60   | R\$ 52,50  | R\$ 50,00 | R\$ 0,00   |
| <b>Circuitaria</b>          | R\$ 645,71   | R\$ 200,95 | R\$ 0,00  | R\$ 532,00 |
| <b>Total da parte</b>       | R\$ 1.269,31 | R\$ 253,45 | R\$ 50,00 | R\$ 532,00 |
| <b>Total Geral</b>          | R\$ 1.924,76 |            |           |            |

## 4 Considerações finais

No decorrer da realização do projeto abordado foram encontradas diversas dificuldades, mas também houve a possibilidade de aprendizado de novas técnicas,

tecnologias e refinamento de conhecimentos vistos em outras disciplinas.

#### 4.1 Dificuldades

A mecânica da empilhadeira do robô que inicialmente previa o uso de um trilho de impressora demonstrou-se instável e o torque do motor era insuficiente para realizar a elevação do garfo, com a bandeja, dentro das especificações do projeto. Foi então utilizado um sistema de correção de gaveta que forneceu a estabilidade necessária e um sistema de polia em conjunto com um motor DC, assegurando que o torque do motor aguentasse realizar a ação de elevação dentro dos parâmetros especificados.

A alimentação do robô foi realizada através de *power banks* que se demonstrou um fonte de energia insatisfatória, que não apresentou características de duração, mesmo utilizando uma placa para ligação em série das baterias, para elevação tensão, e ligações em paralelo, para elevação da corrente, as portas de energização se sobrecarregam e as baterias tem sua vida útil reduzida drasticamente. A placa de circuito elétrico de elevação da tensão e corrente foi reutilizada do projeto Robcom.

Os motores utilizados para deslocamento do robô também apresentaram um funcionamento inesperado, sendo que para uma mesma potência fornecida ao conjunto de motores cada um apresentou uma velocidade diferente, o que causou problemas de desvio em seu trajeto. Foi então comparado individualmente cada velocidade do motor e combinado os dois motores, com velocidades mais próximas, posicionados na parte da frente do robô.

A calibração da balança gerou alguns comportamentos inesperados do *software* gerencial da mesa, como a célula de carga é conectada em um HX711, e este possui um tempo de decodificação relativamente alto em comparação com os outros sensores, a performance das medidas ficou reduzida ao seu tempo.

O controle de movimentação do robô através de um controlador PID consumiu um demasiado tempo, apresentando por parte do integrante da equipe um dificuldade no entendimento da parte teórica, implementação em linguagem computacional e teste. Encontrar os parâmetros que atendessem as necessidades do robô gerou atrasos no cronograma, que posteriormente foram compensados.

A confecção das placas de circuitos impressos para interfaceamento da ESP32 gerou uma grande carga de retrabalho. Foi necessário a construção de três placas para substituição das iniciais, além de um novo processo de transferência do circuito para a placa que apresentou um resultado muito superior ao que foi ensinado para os alunos na matéria de Circuitos Digitais, que dispensava a utilização da sabão.

## 4.2 Conclusão Final

Conclui-se com a finalização do projeto que além das dificuldades apresentadas a necessidade de uma interação interpessoal produtiva, organização e execução de tarefas, coordenação da equipe exige de todos um deslocamento de suas áreas de conforto em detrimento ao desenvolvimento do projeto.

## Referências

- [1] José Morais. Conhecendo o esp32. <https://portal.vidadesilicio.com.br/conhecendo-o-esp32/>.
- [2] Vitor Vidal. Ultrassônico hc-sr04. <http://blog.eletrogate.com/sensor-ultrassonico-hc-sr04-com-arduino/>.
- [3] Build a digital Raspberry Pi Scale (with Weight Sensor HX711). <https://tutorials-raspberrypi.com/digital-raspberry-pi-scale-weight-sensor-hx711/>.
- [4] Baú da Eletrônica. Conversor hx711. <http://blog.baudaeletronica.com.br/conversor-hx711-para-balanca-eletronica/>.
- [5] Euler Oliveira. Sensor rfid mrfc522 <http://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-kit-rfid-mfrc522/>.
- [6] Fabio Souza. Primeiros passos com micropython no esp8266 <https://www.embarcados.com.br/micropython-no-esp8266/>.
- [7] Fabio Souza. Arduino mega 2560 <https://www.embarcados.com.br/arduino-mega-2560/>.
- [8] Usinainfo. Sensor de linha infravermelho bfd-1000 com 5 sondas <https://tinyurl.com/usinainfo/>.
- [9] Wikipédia. Raspberry pi 3 b+ <https://www.embarcados.com.br/nova-raspberry-pi-3-modelo-b/>.
- [10] Wikipédia. Mqtt <https://pt.wikipedia.org/wiki/mqtt>.
- [11] Wikipédia. Bluetooth <https://pt.wikipedia.org/wiki/bluetooth>.
- [12] Lucas C. Tavano Lucas K. Amin Rubia T. Z. Isobe Ewerton G. Fiel, Guilherme E. de Souza. Robcom, o robô garçom. <https://roborobcom.wordpress.com/>.
- [13] Code Academy. What is rest. <https://www.codecademy.com/articles/what-is-rest>.
- [14] Armin Ronacher. Flask. <http://flask.pocoo.org>.