

---

**Etec "Profª Anna de Oliveira Ferraz"**

**ROBÔ TRANSPORTADOR DE PEQUENAS CARGAS (RTPC)**  
***SMALL LOAD TRANSPORT ROBOT***

Arthur Grego Maduro– Arthurmaduroz@gmail.com

Davi Rodrigues Pedracho Junior- davirodriguesjunior4@gmail.com

Gustavo Daniel Mutti- gustavomutti298@gmail.com

José Lucas Lacerda Cariri- joselucas122005@gmail.com

Etec Profª Anna de Oliveira Ferraz – Araraquara – São Paulo – Brasil

Edgar Bergo Coroa– edgar.coroa@etec.sp.gov.br

Flávio Tadeu Lourencetti- flavio.lourencetti01@etec.sp.gov.br

Etec Profª Anna de Oliveira Ferraz – Araraquara – São Paulo – Brasil

**RESUMO**

O protótipo do Robô Transportador de Pequenas Cargas (RTPC) surgiu para enfrentar os desafios logísticos e de transporte em áreas industriais, com possibilidade de aplicação em ambientes domésticos. Desenvolvido para movimentar cargas de até 1 kg, o RTPC visa oferecer uma solução eficiente, acessível e prática, otimizando o transporte de pequenos itens e promovendo maior produtividade e ergonomia. O objetivo do trabalho foi criar um dispositivo funcional, capaz de atender às demandas de logística interna em indústrias e centros de distribuição, utilizando tecnologia acessível e técnicas de automação. A metodologia incluiu pesquisa bibliográfica e consulta a projetos semelhantes, com destaque para o uso de componentes como o microcontrolador ESP32, sensores ultrassônicos de distância e motores de corrente contínua (CC), além da programação em C++ para controle via Bluetooth. A construção contou com a fabricação de um chassi de acrílico projetado no software TinkerCad, montagem prática dos componentes, interligação por uma protoboard e integração da programação com o microcontrolador. Os testes demonstraram que o RTPC atende aos requisitos propostos, incluindo parada automática ao detectar obstáculos, robustez do chassi para suportar o peso e precisão no controle remoto. Apesar disso, foram enfrentados problemas como danos a componentes durante a montagem, evidenciando a necessidade de maior atenção na instalação elétrica. Como melhoria futura, sugere-se a implementação de navegação autônoma, eliminando a dependência de controle remoto e tornando o sistema ainda mais eficiente. Conclui-se que o RTPC é uma contribuição relevante para a logística e o transporte, destacando o aprendizado em robótica, programação e eletromecânica.

**Palavras-chave:** Robótica. Logística. Automação. Transporte. Programação C++. Microcontroladores. Sensores.

## ABSTRACT

The prototype of the Small Load Transport Robot (SLTR) was created to address logistical and transportation challenges in industrial areas, with potential applications in domestic environments. Designed to move loads of up to 1 kg, the RTPC aims to provide an efficient, accessible, and practical solution, optimizing the transport of small items and promoting greater productivity and ergonomics. The goal of the project was to create a functional device capable of meeting the internal logistics demands of industries and distribution centers, using accessible technology and automation techniques. The methodology included bibliographic research and consultation of similar projects, focusing on components such as the ESP32 microcontroller, ultrasonic distance sensors, and DC motors, along with C++ programming for Bluetooth control. The construction involved manufacturing an acrylic chassis designed in TinkerCad, practical assembly of the components, interconnection via a breadboard, and integration of the programming with the microcontroller. The tests demonstrated that the RTPC met the proposed requirements, including automatic stop when detecting obstacles, chassis robustness to support the weight, and accuracy in remote control. However, issues such as component damage during assembly were encountered, highlighting the need for greater attention to electrical installation. As a future improvement, the implementation of autonomous navigation is suggested, eliminating the dependence on remote control and making the system even more efficient. In conclusion, the RTPC is a relevant contribution to logistics and transportation, highlighting the learning in robotics, programming, and electromechanics.

**Keywords:** Robotics. Logistics. Automation. Transportation. C++ Programming. Microcontrollers. Sensors.

## 1 INTRODUÇÃO

De acordo com Jornal da USP (2020), o projeto original Delivery Robot, foi iniciado em maio de 2019 pelo Grupo de Mobilidade da Escola Politécnica da USP. Em seu principal objetivo, foi desenvolvido um robô autônomo para executar atividades administrativas, como o transporte de documentos dentro do próprio campus da universidade.

O Robô Transportador (RT) tem em seu propósito a eficiência e produtividade, tendo um robô que carregue materiais que seu operador desejar, transportando cargas leves quanto pesadas. Dessa forma, ele obedece às ordens de seu portador por meio de ações básicas, utilizando um dispositivo móvel para se parear com o microcontrolador via bluetooth.

Essas informações destacaram que dois fatores estavam impactando na realização do projeto. O primeiro foi o próprio chassi, se este era capaz de suportar o peso imposto sobre ele. O segundo fator foi a definição dos componentes a serem utilizados no protótipo.

O RTPC desenvolvido neste artigo, foi projetado para movimentar pequenas cargas de

---

## Etec "Profª Anna de Oliveira Ferraz"

até 1 kg e terá seu impacto significativo no dia a dia das pessoas, como levar sacolas de compras, dentro de indústrias para a locomoção de peças, evitando contigências e aumentando a ergonomia.

## 2 FUNDAMENTAÇÃO TEÓRICA

Notoriamente, os avanços da robótica e da automação estão presentes nos dias de hoje, sendo sua maior participação em ambientes industriais, com foco na logística interna e ao transporte de materiais.

Dessa forma, é importante destacar os sistemas de veículos guiados automatizados (AGVs) cujo princípio é aumentar a eficiência e reduzir custos operacionais em fábricas e centros de distribuição.

De acordo com Le-Anh e De Koster (2006), "os AGVs são veículos autônomos que seguem rotas predefinidas ou adaptativas dentro de fábricas, armazéns e outros ambientes industriais para transportar materiais de um ponto a outro". Ou seja, a movimentação desses veículos é baseada em ordens de seus operadores ou até mesmo uma programação avançada capaz de arquitetar sua própria rota.

Os autores citam diferentes métodos de navegação utilizados em AGVs. Dentre os citados podemos destacar os trilhos físicos e a navegação baseada em sensores (visão computacional). No entanto, a solução mais procurada para aplicação em lugares industriais são os que possuem maior flexibilidade, permitindo operar de forma autônoma em ambientes dinâmicos, sem a demanda de infraestrutura física fixa.

O livro *Probabilistic Robotics* aborda sistemas de navegação e controle, com ênfase na localização e controle de robôs móveis autônomos, citando a incerteza ao ambiente e aos sensores utilizados nos robôs, propondo algoritmos que consigam lidar contra essas incertezas (THRUN et al., 2005).

Além disso, são apresentados algoritmos capazes de ajudar na localização e no funcionamento dos robôs. Esses algoritmos são; Filtros de Kalman, Filtros de Partículas e o algoritmo Monte Carlo Localization (MCL), que são amplamente utilizados para a localização em tempo real. Esses métodos ajudam os robôs a entenderem sua posição atual com base em dados de sensores e movimentos anteriores.

## Etec "Profª Anna de Oliveira Ferraz"

No artigo “robôs móveis na logística”, são mencionados os AMRs (*Automated Mobile Robots*) que são uma evolução dos AGVs. Os AMRs utilizam mapas digitalizados e sensores para ter sua navegação mais autônoma, assim, permitindo que eles sejam completamente mais flexíveis e dinâmicos na hora do trabalho. (PELEGRINI, Camila et al., 2023)

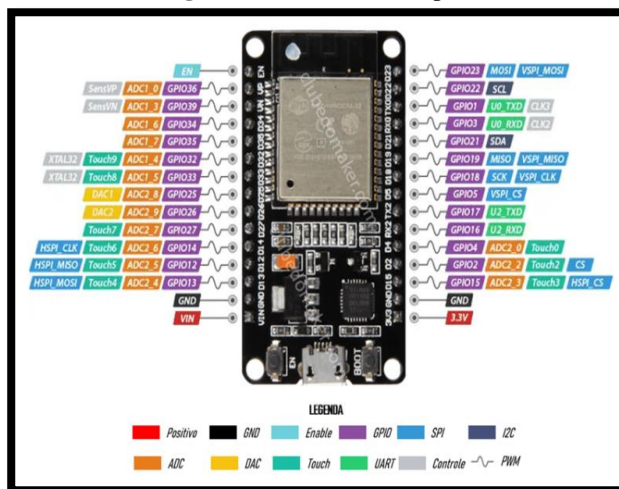
Mesmo com a evolução dos AGVs, os autores relatam que os robôs autônomos ainda possuem muitos benefícios para ser utilizados nas indústrias, “A utilização dos robôs AGVs para a movimentação de cargas resulta em redução de custos operacionais e aumento de produtividade, tornando-se indispensáveis em indústrias com grande volume de movimentação, como a automotiva.” (PELEGRINI, Camila et al., 2023, p.95)

Para o controle dos motores presentes nos AGVs, é amplamente utilizado a ponte H L298N, pois é um circuito integrado capaz de controlar a tensão em dois motores de corrente contínua (CC). Ele opera com duas pontes H integradas dentro do próprio componente, permitindo a alteração do sentido de rotação dos motores, capaz de aguentar tensões de até 46V (Volts) e correntes de até 2A (Amperes). (STMICROELECTRONICS, 2000)

Com o intuito de comandar todo o projeto, são necessários microcontroladores capazes de oferecer todas as funções essenciais. Dentre os existentes, o ESP32 se destaca.

Segundo o Guia prático ESP32 (Fig.1), possui uma variedade de pinos tanto analógicos quanto digitais, aptos para a transferência de dados para os demais componentes. Ele conta com dois processadores que facilitam o processo de execução de várias tarefas simultâneas e também oferece o modo de economia de energia, para projetos que exigem muito gasto energético (FERREIRA, Aida Araújo et al., 2024).

Figura 1 – ESP32 e seus pinos



Fonte: Clube do maker, 2024.

**Etec "Profª Anna de Oliveira Ferraz"**

### 3 PROCEDIMENTOS METODOLÓGICOS

Foi realizado pesquisas bibliograficas relacionadas ao tema juntamente a vídeos e busca por projetos semelhantes. Foram os maiores métodos utilizados para a produção do projeto, sendo assim possível obter referências de materiais e montagens para a realização. Com base nesse estudo, o RTPC possui em seus componentes duas placas de acrílico de 250x250mm para o chassi, ambas com 5mm de espessura, um ESP32 para comandar todo o robô, duas ponte H duplas L 298N controlando a velocidade de rotação dos motores, uma bateria de furadeira 5V para a alimentação, uma bateria recarregável, uma protoboard mega para interligar todos os componentes, um módulo sensor de distância ultrassônico (HC-SR04) com o intuito de medir a distância de um obstáculo e parar os motores se houver risco de colisão e quatro motores micro controlador com redutor DC integrado as rodas. A seguir o quadro 1 ilustra os preços de cada item citado acima.

**Quadro 1 - Preços e componentes**

<b>Material</b>	<b>Preço no mercado</b>	<b>Quantidade total</b>	<b>Preço no total</b>
Placa de acrilico	R\$75,00	2	R\$150,00
ESP32	R\$40,00	1	R\$40,00
Ponte HL298N	R\$15,00	2	Os três vieram junto em um kit
Motor redutor	R\$15,00	4	
Roda	R\$5,00	4	
Protoboard mega	R\$60,00	1	R\$60,00
Sensor HCR04	R\$20,00	1	R\$20,00
<b>TOTAL</b>			<b>R\$352,00</b>

Fonte: Autores, (2024).

#### 3.1 Programação C++

A programação foi feita a partir do Arduino IDE (*Integrated Development Environment*), visto que o aplicativo era o mais acessível para realizar as linguagens de código. Foi necessário baixar uma nova biblioteca para utilizar a função Bluetooth do microcontrolador, integrando-o ao dispositivo móvel na mesma rede.

## Etec "Prof.ª Anna de Oliveira Ferraz"

```
#include <BluetoothSerial.h>
BluetoothSerial SerialBT;

// Definindo os pinos da ponte H
#define IN1 12 // Controle Motor 1 (frente)
#define IN2 14 // Controle Motor 1 (trás)
#define IN3 26 // Controle Motor 2 (frente)
#define IN4 27 // Controle Motor 2 (trás)

int velocidade = 128; // Velocidade inicial (0-255)

void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  // Configuração do Bluetooth
  SerialBT.begin("ESP32_Carrinho");
  Serial.begin(9600);

  // Configuração do PWM
  ledcSetup(0, 5000, 8); // Canal 0, 5kHz, 8 bits
  ledcAttachPin(IN1, 0);

  ledcSetup(1, 5000, 8); // Canal 1, 5kHz, 8 bits
  ledcAttachPin(IN3, 1);

  ledcSetup(2, 5000, 8); // Canal 2, 5kHz, 8 bits
  ledcAttachPin(IN2, 2);

  ledcSetup(3, 5000, 8); // Canal 3, 5kHz, 8 bits
  ledcAttachPin(IN4, 3);
}

void loop() {
  // Verificando comandos Bluetooth
  if (SerialBT.available()) {
    char command = SerialBT.read();
    executeCommand(command);
  }
}

void executeCommand(char command) {
  switch (command) {
    case 'F':
      moveForward();
      break;
    case 'B':
      moveBackward();
      break;
    case 'L':
      moveLeftward();
      break;
    case 'R':
      moveRightward();
      break;
    case 'S':
      stopMotors();
      break;
    case '+': // Aumentar velocidade
      if (velocidade < 255) velocidade += 10;
      Serial.print("Velocidade: ");
      Serial.println(velocidade);
      break;
    case '-': // Diminuir velocidade
      if (velocidade > 0) velocidade -= 10;
      Serial.print("Velocidade: ");
```

```
Serial.println(velocidade);
      break;
    default:
      stopMotors();
      break;
  }
}

void moveForward() {
  digitalWrite(IN2, LOW); // Motor 1 (trás) desativado
  digitalWrite(IN4, LOW); // Motor 2 (trás) desativado
  digitalWrite(IN1, HIGH); // Motor 1 (frente) ativado
  digitalWrite(IN3, HIGH); // Motor 2 (frente) ativado
  ledcWrite(0, velocidade); // Definindo velocidade do motor
  1
  ledcWrite(1, velocidade); // Definindo velocidade do motor
  2
  Serial.println("Movendo para frente");
}

void moveBackward() {
  digitalWrite(IN1, LOW); // Motor 1 (frente) desativado
  digitalWrite(IN3, LOW); // Motor 2 (frente) desativado
  digitalWrite(IN2, HIGH); // Motor 1 (trás) ativado
  digitalWrite(IN4, HIGH); // Motor 2 (trás) ativado
  ledcWrite(2, velocidade); // Definindo velocidade do motor
  1
  ledcWrite(3, velocidade); // Definindo velocidade do motor
  2
  Serial.println("movendo para trás");
}

void moveLeftward() {
  digitalWrite(IN1, LOW); // Motor 1 (frente) desativado
  digitalWrite(IN2, LOW); // Motor 1 (trás) desativado
  digitalWrite(IN3, HIGH); // Motor 2 (frente) ativado
  digitalWrite(IN4, HIGH); // Motor 2 (trás) ativado
  ledcWrite(2, velocidade); // Definindo velocidade do motor
  1
  ledcWrite(3, velocidade); // Definindo velocidade do motor
  2
  Serial.println("movendo para esquerda");
}

void moveRightward() {
  digitalWrite(IN3, LOW); // Motor 2 (frente) desativado
  digitalWrite(IN4, LOW); // Motor 2 (trás) desativado
  digitalWrite(IN1, HIGH); // Motor 1 (frente) ativado
  digitalWrite(IN2, HIGH); // Motor 1 (trás) ativado
  ledcWrite(2, velocidade); // Definindo velocidade do motor
  1
  ledcWrite(3, velocidade); // Definindo velocidade do motor
  2
  Serial.println("movendo para direita");
}

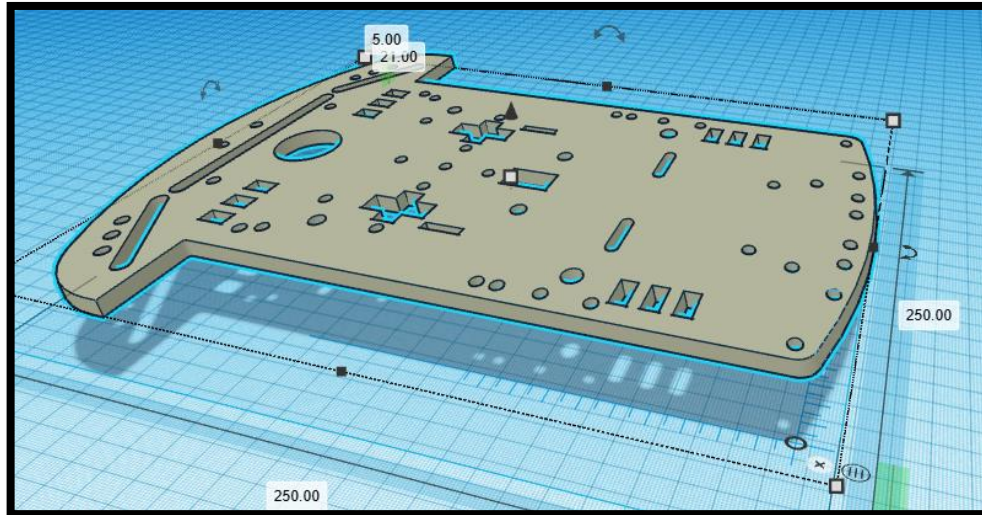
void stopMotors() {
  digitalWrite(IN3, LOW); // Motor 2 (frente) desativado
  digitalWrite(IN4, LOW); // Motor 2 (trás) desativado
  digitalWrite(IN1, LOW); // Motor 1 (frente) ativado
  digitalWrite(IN2, LOW); // Motor 1 (trás) ativado
  Serial.println("parando os motores");
}
```

## Etec "Prof.<sup>a</sup> Anna de Oliveira Ferraz"

### 3.2 Montagem Prática

Primeiramente, foi projetado o molde do chassi no software TinkerCad (Fig.2), após isso, foi encaminhado para a fabricação utilizando cortes a laser.

Figura 2 – Molde do chassi



Fonte: Autores, (2024).

Com a chegada de todos os componentes do RTPC, foram feitas as demais organizações para a melhor montagem (Fig. 3).

Figura 3 – Organizações dos componentes

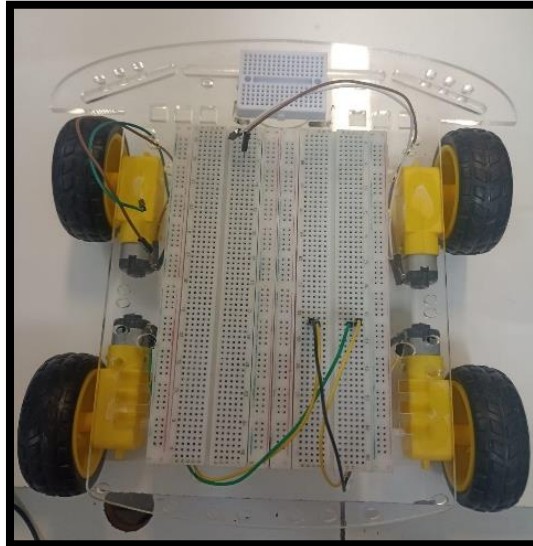


Fonte: Autores, (2024)

### Etec "Prof.<sup>a</sup> Anna de Oliveira Ferraz"

Nesta etapa, iniciou-se a montagem junto com a protoboard em cima do chassi, referente na figura 4.

**Figura 4** – Montagem dos componentes no chassi



Fonte: Autores, (2024)

Com a instalação realizada, a programação foi elaborada através da plataforma Arduino IDE, respectivo a figura 5

**Figura 5** –Programação C++

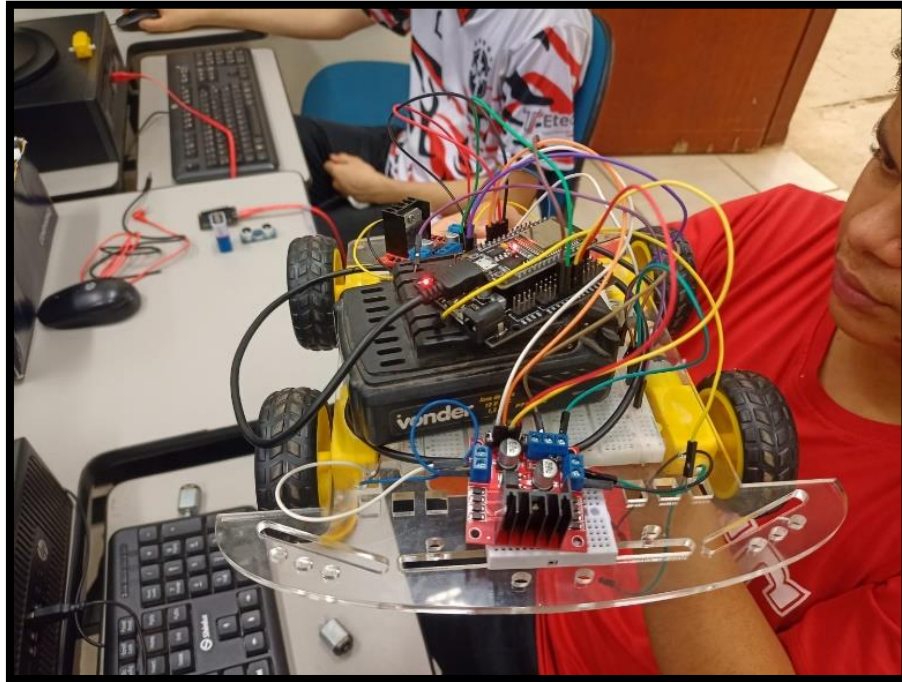
```
pinrest | Arduino IDE 1.3.3
File Edit Sketch Tools Help
ESP32 Wrover Module
SKETCHBOOK
pinrest.ino
1 #include <BluetoothSerial.h>
2 BluetoothSerial SerialBT;
3
4 // Definindo os pinos da ponte H
5 #define IN1 12 // Controle Motor 1 (frente)
6 #define IN2 14 // Controle Motor 1 (trás)
7 #define IN3 26 // Controle Motor 2 (frente)
8 #define IN4 27 // Controle Motor 2 (trás)
9
10 int velocidade = 128; // Velocidade inicial (0-255)
11
12 void setup() {
13   pinMode(IN1, OUTPUT);
14   pinMode(IN2, OUTPUT);
15   pinMode(IN3, OUTPUT);
16   pinMode(IN4, OUTPUT);
17
18   // Configuração do Bluetooth
19   SerialBT.begin("ESP32_Carrinho");
20   Serial.begin(9600);
21
22   // Configuração do PWM
23   ledcSetup(0, 5000, 8); // Canal 0, 5kHz, 8 bits
24   ledcAttachPin(IN1, 0);
25
26   ledcSetup(1, 5000, 8); // Canal 1, 5kHz, 8 bits
27   ledcAttachPin(IN3, 1);
28
29   ledcSetup(2, 5000, 8); // Canal 2, 5kHz, 8 bits
30   ledcAttachPin(IN2, 2);
31
32   ledcSetup(3, 5000, 8); // Canal 3, 5kHz, 8 bits
33   ledcAttachPin(IN4, 3);
34
35
36 void loop() {
37   // Verificando comandos Bluetooth
38   if (SerialBT.available()) {
39     char command = SerialBT.read();
40     executeCommand(command);
41   }
42 }
```

Fonte: Autores, (2024)

### Etec "Prof.ª Anna de Oliveira Ferraz"

Em seguida, a instalação de todos os componentes juntamente com o microcontrolador ESP32 com a programação em sua memória, mostrado na figura 6.

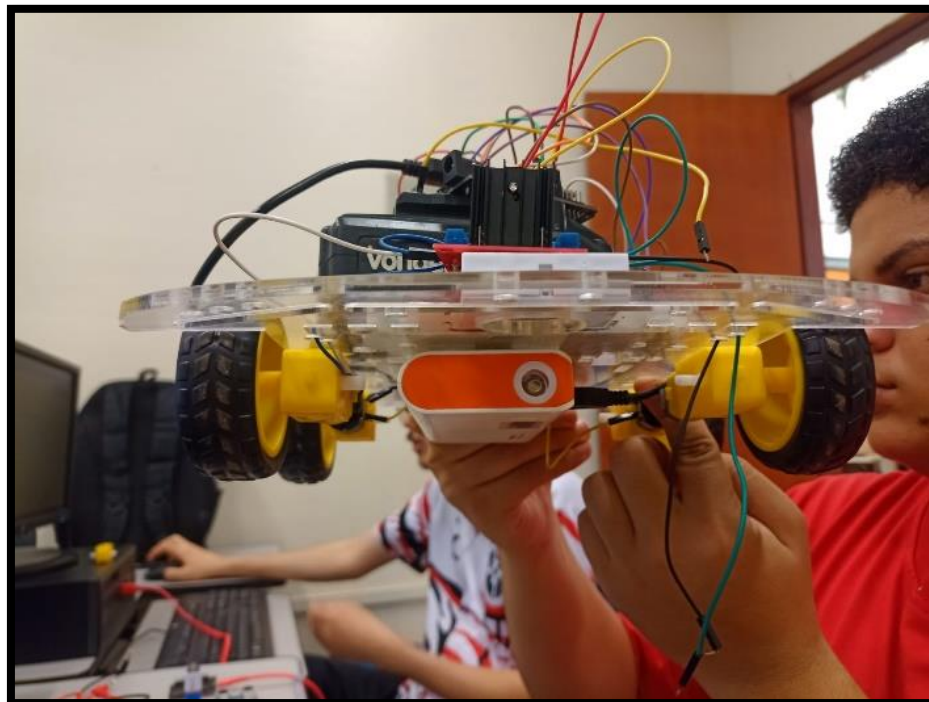
**Figura 6** – instalação dos componentes com a programação



Fonte: Autores, (2024)

Após, a bateria recarregável foi instalada em baixo do protótipo, visto na figura 7.

**Figura 7** – bateria recarregável instalada



Fonte: Autores, (2024)

### Etec "Prof.<sup>a</sup> Anna de Oliveira Ferraz"

Com a parte de baixo finalizada, a caixa da proteção do RTPC foi começada, utilizando o material MDF (*Medium Density Fiberboard*), este material possui em suas medidas 255x250mm para a parte de baixo como a de cima e 105x250mm suas laterais, referente a figura 8

**Figura 8** - caixa de proteção



**Fonte:** Autores, (2024)

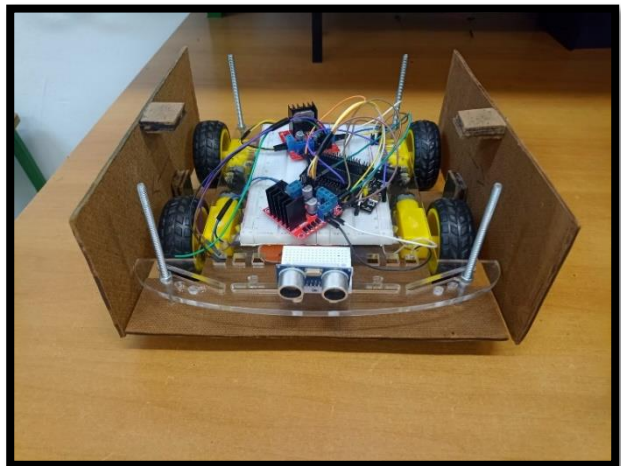
Em seguida, foi feito 4 furos nos dois chassis, furos com 45mm de diâmetro, assim podendo passar os parafusos na caixa de proteção, e também, na caixa foi realizada esses furos. (fig 9a e 9b)

**Figura 9a** – furos no chassi



**Fonte:** Autores, (2024)

**Figura 9b** – parafusos no chassi e na caixa



**Fonte:** Autores, (2024)

### **Etec “Prof.ª Anna de Oliveira Ferraz”**

Após as verificações dos furos, o processo de pintura da caixa foi iniciado para que o RTPC tenha uma aparência de uma abelha, visto na figura 10.

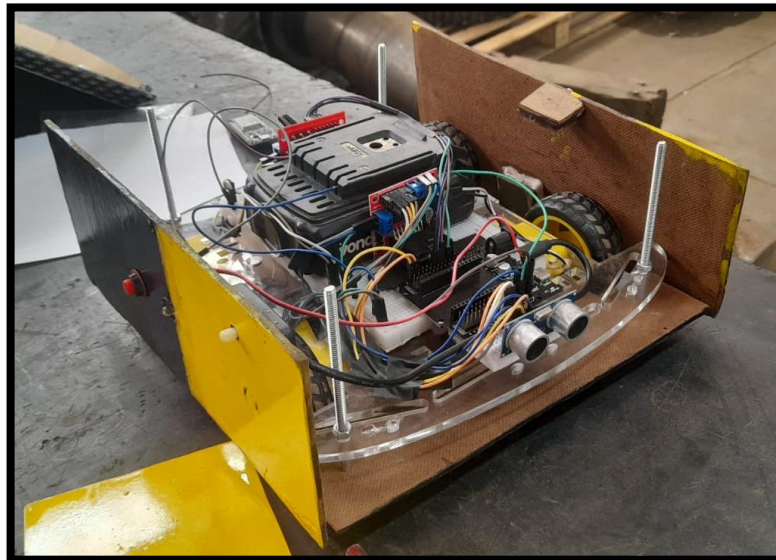
**Figura 10** – Pintura da caixa



**Fonte:** Autores, (2024)

Com a pintura realizada, a montagem da caixa foi realizada pelas laterais em contato com sua base. (Fig 11)

**Figura 11** – montagem da caixa pelas laterais



**Fonte:** Autores, (2024)

### Etec “Prof.<sup>a</sup> Anna de Oliveira Ferraz”

Por último, as partes dianteira e traseira foram acopladas, seguidamente da parte de cima aonde é o suporte para manter a carga em segurança, mostrado na figura 12.

**Figura 12** – Caixa finalizada sobre o RTPC



Fonte: Autores, (2024)

## 4 RESULTADOS E DISCUSSÃO

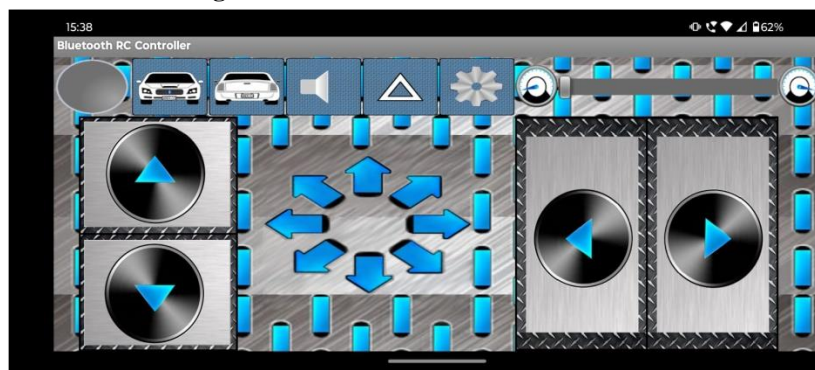
Após todo o procedimento de montagem dos componentes, foram realizados alguns testes para alcançar o melhor resultado sobre o RTPC.

Durante os testes, alguns componentes acabaram sendo danificados, sendo eles a ponte H L298N e dois motores, ocorreu devido a falta de atenção durante a montagem, assim, a energia foi maior do que ambos poderiam aguentar.

Para os testes, foram utilizados um celular com o app “RC controller bluetooth” (Fig 13) conectado no bluetooth do ESP32.

As respostas para os testes foram eficazes, pois a programação funcionou de forma satisfatória. O sensor de proximidade conseguiu parar os motores antes de uma possível colisão e o chassi junto com os motores foram capazes de aguentar o peso imposto sobre ele.

**Figura 13** – tela do RC controller bluetooth



Fonte: Autores, (2024)

## 5 CONSIDERAÇÕES FINAIS

Com o RTPC finalizado, o grupo teve uma maior noção sobre os conceitos da elétrica e eletromecânica envolvendo os motores, juntamente com o conhecimento e aprendizagem sobre os microcontroladores e a programação C++, mostrando que um robô transportador pode influenciar de forma positiva na logística e o transporte dentro de indústrias.

O grupo também deixa uma melhoria em aberto para quem quiser realiza-lá, seria a navegação autônoma sem o uso de um controle para o RTPC, deixando ainda mais o transporte robotizado.

## 6 AGRADECIMENTOS

O grupo gostaria de agradecer a instituição Etec Prof<sup>a</sup> Anna de Oliveira Ferraz e também, aos familiares de cada indivíduo, pois sem o apoio deles o grupo não conseguiria finalizar o protótipo, entre eles, o Márcio Mutti e Luciana Batista responsáveis pelo aluno Gustavo Mutti e especialmente a sua namorada, Julia Braga. Continuadamente, a Neusa Cariri e Rogerio Formigone, responsáveis pelo aluno José Cariri. Os responsáveis Davi Pedracho Rodrigues e Ariane Muratti, pais do aluno Davi Rodrigues. E por último, o sincero agradecimentos aos responsáveis do aluno Arthur Maduro, Sandra R. Grego Maduro e Nivaldo Donisete Maduro.

## REFERÊNCIAS

Disponível em: <https://clubedomaker.com/esp32-pinout>. Acesso em 04 nov 2024

FERREIRA, Aida Araújo. **Guia Prático ESP32**. Recife, Brasil: IFPE (Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco), 2023. Disponível em: <https://repositorio.ifpe.edu.br/xmlui/handle/123456789/1285>. Acesso em: 07 out. 2024

Jornal da USP. "**Robô Transportador Hospitalar**" inicia fase de testes no Hospital Universitário da USP, 2020. Disponível em: <https://jornal.usp.br/ciencias/robo-transportador-hospitalar-inicia-a-fase-de-testes-no-hospital-universitario-da-usp/#:~:text=O%20projeto%20original%2C%20chamado%20de,de%20documentos%20entre>

---

**Etec "Prof.ª Anna de Oliveira Ferraz"**

%20as%20unidades.. Acesso em: 12 abr. 2024.

LE-AHN. **A review of design and control of automated guided vehicle systems**. European Journal Of Operational Research. Amesterdã, p. 1-23. maio 2006. Disponível em: <https://doi.org/10.1016/j.ejor.2005.01.036>. Acesso em: 20 jun. 2024.

REZENDE, Camila Selles Pelegrini. **Robôs móveis na logística**. Jundiaí, São Paulo, 2023. Disponível em: <https://ric.cps.sp.gov.br/handle/123456789/15597>. Acesso em: 12 set. 2024.

STMICROELECTRONICS. **L298N Datasheet (HTML)**. Genébra: Stmicroelectronics, 1990. Disponível em: <https://www.alldatasheet.com/datasheet-pdf/pdf/22440/STMICROELECTRONICS/L298N.html>. Acesso em: 15 set. 2024.

THRUN, Sebastian. **Probabilistic robotics**. Cambridge, Massachusetts: Mit Press, 2005. Disponível em: [https://www.google.com.br/books/edition/Probabilistic\\_Robotics/2Zn6AQAQBAJ?hl=pt-BR&gbpv=0](https://www.google.com.br/books/edition/Probabilistic_Robotics/2Zn6AQAQBAJ?hl=pt-BR&gbpv=0). Acesso em: 25 set. 2024.