

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA  
SOUZA**

**ETEC SYLVIO DE MATTOS CARVALHO**

**Curso Técnico em Mecatrônica M-tec**

**CAUÃ HENRIQUE PRADO DE SOUZA**

**GABRIEL MARQUES DA SILVA**

**GUSTAVO DE SOUZA RAZO**

**PEDRO HENRIQUE BARBIERI DA SILVA**

**Projeto e Construção de um Robô Sumô Lutador para  
Competições: Aspectos Mecânicos e Eletrônicos**

**Matão, SP**

**2024**

## SUMÁRIO

|     |                                                           |    |
|-----|-----------------------------------------------------------|----|
| 1   | INTRODUÇÃO .....                                          | 3  |
| 2   | OBJETIVOS (GERAL E ESPECÍFICOS) .....                     | 3  |
| 3   | DESCRIÇÃO DO PROJETO .....                                | 4  |
| 3.1 | Apresentação Geral do Projeto .....                       | 4  |
| 3.2 | Materiais Utilizados e suas Especificações Técnicas ..... | 5  |
| 4   | CRONOGRAMA DE ATIVIDADES .....                            | 10 |
| 4.1 | Atividades e Metas 1º Semestre de 2024.....               | 10 |
| 4.2 | Atividades e Metas 2º Semestre de 2024.....               | 11 |
| 5.  | DESENVOLVIMENTO DO PROJETO.....                           | 12 |
| 5.1 | Descrição do Funcionamento.....                           | 12 |
| 5.2 | Montagem Física do Projeto .....                          | 14 |
| 5.3 | Desenvolvimento do programa do robô.....                  | 18 |
| 6   | CONSIDERAÇÕES FINAIS.....                                 | 20 |
|     | APÊNDICE .....                                            | 22 |

## **1 INTRODUÇÃO**

Este projeto busca desenvolver habilidades técnicas e criativas na construção e programação de robôs, aplicando conceitos de eletrônica, programação e mecânica. A ideia é criar um robô sumô, que participa de competições onde deve empurrar os adversários para fora de um ringue circular. O objetivo é, além de construir o robô, despertar o interesse dos alunos pela robótica e prepará-los para carreiras futuras na área de tecnologia.

O projeto envolve projetar a estrutura física do robô, escolhendo materiais resistentes e montando componentes como motores e rodas. Na parte elétrica, é preciso criar os circuitos que controlam o robô. A programação define como ele se comportará na arena, incluindo estratégias para atacar, se defender e evitar ser empurrado. A integração de todas essas partes exige testes para garantir que o robô funcione corretamente.

Além das habilidades técnicas, o projeto ensina a trabalhar em equipe e resolver problemas reais. Isso prepara os alunos para desafios que encontrarão na vida profissional. As habilidades adquiridas podem ser usadas em áreas como automação industrial e robótica de serviços, tornando este projeto uma ótima base para o futuro.

## **2 OBJETIVOS (GERAL E ESPECÍFICOS)**

### **2.1 Objetivo Geral**

O objetivo geral do projeto é:

- Construir um robô sumô.

### **2.2 Objetivos Específicos**

Os objetivos específicos do projeto são:

- Definir os componentes ideais para o robô;
- Elaborar o projeto estrutural do robô;
- Montar o robô, efetuando as conexões dos seus componentes;
- Desenvolver a programação do robô.

### 3 DESCRIÇÃO DO PROJETO

#### 3.1 Apresentação Geral do Projeto

O dispositivo é estruturado como um pequeno veículo, com duas esteiras de plástico localizadas em uma das extremidades do chassi, cada uma acionada por um motor de corrente contínua.

Uma placa de Arduino modelo Nano foi fixada no chassi, utilizando o microcontrolador Atmega 328P, que está programado para movimentar o dispositivo por meio de comandos enviados de um celular, que se comunica com o Arduino via um módulo bluetooth.

O robô é alimentado por uma bateria recarregável de 9V, que também foi acoplada ao chassi.

A Figura 1 apresenta uma foto do nosso projeto final montado.

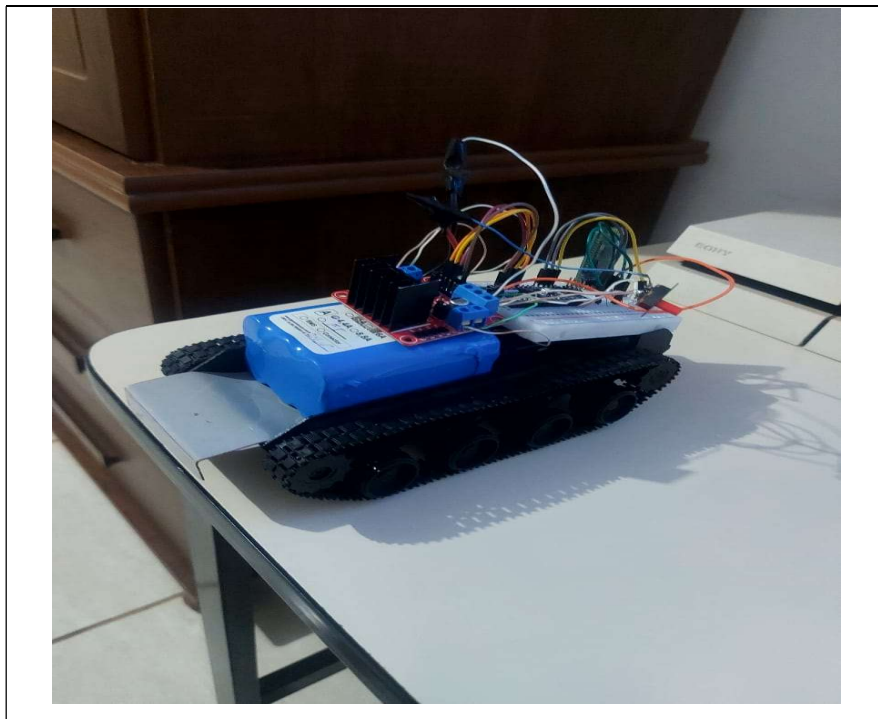


Figura 1 – Foto do projeto montado

FONTE: Elaborado pelos autores (2024).

### 3.2 Materiais Utilizados e suas Especificações Técnicas

O Arduino Nano é uma placa de prototipagem que contém o microcontrolador **ATmega328p**, um chip responsável por processar dados e executar comandos programados. A placa também inclui um **conversor USB-UART**, que facilita a comunicação com o computador, e componentes para regular a **alimentação e temporização** do circuito. No robô, o Arduino Nano é responsável por controlar os motores e gerenciar a comunicação sem fio, garantindo que o robô se mova corretamente de acordo com os comandos recebidos via Bluetooth. [2]

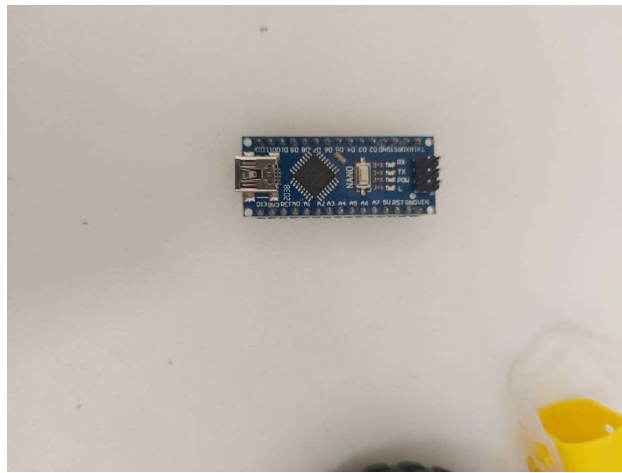


Figura 2: Modelo Microcontrolador Arduino Nano

FONTE: Elaborado pelos autores (2024).

O motor com redução é um componente que combina um motor elétrico com um conjunto de engrenagens para diminuir a velocidade de rotação, aumentando o torque. No robô, ele permite movimentos precisos e controlados, sendo utilizado para avançar, recuar e realizar curvas. A redução de velocidade melhora a força do motor, permitindo que o robô navegue por terrenos difíceis e execute giros de 360° com estabilidade. [3]



Figura 3: Modelo de Motor com Redutor  
FONTE: Elaborado pelos autores (2024).

A protoboard é uma ferramenta de prototipagem essencial que facilita a montagem de circuitos temporários sem a necessidade de solda. Ela permite conectar componentes eletrônicos, como resistores, capacitores e circuitos integrados, de forma organizada. Neste projeto, a protoboard é utilizada para conectar o módulo Bluetooth e outros elementos do robô, permitindo testes e ajustes antes da montagem final, assegurando a confiabilidade do circuito. [4]

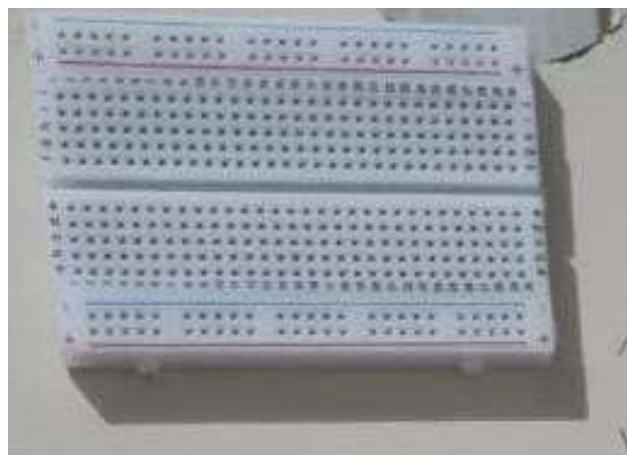


Figura 4: Modelo de Protoboard  
FONTE: Elaborado pelos autores (2024).

O módulo Bluetooth HC-05 é um dispositivo de comunicação sem fio que permite enviar e receber dados via Bluetooth. No projeto do robô, ele opera no modo **Master**, enviando comandos diretamente para o Arduino para controlar o movimento. O HC-05 é fundamental para a comunicação remota, permitindo que o robô se mova para frente, para trás e execute curvas com precisão, tudo através de comandos recebidos do usuário. [5]

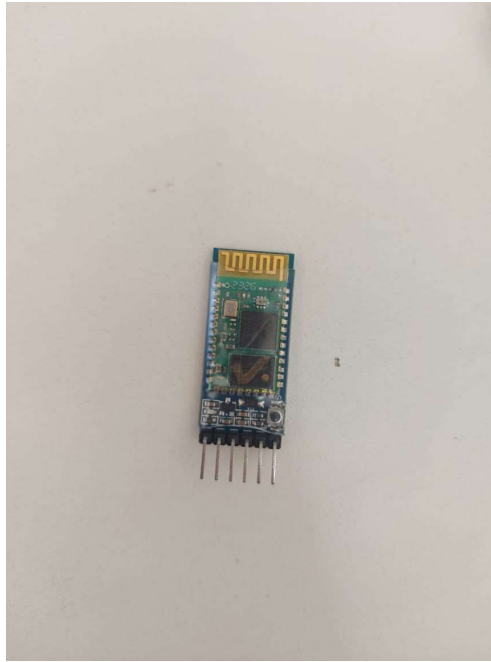


Figura 5: Modelo Módulo Bluetooth HC-05

FONTE: Elaborado pelos autores (2024).

A bateria é a fonte de energia do sistema, fornecendo uma tensão nominal de 12 volts para alimentar o circuito. A tensão pode variar de 12,6 a 12,8 volts dependendo do tipo de bateria utilizada (por exemplo, íon-lítio ou alcalina). No robô, a bateria alimenta o Arduino, os motores e outros componentes eletrônicos, garantindo que o sistema funcione de maneira estável e contínua. A escolha de uma bateria adequada é crucial para fornecer energia suficiente e manter o robô operando pelo tempo necessário. [6]



Figura 6: Modelo Bateria De 12 Volts

FONTE: Elaborado pelos autores (2024).

A esteira substitui as rodas tradicionais do robô, oferecendo uma tração mais estável e suave em terrenos variados. Feita de um material resistente e flexível, a esteira é montada em torno de roletes, que são acionados por motores elétricos de alta eficiência. Este sistema de tração melhora a aderência do robô em superfícies irregulares, permitindo movimentos fluidos e contínuos, além de facilitar manobras complexas, como giros completos. [7]



Figura 7: Modelo Esteira

FONTE: Elaborado pelos autores (2024).

A Ponte H é um circuito eletrônico projetado para controlar motores de corrente contínua (DC), permitindo tanto a variação da velocidade quanto o sentido de rotação. A estrutura da Ponte H possibilita a reversão da polaridade nos motores, controlando seu movimento para frente ou para trás. No robô, a Ponte H recebe comandos do Arduino e direciona a energia da bateria aos motores conforme as instruções recebidas, ajustando a direção e velocidade do robô de acordo com os comandos do Bluetooth. Ela também pode ser usada em outros contextos, como controlar geradores DC ou como inversor de corrente. [8]

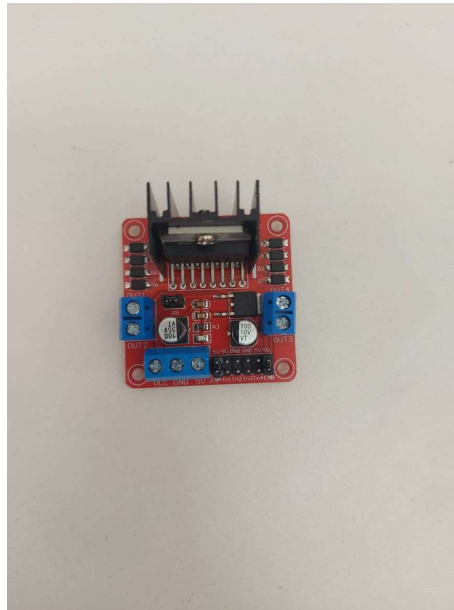


Figura 8: Modelo Driver Ponte H L298

FONTE: Elaborado pelos autores (2024).

Os materiais utilizados em nosso projeto estão apresentados na Tabela 1.

Tabela 1 – Lista de Materiais

| Material               | Quantidade |
|------------------------|------------|
| Arduino Nano           | 1          |
| Motores com Redutor    | 2          |
| Módulo Bluetooth HC-05 | 1          |
| Bateria                | 1          |
| Esteira                | 2          |
| Driver Ponte H L298    | 1          |
| Protoboard             | 1          |

FONTE: Elaborado pelos autores (2024).

## 4 CRONOGRAMA DE ATIVIDADES

### 4.1 Atividades e Metas 1º Semestre de 2024

As atividades do 1º semestre foram a montagem da equipe, e a, definição de ideias juntamente ao estudo, para a primeira batalha para vermos o funcionamento do robô, já as metas foram a montagem do protótipo durante as aulas junto com sua programação e o seu funcionamento, juntamente com o desenvolvimento do boneco.

Tabela 2: Cronograma de atividades e metas no 1º Semestre de 2024

| Atividade                     | Fev.         |              | Março        |              | Abril        |              | Maio         |              | Junho        |              |
|-------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                               | 1ª Quinz ena | 2ª Quinz ena | 1ª Quinz ena | 2ª Quinz ena | 1ª Quinz ena | 2ª Quinz ena | 1ª Quinz ena | 2ª Quinz ena | 1ª Quinz ena | 2ª Quinz ena |
| Formação da equipe            | x            |              |              |              |              |              |              |              |              |              |
| Definição do nome do robô     |              | x            |              |              |              |              |              |              |              |              |
| Definição de metas            |              | x            |              |              |              |              |              |              |              |              |
| Lista de materiais            |              |              | x            |              |              |              |              |              |              |              |
| Construção do chassi          |              |              |              | x            |              |              |              |              |              |              |
| Relatório técnico             |              |              |              | x            | x            |              |              |              |              |              |
| Desenvolvimento do capítulo 3 |              |              |              |              | x            | x            | x            | x            |              |              |
| Montagem                      |              |              |              |              | x            | x            | x            | x            |              |              |
| Programação robótica          |              |              |              |              |              |              | x            | x            |              |              |
| Batalha de Robôs              |              |              |              |              |              |              |              |              | x            | x            |

FONTE: Elaborado pelos autores (2024).

## 4.2 Atividades e Metas 2º Semestre de 2024

As atividades do 2º semestre estão de acordo com o cronograma abaixo, já as metas foram a finalização do boneco junto com o diário de bordo, e a inicialização e finalização do protótipo e seu funcionamento, para a participação da competição entre robôs na 2ª segunda quinzena de novembro.

Tabela 3: Cronograma de atividades e metas no 2º Semestre de 2024

| Atividade                          | Julho              |                    | Agosto             |                    | Setembro           |                    | Outubro            |                    | Novembro           |                    |
|------------------------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
|                                    | 1ª<br>Quinze<br>na | 2ª<br>Quinze<br>na | 1ª<br>Quinze<br>na | 2ª<br>Quinze<br>na | 1ª<br>Quinze<br>na | 2ª<br>Quinze<br>na | 1ª<br>Quinze<br>na | 2ª<br>Quinze<br>na | 1ª<br>Quinze<br>na | 2ª<br>Quinze<br>na |
| Criação do protótipo               | x                  | x                  | x                  | x                  | x                  | x                  | x                  | x                  | x                  |                    |
| Avaliação do protótipo             |                    |                    |                    |                    | x                  | x                  | x                  | x                  | x                  |                    |
| Desenvolvimento do sistema         | x                  | x                  | x                  | x                  | x                  | x                  | x                  |                    |                    |                    |
| Ajustes finais                     |                    |                    |                    |                    |                    |                    | x                  | x                  | x                  |                    |
| Apresentação final                 |                    |                    |                    |                    |                    |                    |                    |                    | x                  | x                  |
| Desenvolvimento do boneco          | x                  | x                  | x                  | x                  | x                  | x                  | x                  | x                  | x                  | x                  |
| Entrega final do relatório técnico |                    |                    |                    |                    |                    |                    |                    |                    |                    | x                  |
| Competição de robôs                |                    |                    |                    |                    |                    |                    |                    |                    |                    | x                  |

FONTE: Elaborado pelos autores (2024).



A montagem permite controlar os motores do robô por comandos enviados via Bluetooth para o Arduino Nano. A ponte H funciona como intermediária, recebendo sinais do Arduino e controlando a energia da bateria que vai para os motores. O módulo Bluetooth recebe os comandos e os envia para o Arduino, permitindo controlar o movimento do robô remotamente

### 1. Alimentação:

O circuito começa com a alimentação fornecida por uma bateria.

O positivo da bateria está conectado a um botão, que atua como um interruptor para ligar/desligar o sistema.

Quando o botão é pressionado, a corrente passa para o VCC da ponte H, que é o componente responsável por controlar os motores.

O negativo da bateria vai direto ao GND, formando o circuito de alimentação.

### 2. Controle do Arduino:

O Arduino Nano é usado para enviar sinais de controle para a ponte H.

O VCC da ponte H também está conectado ao VIN do Arduino Nano, fornecendo energia ao microcontrolador.

O GND da ponte H é ligado ao GND do Arduino, criando uma referência de terra comum para todo o sistema.

Os pinos digitais do Arduino (D6, D7, D8, e D9) são conectados aos pinos de controle da ponte H (IN1, IN2, IN3, e IN4). Esses pinos permitem ao Arduino controlar a direção dos motores, decidindo quais motores giram e em que sentido (para frente ou para trás).

### 3. Conexão Bluetooth:

O módulo Bluetooth recebe a alimentação através do pino 5V da ponte H, garantindo que ele seja energizado para receber e enviar comandos.

O GND do módulo Bluetooth é conectado ao GND do sistema, mantendo uma referência de terra comum.

O pino RX do módulo Bluetooth foi conectado ao GND, mas essa conexão parece incomum e talvez precise ser revisada para entender sua função específica no sistema.

Para evitar danos ao Bluetooth devido a diferenças de tensão, foram usados três resistores em série como divisor de tensão. Essa configuração reduz a tensão ao nível apropriado para o Bluetooth.

O pino TX do Bluetooth está conectado ao OUT1, permitindo que os comandos recebidos do Bluetooth possam interagir com a ponte H.

#### 4. Controle dos Motores:

A ponte H possui saídas específicas para controlar os motores.

O primeiro motor está conectado aos pinos OUT1 e OUT2 da ponte H, permitindo controle completo de sua direção.

O segundo motor está conectado aos pinos OUT3 e OUT4, de forma similar, para permitir que ele gire para frente ou para trás.

## 5.2 Montagem Física do Projeto

As Figuras 10 a 13 apresentam a sequência da montagem física do nosso projeto.

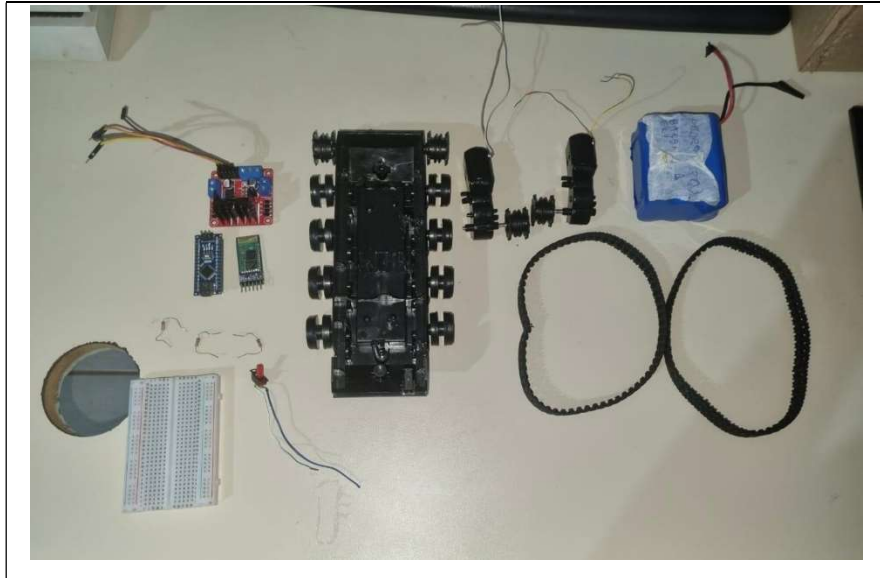


Figura 10 – Peças utilizadas no robô  
FONTE: Elaborado pelos autores (2024).



Figura 11 – Instalação das rodas e motores no chassi  
FONTE: Elaborado pelos autores (2024).

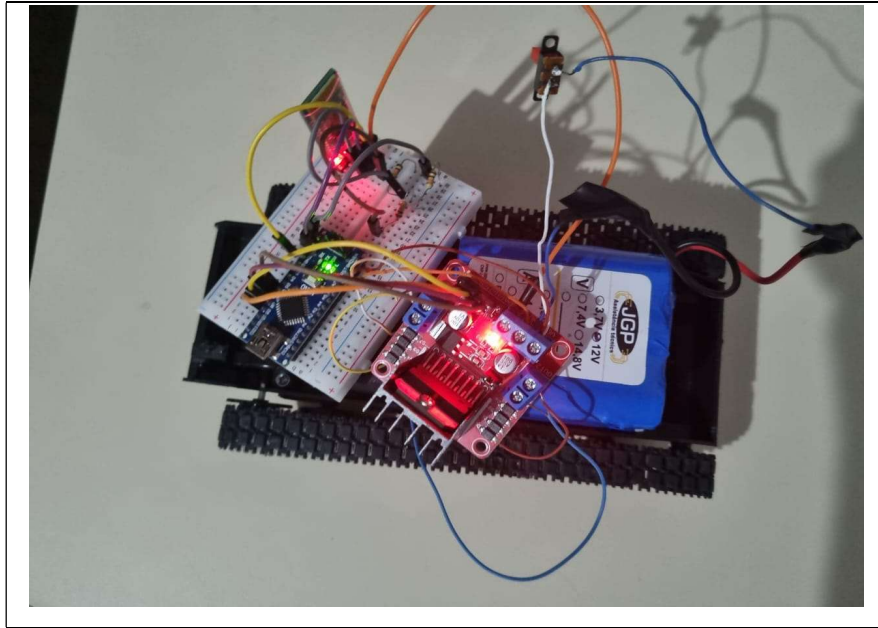


Figura 12 – Instalação do Arduino e módulos no chassi  
FONTE: Elaborado pelos autores (2024).



Figura 13 – Instalação da bateria no chassi  
FONTE: Elaborado pelos autores (2024).

As Figuras 14 e 15 apresentam os testes finais efetuados em nosso projeto.

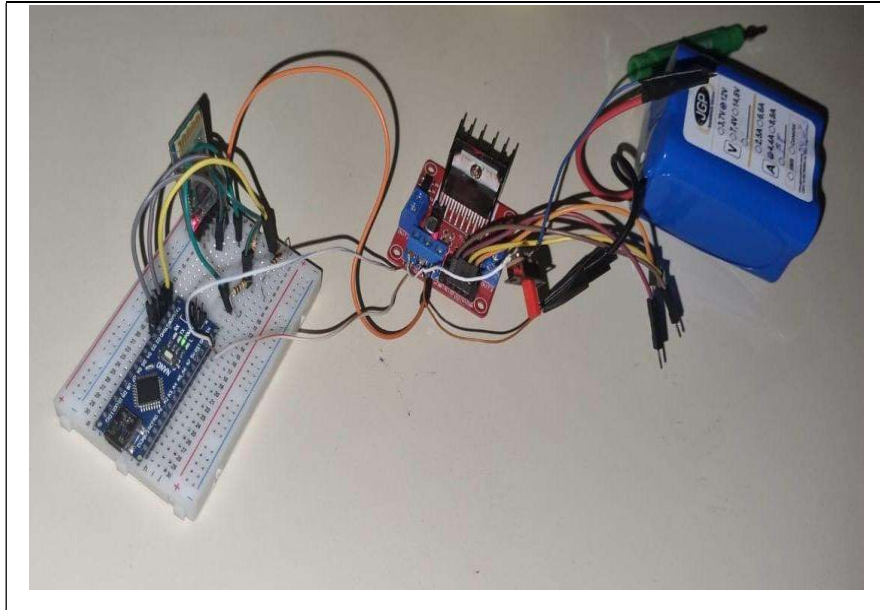


Figura 14 – Teste do hardware  
FONTE: Elaborado pelos autores (2024).

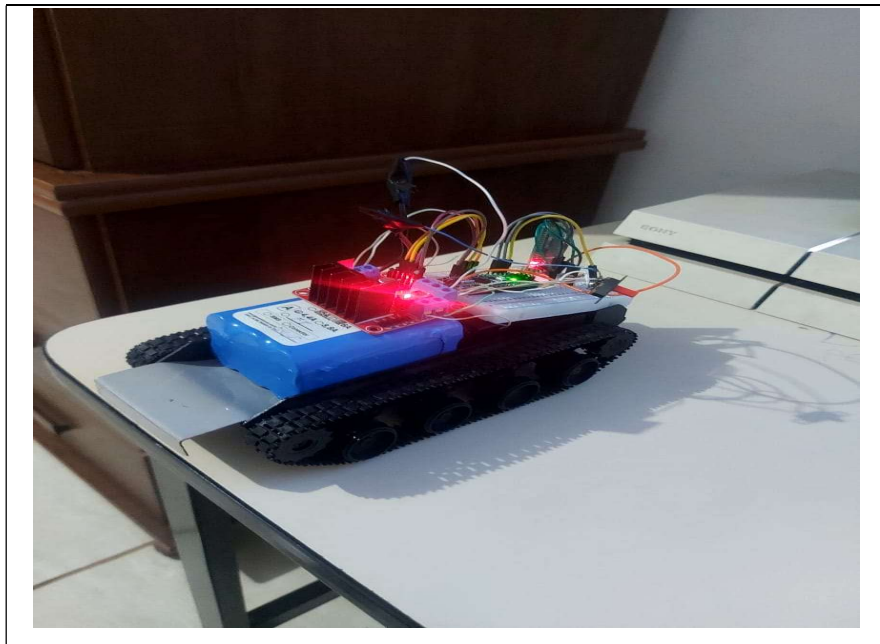


Figura 15 – Teste geral do robô  
FONTE: Elaborado pelos autores (2024).

### 5.3 Desenvolvimento do programa do robô

A programação do robô foi desenvolvida no aplicativo Arduino IDE versão 2.2.1.

A Figura 16 apresenta o fluxograma do programa.

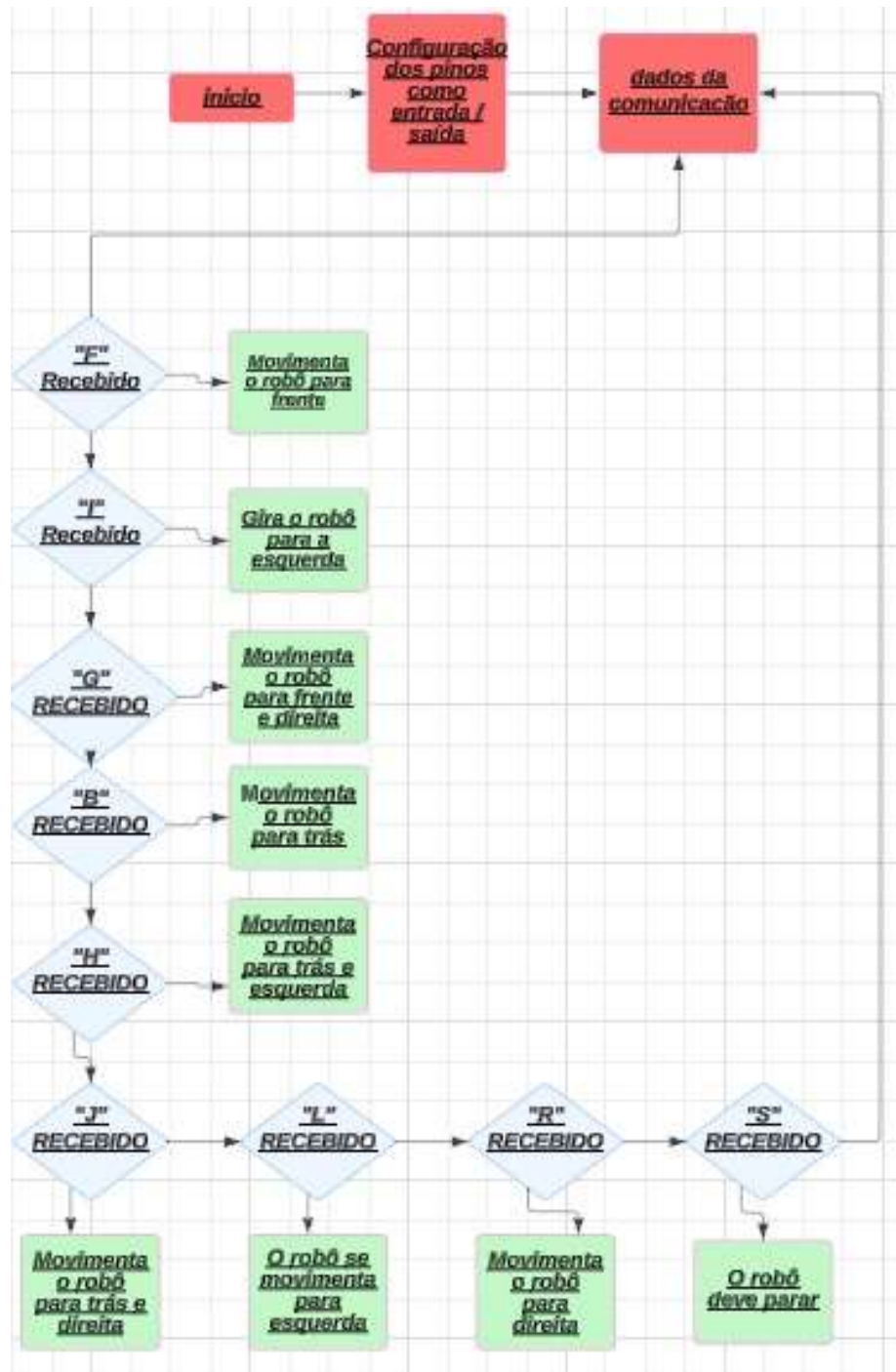


Figura 16 – Fluxograma do programa do robô  
 FONTE: Elaborado pelos autores (2024).

- Início: O programa começa e configura os pinos do robô para enviar e receber informações.
- Inicia a comunicação serial: O sistema começa a escutar comandos recebidos via comunicação serial.
- Comandos recebidos e ações:
  - Se o comando for 'F', o robô anda para frente.
  - Se o comando for 'I', o robô vira para a esquerda.
  - Se o comando for 'G', o robô anda para frente e para a direita ao mesmo tempo.
  - Se o comando for 'B', o robô anda para trás.
  - Se o comando for 'H', o robô anda para trás e para a esquerda.
  - Se o comando for 'J', o robô anda para trás e para a direita.
  - Se o comando for 'L', o robô move apenas para a esquerda.
  - Se o comando for 'R', o robô move apenas para a direita.
  - Se o comando for 'S', o robô para de se mover.

O código fonte desenvolvido no Arduino IDE é apresentado no Apêndice A.

## **6 CONSIDERAÇÕES FINAIS**

Este projeto de construção e programação de um robô sumô nos permitiu aprender e aplicar conceitos importantes de robótica, como eletrônica, mecânica e programação. Durante o processo, eles enfrentaram desafios reais que ajudaram a desenvolver habilidades técnicas e a aprender a trabalhar em equipe.

A criação do robô envolveu desde a escolha dos materiais e montagem das peças até a programação de estratégias para as competições. Houve muitos testes para garantir que tudo funcionasse corretamente. Além do aprendizado técnico, o projeto também incentivou a colaboração e a resolução de problemas práticos, habilidades que serão úteis na vida profissional.

O projeto não foi apenas sobre construir um robô para competições, mas também sobre despertar o interesse pela tecnologia e mostrar aos alunos as oportunidades de carreira na área de robótica. As habilidades adquiridas podem ser aplicadas em muitas áreas diferentes, tornando este projeto uma base importante para o futuro dos alunos no mundo da tecnologia.

## **REFERÊNCIAS BIBLIOGRÁFICAS**

- [1] **ROBOCORE. Traktor V2 Esteira.** Disponível em:  
<https://www.robocore.net/roda/esteira-traktor-v2>. Acesso em: 24 out. 2024.
- [2] **MAKERHERO. Como funciona uma protoboard?** Disponível em:  
<https://www.makehero.com/blog/como-funciona-uma-protoboard/>. Acesso em: 24 out. 2024.
- [3] **ELETROGATE. Como montar e programar um robô lutador de sumô.** Disponível em: <https://blog.eletrogate.com/como-montar-e-programar-um-robo-lutador-de-sumo/>. Acesso em: 26 out. 2024.
- [4] **INATEL. Sumô 3kg - Robótica e IA.** Disponível em:  
<https://inatel.br/robotica/categorias/sumo-3kg>. Acesso em: 26 out. 2024.
- [5] **SALÃO DE ROBÓTICA. Regras do sumô.** Disponível em:  
<https://www.salaoderobotica.org.br/competicao/regras-do-sumo/>. Acesso em: 14 nov. 2024.
- [6] **CTRL 3D. Sumô de robôs.** Disponível em: <https://sumoderobo.com.br>. Acesso em: 14 nov. 2024.
- [7] **ROBOCORE. Regulamento para competições de sumô de robôs 3kg.** Disponível em:  
[https://www.robocore.net/upload/attachments/regras\\_sumo\\_\\_ajrst\\_2019\\_pt\\_197.pdf](https://www.robocore.net/upload/attachments/regras_sumo__ajrst_2019_pt_197.pdf). Acesso em: 14 nov. 2024.
- [8] **ELECTRICITY. Como funcionam as baterias em um circuito?.** Disponível em:  
<https://www.electricity-magnetism.org/pt-br/como-funcionam-as-baterias-em-um-circuito/>. Acesso em: 14 nov. 2024.
- [9] **FVML. Módulo Bluetooth HC-05 - características e especificações.** Disponível em:  
<https://www.fvml.com.br/2023/04/modulo-bluetooth-hc-05-caracteristicas.html>. Acesso em: 14 nov. 2024.
- [10] **Zanini Renk. Moto-redutor vs redutor: diferenças e aplicações.** Disponível em:  
<https://zaninirenk.com.br/moto-redutor-vs-redutor-diferencas-e-aplicacoes>. Acesso em: 14 nov. 2024.

[11] **HWLibre. Nano Arduino.** Disponível em: <https://pt.hwlibre.com/nano-arduino/>. Acesso em: 14 nov. 2024.

[12] **Manual da Eletrônica. Ponte H: o que é? Como funciona?** Disponível em: [https://www.manualdaeletronica.com.br/ponte-h-o-que-e-como-funciona/#google\\_vignette](https://www.manualdaeletronica.com.br/ponte-h-o-que-e-como-funciona/#google_vignette). Acesso em: 14 nov. 2024.

## **APÊNDICE**

Programa do Robô

```
// CARRO CONTROLADO POR BLUETOOTH
```

```
// === Pinos utilizados no driver L298 === //
```

```
#define ENA 10    //o pino ENA do Driver será ligado ao pino 10 do arduino
(motor da esquerda)
```

```
#define IN1 9     //o pino IN1 do Driver será ligado ao pino 9 do arduino
```

```
#define IN2 8     //o pino IN2 do Driver será ligado ao pino 8 do arduino
```

```
#define IN3 7     //o pino IN3 do Driver será ligado ao pino 7 do arduino
```

```
#define IN4 6     //o pino IN4 do Driver será ligado ao pino 6 do arduino
```

```
#define ENB 5     //o pino ENB do Driver será ligado ao pino 5 do arduino
(motor da direita)
```

```
// ==== Variáveis utilizadas ==== //
```

```
char Texto_Recebido; //variável do tipo caracter (char) - recebe os dados do
bluetooth/comunicação serial
```

```
byte velocidade = 70; //variável do tipo byte - usada no controle da
velocidade
```

```
void setup() {
```

```
// === Definição dos pinos de saída === //
```

```
pinMode(ENA, OUTPUT); //configura o pino do ENA como saída
```

```
pinMode(IN1, OUTPUT); //configura o pino do IN1 como saída
```

```
pinMode(IN2, OUTPUT); //configura o pino do IN2 como saída
```

```
pinMode(IN3, OUTPUT); //configura o pino do IN3 como saída
```

```
pinMode(IN4, OUTPUT); //configura o pino do IN4 como saída
```

```
pinMode(ENB, OUTPUT); //configura o pino do ENA como saída
```

```
pinMode(13, OUTPUT); //configura o pino do 13 (ligado ao led L)
como saída
```

```
Serial.begin(9600); //Inicia a comunicação serial
```

```
}
```

```

void loop() {
// === Dados recebidos na comunicação === //
  //if (Serial.available() == 0) {          //"SE" não tiver dados disponíveis na
  comunicação, faça...
    // Texto_Recebido = 'S';                //Situação necessária para parar o carro
    (incluindo perda de sinal)
  // }

  if (Serial.available() > 0) {            //"SE" tiver dados disponíveis na
  comunicação, faça...
    Texto_Recebido = Serial.read();        //o valor da leitura serial é atribuído
    à variável Texto_Recebido
  }

// === Movimentação do carro === //
  if (Texto_Recebido == 'F') {             //"SE" a letra recebida for "F", faça...
    Serial.println("Para Frente");         //imprime o texto "Para Frente" no
    monitor serial
    analogWrite (ENA, velocidade);         //envia ao pino ENA o valor da
    velocidade do motor 1
    analogWrite (ENB, velocidade);         //envia ao pino ENB o valor da
    velocidade do motor 2
    digitalWrite(IN1, 1);                  //envia nível lógico alto para o pino
    IN1 //liga o motor 1
    digitalWrite(IN2, 0);                  //envia nível lógico baixo para o pino
    IN2 //para frente
    digitalWrite(IN3, 1);                  //envia nível lógico alto para o pino
    IN3 **liga o motor 2
    digitalWrite(IN4, 0);                  //envia nível lógico baixo para o pino IN4
    **para frente
    digitalWrite(13,1);                    //liga o led L ligado ao pino 13
  }
  else if (Texto_Recebido == 'I') {        //"MAS SE" a letra recebida for "I",
  faça...
    Serial.println("Para Frente e Esquerda"); //imprime o texto "Para Frente e

```

Esquerda" no monitor serial

```

    analogWrite (ENA, velocidade);      //envia ao pino ENA o valor da
velocidade do motor 1
    analogWrite (ENB, velocidade-150);  //envia ao pino ENB o valor da
velocidade-150 do motor 2
    digitalWrite(IN1, 1);                //envia nível lógico alto para o pino
IN1  //liga o motor 1
    digitalWrite(IN2, 0);                //envia nível lógico baixo para o pino
IN2  //para frente
    digitalWrite(IN3, 1);                //envia nível lógico alto para o pino
IN3  **liga o motor 2
    digitalWrite(IN4, 0);                //envia nível lógico baixo para o pino IN4
**para frente
}

```

else if (Texto\_Recebido == 'G') { // "SE" a letra recebida for igual a 'G', o carro se movimenta para Frente Direita.

Serial.println("Para Frente e Direita"); //imprime o texto "Para Frente e Direita" no monitor serial

```

    analogWrite (ENA, velocidade-150);  //envia ao pino ENA o valor da
velocidade-150 do motor 1
    analogWrite (ENB, velocidade);      //envia ao pino ENB o valor da
velocidade do motor 2
    digitalWrite(IN1, 1);                //envia nível lógico alto para o pino
IN1  //liga o motor 1
    digitalWrite(IN2, 0);                //envia nível lógico baixo para o pino
IN2  //para frente
    digitalWrite(IN3, 1);                //envia nível lógico alto para o pino
IN3  **liga o motor 2
    digitalWrite(IN4, 0);                //envia nível lógico baixo para o pino IN4
**para frente
}

```

else if (Texto\_Recebido == 'B') { // "SE" a letra recebida for igual a 'B', o carro se movimenta para Trás.

Serial.println("Para Trás"); //imprime o texto "Para Trás" no monitor

```

serial
    analogWrite (ENA, velocidade);      //envia ao pino ENA o valor da
velocidade do motor 1
    analogWrite (ENB, velocidade);      //envia ao pino ENB o valor da
velocidade do motor 2
    digitalWrite(IN1, 0);                //envia nível lógico baixo para o pino
IN1 //liga o motor 1
    digitalWrite(IN2, 1);                //envia nível lógico alto para o pino
IN2 //para trás
    digitalWrite(IN3, 0);                //envia nível lógico baixo para o pino
IN3 **liga o motor 2
    digitalWrite(IN4, 1);                //envia nível lógico alto para o pino
IN4 **para trás
}

else if (Texto_Recebido == 'H') {        //"SE" a letra recebida for igual a
'H', o carro se movimenta para Trás e esquerda.
    Serial.println("Para Trás e Esquerda"); //imprime o texto "Para Trás e
Esquerda" no monitor serial
    analogWrite (ENA, velocidade-150);   //envia ao pino ENA o valor da
velocidade-150 do motor 1
    analogWrite (ENB, velocidade);       //envia ao pino ENB o valor da
velocidade do motor 2
    digitalWrite(IN1, 0);                //envia nível lógico baixo para o pino
IN1 //liga o motor 1
    digitalWrite(IN2, 1);                //envia nível lógico alto para o pino
IN2 //para trás
    digitalWrite(IN3, 0);                //envia nível lógico baixo para o pino
IN3 **liga o motor 2
    digitalWrite(IN4, 1);                //envia nível lógico alto para o pino
IN4 **para trás
}

else if (Texto_Recebido == 'J') {        //"SE" a letra recebida for igual a
'J', o carro se movimenta para Trás e direita.
    Serial.println("Para Trás e Direita"); //imprime o texto "Para Trás e

```

```

Direita" no monitor serial
    analogWrite (ENA, velocidade);      //envia ao pino ENA o valor da
velocidade do motor 1
    analogWrite (ENB, velocidade-150);  //envia ao pino ENB o valor da
velocidade-150 do motor 2
    digitalWrite(IN1, 0);                //envia nível lógico baixo para o pino
IN1 //liga o motor 1
    digitalWrite(IN2, 1);                //envia nível lógico alto para o pino
IN2 //para trás
    digitalWrite(IN3, 0);                //envia nível lógico baixo para o pino
IN3 **liga o motor 2
    digitalWrite(IN4, 1);                //envia nível lógico alto para o pino
IN4 **para trás
}

else if (Texto_Recebido == 'L') {        //"SE" a letra recebida for igual a
'L', o carro se movimenta para a esquerda.
    Serial.println("Para Esquerda");     //imprime o texto "Para Esquerda" no
monitor serial
    analogWrite (ENA, velocidade);      //envia ao pino ENA o valor da
velocidade do motor 1
    analogWrite (ENB, velocidade);      //envia ao pino ENB o valor da
velocidade do motor 2
    digitalWrite(IN1, 0);                //envia nível lógico baixo para o pino
IN1 //liga o motor 1
    digitalWrite(IN2, 1);                //envia nível lógico alto para o pino
IN2 //para trás
    digitalWrite(IN3, 1);                //envia nível lógico alto para o pino
IN3 **liga o motor 2
    digitalWrite(IN4, 0);                //envia nível lógico baixo para o pino IN4
**para frente
}

else if (Texto_Recebido == 'R') {        //"SE" a letra recebida for igual a
'R', o carro se movimenta para a direita.
    Serial.println("Para Direita");      //imprime o texto "Para Direita" no

```

```

monitor serial
  analogWrite (ENA, velocidade);      //envia ao pino ENA o valor da
velocidade do motor 1
  analogWrite (ENB, velocidade);      //envia ao pino ENB o valor da
velocidade do motor 2
  digitalWrite(IN1, 1);                //envia nível lógico alto para o pino
IN1  //liga o motor 1
  digitalWrite(IN2, 0);                //envia nível lógico baixo para o pino
IN2  //para frente
  digitalWrite(IN3, 0);                //envia nível lógico baixo para o pino
IN3  **liga o motor 2
  digitalWrite(IN4, 1);                //envia nível lógico alto para o pino
IN4  **para trás
}

else if (Texto_Recebido == 'S') {      //"SE" a letra recebida for igual a
'S', o carro deve parar.
  Serial.println("Parado");            //imprime o texto "Parado" no monitor
serial
  digitalWrite(IN1, 0);                //envia nível lógico baixo para o pino
IN1  //desliga
  digitalWrite(IN2, 0);                //envia nível lógico baixo para o pino
IN2  //o motor 1
  digitalWrite(IN3, 0);                //envia nível lógico baixo para o pino
IN3  **desliga
  digitalWrite(IN4, 0);                //envia nível lógico baixo para o pino
IN4  **o motor 2
  digitalWrite(13,0);                  //desliga o led L ligado ao pino 13
}
}

```

