

CENTRO UNIVERSITÁRIO FUNDAÇÃO SANTO ANDRÉ
FACULDADE DE ENGENHARIA ENGENHEIRO CELSO DANIEL

MICROCONTROLADORES II

COSSA. André, RA: 752337; RUBERTONE. Gabriel, RA: 747252; BARROS. Maria
Eduarda, RA: 746820; DOMINGOS. Renatha, RA: 752542.

ROBÔ DE SUMÔ - OVERTORQ

**Santo André
2025**

COSSA. André, RA: 752337; RUBERTONE. Gabriel, RA: 747252; BARROS. Maria Eduarda, RA: 746820; DOMINGOS. Renatha, RA: 752542.

ROBÔ DE SUMÔ - OVERTORQ

Relatório apresentado à disciplina de Microcontroladores II do Centro Universitário Fundação Santo André.

Coordenador: Prof. Edson Antonio Abreu.

**Santo André
2025**

SUMÁRIO

1. INTRODUÇÃO	4
2. OBJETIVO	4
3. MATERIAIS E MÉTODOS	5 a 9
4. ANÁLISE E RESULTADOS	10 e 11
5. CONCLUSÃO	11
6. DESENHO TÉCNICO DO ROBÔ	12 a 19
7. CÓDIGO C++ DO ROBÔ	20 a 26

1. INTRODUÇÃO

O robô de sumô de 3 kg trata-se de um projeto inspirado no tradicional esporte japonês de luta livre *Sumô*, adaptado para o campo da robótica educacional e competitiva. Nessa modalidade, dois robôs se enfrentam em uma arena circular – o *Dohyo* – com o objetivo de empurrar o oponente para fora dos limites do ringue.

Para que seja possível o desenvolvimento do robô, torna-se necessário a aplicação de conceitos de eletrônica, mecânica e programação, proporcionando assim uma integração prática entre teorias aplicadas em sala de aula e a vivência na prática sobre os conceitos abordados. Apresentando com sua massa máxima de 3 kg e dimensões de até 20cm x 20cm, o robô apresenta o seu controle por meio do microcontrolador ESP32, que se comunica via Bluetooth ou wi-fi com um celular, permitindo assim total controle durante a competição, visando também a segurança do público. Torna-se necessário também a especificação adequada de bateria, garantindo autonomia e eficiência durante a competição.

Desta forma, nota-se que a execução deste projeto representa mais do que uma competição entre os alunos, torna-se uma oportunidade de aprendizado prático e multidisciplinar, unindo a engenharia, tecnologia e o trabalho em grupo, estimulando assim o raciocínio lógico, a criatividade e a capacidade de solucionar problemas reais por meio da automação e o controle inteligente de sistemas, promovendo o interesse pela inovação tecnológica e novos desafios voltados para a área da robótica e engenharia.

2. OBJETIVO

Desenvolver um robô de sumô controlado via Bluetooth ou wi-fi com massa máxima de 3kg e dimensões 20cm x 20cm, capaz de enfrentar o seu oponente em um *Dohyo* empurrando-o para fora dos limites da arena, bem como, relacionar a teoria ensinada em sala de aula com a prática, promovendo aplicação e conhecimento em eletrônica, mecânica e programação visando o uso do ESP32 em sua comunicação Bluetooth ou wi-fi.

3. MATERIAIS E MÉTODOS

O desenvolvimento do robô de sumô seguiu uma metodologia estruturada em etapas, abrangendo desde o desenho do projeto até a montagem e testes funcionais. Para a construção de um robô competitivo, resistente e de alto desempenho respeitando os limites de massa (3 kg) e dimensões (20cm x 20cm) foram empregados os seguintes componentes:

- 1x ESP 32;
- 1x Cola Epóxi Araldite;
- 40x Cabos Jumper para ESP32;
- 2x Fita Isolante Auto Fusão 2 metros;
- 1x Par de motor de vidro elétrico;
- 1x Bateria selada de chumbo 5ah;
- 1x Regulador de tensão DCDC LM2596;
- 2x Driver Ponte-H BTS7960;
- 2x Rolamento 608zz;
- 2x Rodas de madeira 100mm∅X20mmL;
- Tarugo de metal usinado para a criação do eixo;
- Fios de cobre para ligações dos motores;
- Estrutura impressa em 3D.

Partindo para o desenvolvimento do robô, as seguintes etapas foram desenvolvidas:

- **Modelagem 3D:** Elaboração de um modelo tridimensional do robô, permitindo uma boa visualização de sua carcaça;
- **Fabricação da estrutura:** Estrutura impressa em 3D (desenho modelado convertido em .stl) dimensionados para resistir aos impactos. Após a impressão

das peças, estas foram unidas e reforçadas com a cola Epóxi garantindo assim, mais rigidez e durabilidade;

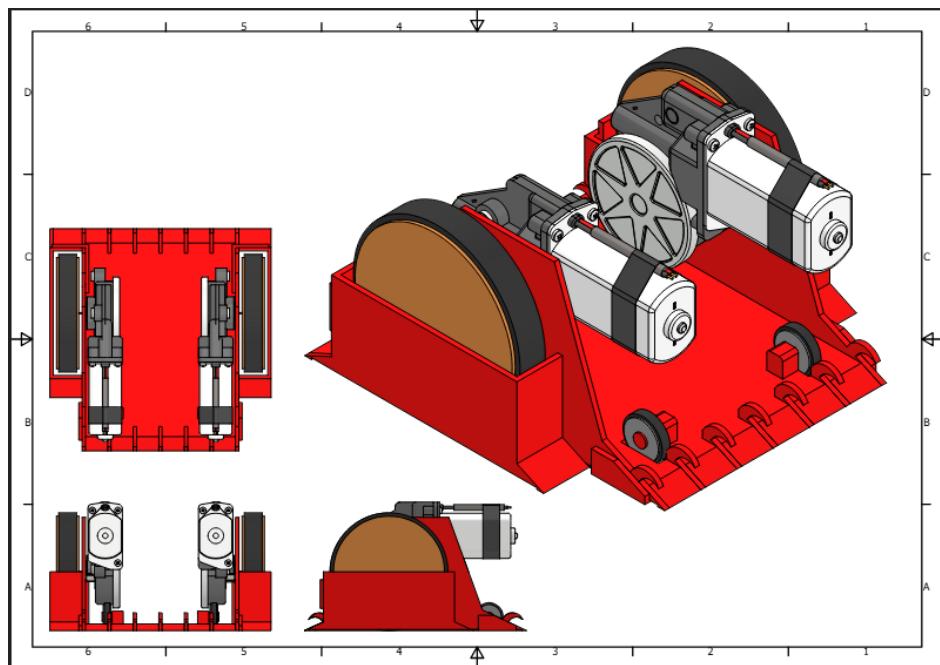


Figura 1 Desenho do robô

- **Programação e controle:** O código para o controle do robô de sumô foi desenvolvido na linguagem C++ na plataforma Arduino IDE utilizando comunicação Bluetooth, sendo testadas funções de direção, aceleração e frenagem, sendo possível realizar ajustes na programação para um melhor desempenho;
- **Montagem mecânica:** O eixo original do motor de 8 dentes foi substituído por um novo eixo desenvolvido na plataforma Inventor, e então foi realizado a usinagem do tarugo de metal para a criação do eixo do motor, logo após, a roda foi parafusada diretamente no eixo usinado. Para garantir uma melhor aderência nas rodas, foram adicionadas 4 voltas de fita isolante de Auto Fusão, gerando uma maior aderência das rodinhas ao solo;



Figura 2 Eixo usinado

- **Montagem final:** Após a realização de todas as etapas descritas anteriormente, foi realizada a montagem do robô, com montagem da carcaça e a junção com todos os componentes que fazem parte do robô, bem como, a instalação do sistema eletrônico, onde foram utilizados cabos jumper e fios de cobre para realizar as conexões.



Figura 3 Estrutura

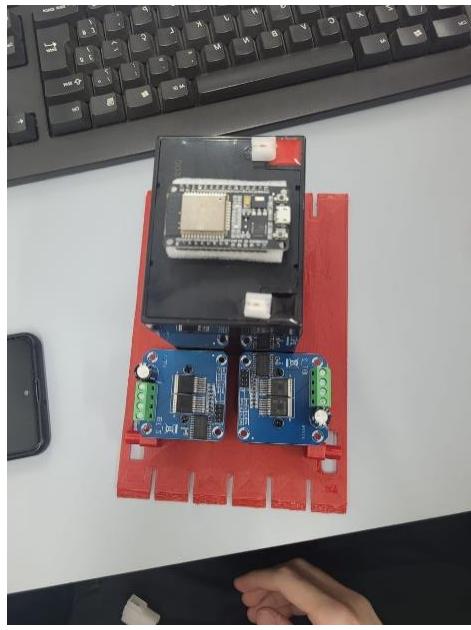


Figura 4 Simulação da montagem

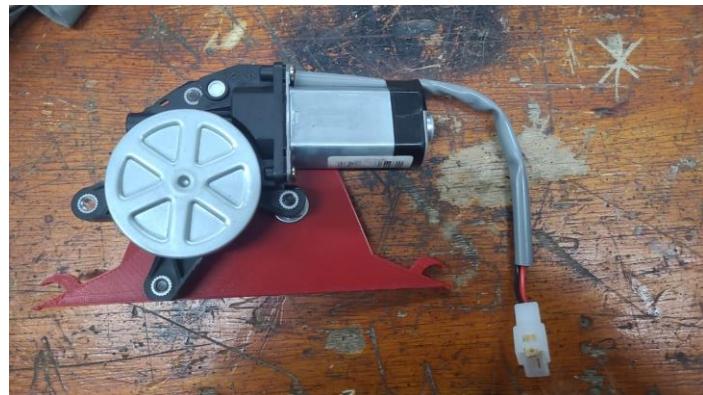


Figura 5 Motor acoplado na estrutura



Figura 6 Motor acoplado na estrutura

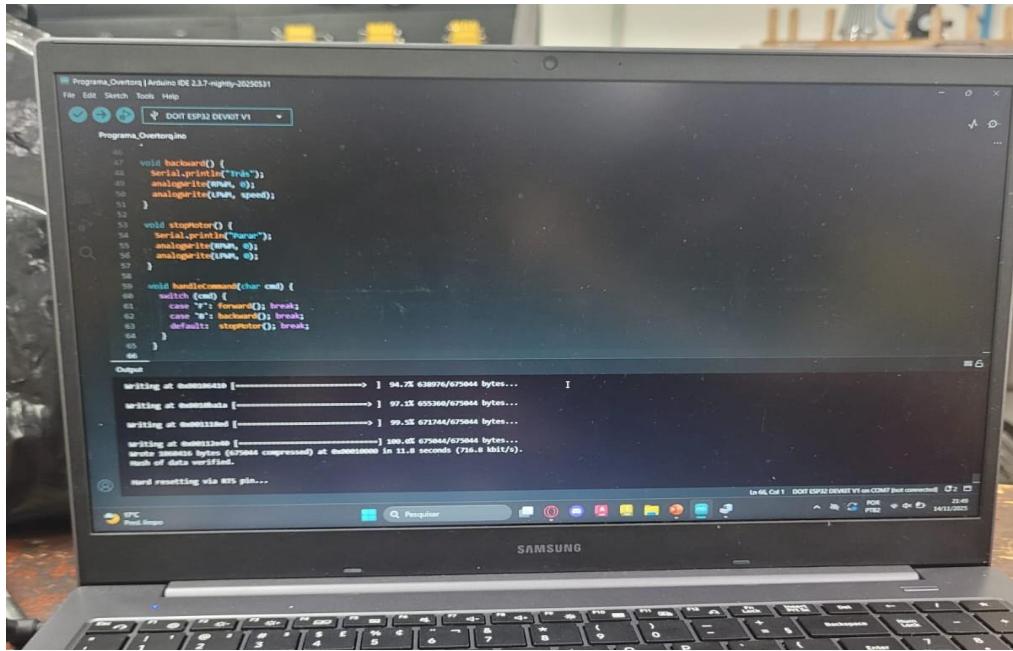


Figura 7 Código

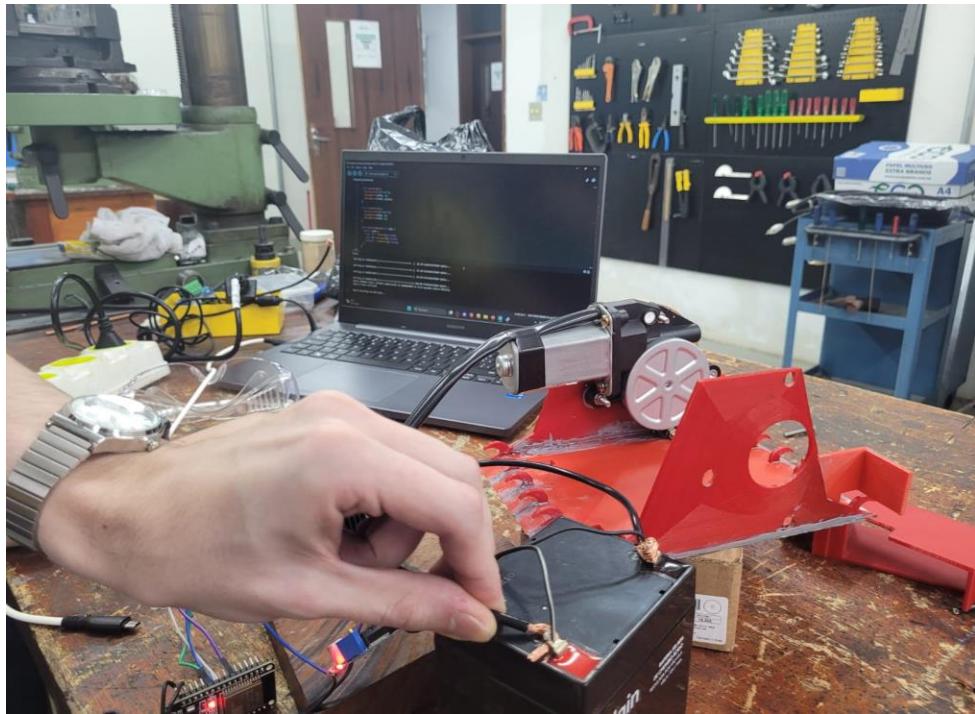


Figura 8 Teste código

4. ANÁLISE E RESULTADOS

Durante o desenvolvimento do robô, foi possível analisar diversos aspectos técnicos e funcionais que influenciam diretamente no desempenho do sistema. A etapa inicial do projeto concentrou-se no debate e elaboração de um modelo de robô, o qual, após análises dimensionais, foi elaborado um modelo tridimensional 3D por meio do software Autodesk Inventor, permitindo uma visualização de grande precisão do modelo desenvolvido pelo grupo, auxiliando em encaixe das peças e facilitando os ajustes desenvolvidos no design visando uma melhor estabilidade e equilíbrio durante as batalhas. No decorrer do desenvolvimento, foi realizado o dimensionamento dos materiais utilizados no robô, bem como, qual seria o material utilizado para criar a sua carcaça, esta realizada em impressora 3D oferecendo leveza, resistência e precisão.

A escolha dos motores, foi desenvolvida de maneira estratégica onde se optou na escolha de motores de maior torque, garantindo assim maior força de empurrão contra o robô adversário, mesmo que resultasse em uma redução de velocidade, gerando um aumento no desempenho do robô, tornando-o mais resistente e competitivo. O driver BTS7960, foi o responsável pelo controle eficiente dos motores da direção e velocidade, garantindo uma resposta rápida aos comandos, enquanto o sistema de alimentação pela bateria selada de chumbo 12V 5ah gerou boa autonomia e estabilidade de tensão durante os testes, atendendo os requisitos de potência dos motores e do circuito de potência.

Tendo em vista que a tensão de alimentação do microcontrolador ESP32 é de 5V, tornou-se necessário o uso de regulador de tensão DCDC LM2596, responsável em receber os 12V da bateria baixando para 5V alimentando o microcontrolador, responsável em realizar os movimentos implantados em linguagem C++ no robô por meio da comunicação Bluetooth com o celular, garantindo respostas rápidas e precisas durante a operação.

A seleção das rodinhas proporcionou uma excelente tração e aderência à superfície da arena, sendo estas feitas de madeira com dimensões 100mmØX20mmL e revestidas com fita isolante de Auto Fusão. Bem como, para redução do atrito e melhoria na mobilidade, foi utilizado rolamento 608zz na parte da frente do robô.

De modo geral, o projeto atingiu os objetivos propostos, demonstrando uma ótima integração entre o design, eletrônica e controle, contribuindo para um bom desempenho do robô.

5. CONCLUSÃO

O desenvolvimento do robô de sumo proporcionou uma experiência interdisciplinar, podendo realizar a integração entre as áreas da mecânica, eletrônica, automação e design, consolidando conhecimentos teóricos por meio da prática. Desde o projeto tridimensional até a montagem física e os testes durante o processo, foi possível desenvolver e criar um sistema funcional e competitivo. O uso da impressão 3D na fabricação da estrutura da carcaça permitiu criar uma estrutura leve e resistente, garantindo uma precisão no encaixe dos componentes.

A escolha dos motores aliado as baterias, resultou na força desejada pelo grupo para empurrar o adversário, visto que o motor apresentou um grande torque. Bem como, a integração eletrônica comandada pelo ESP32 assegurou um controle preciso via Bluetooth, com respostas rápidas aos comandos do operador. O processo de usinagem do eixo e o acoplamento das rodas de madeira, foi uma etapa fundamental para gerar uma eficiência mecânica, garantindo melhor transmissão de torque e estabilidade nas manobras, bem como, o uso da fita isolante colaborou com a aderência do robô ao solo.

Em síntese, o projeto atingiu as expectativas determinadas pelo grupo, demonstrando um robô de sumô equilibrado e com uma boa performance, representando a aplicação direta dos conceitos da engenharia, inovação tecnológica no campo da robótica competitiva bem como, o desenvolvimento do trabalho em grupo.

6. DESENHO TÉCNICO DO ROBÔ

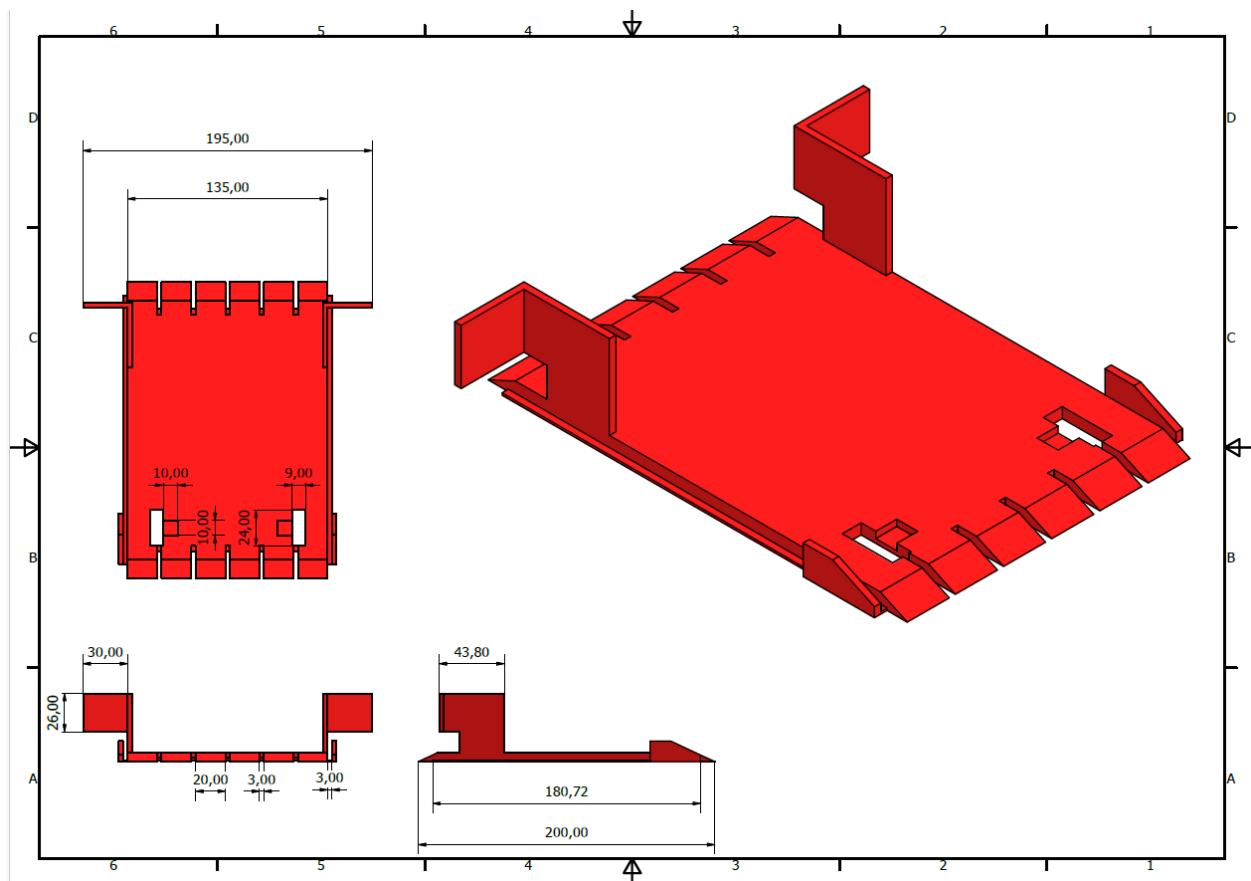


Figura 3 Estrutura 1

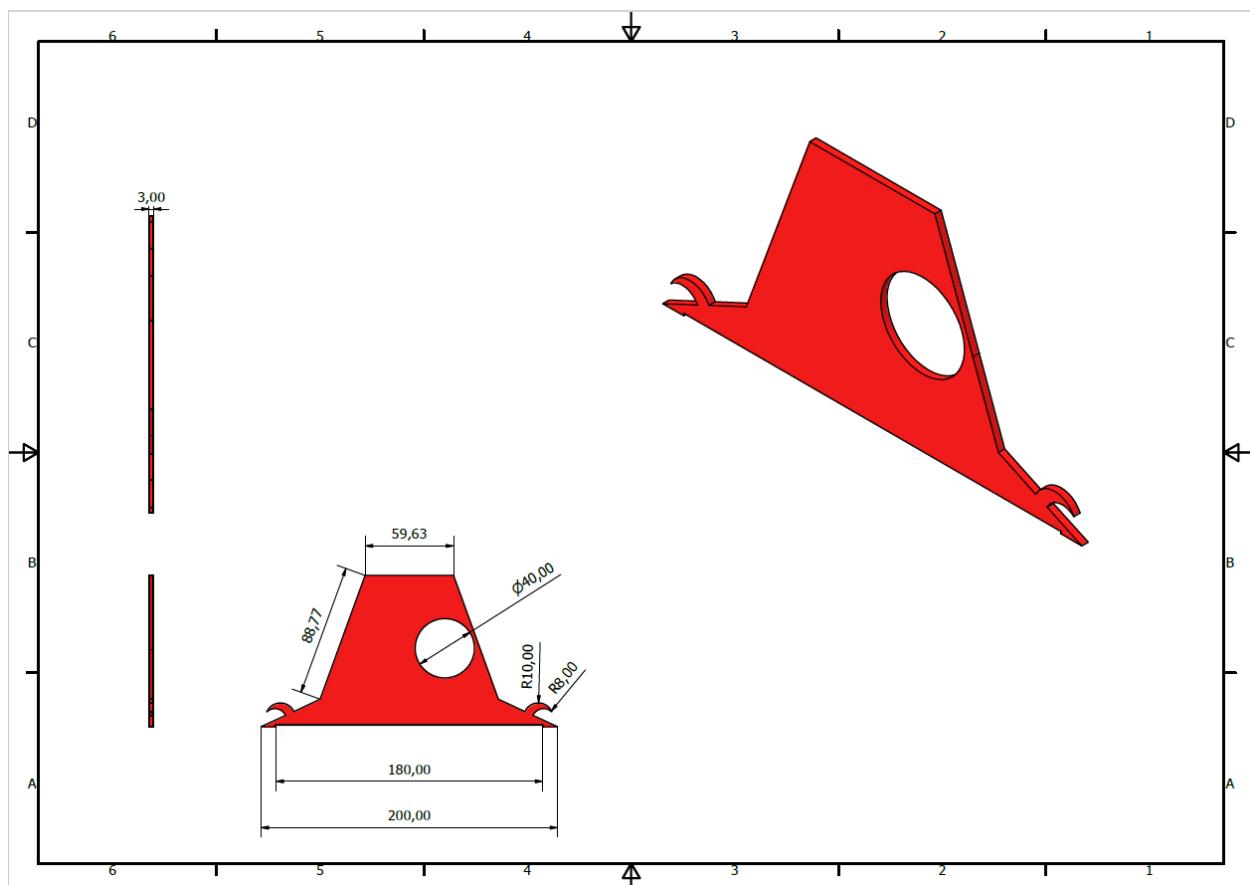


Figura 4 Estrutura 1.1

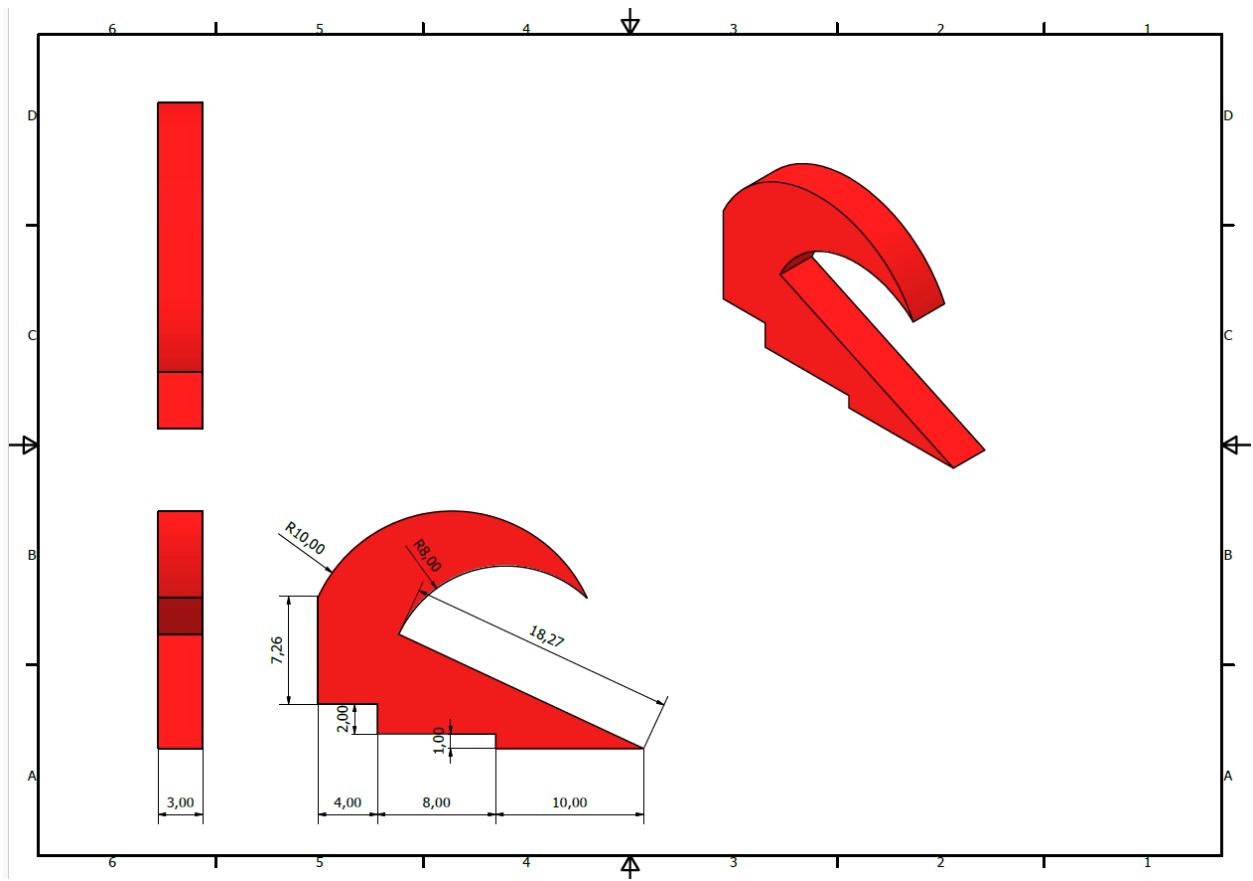


Figura 5 Estrutura 1.2

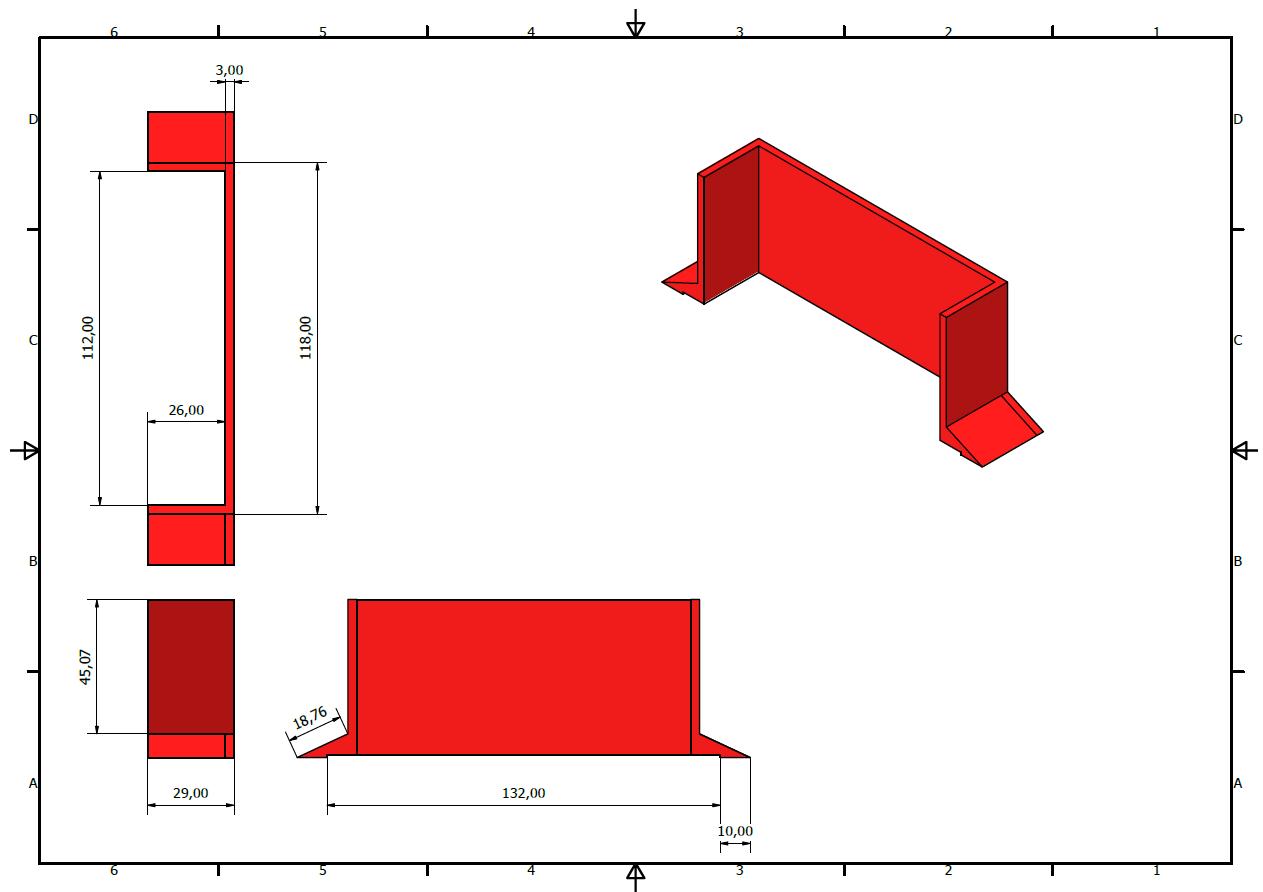


Figura 6 Estrutura 1.3

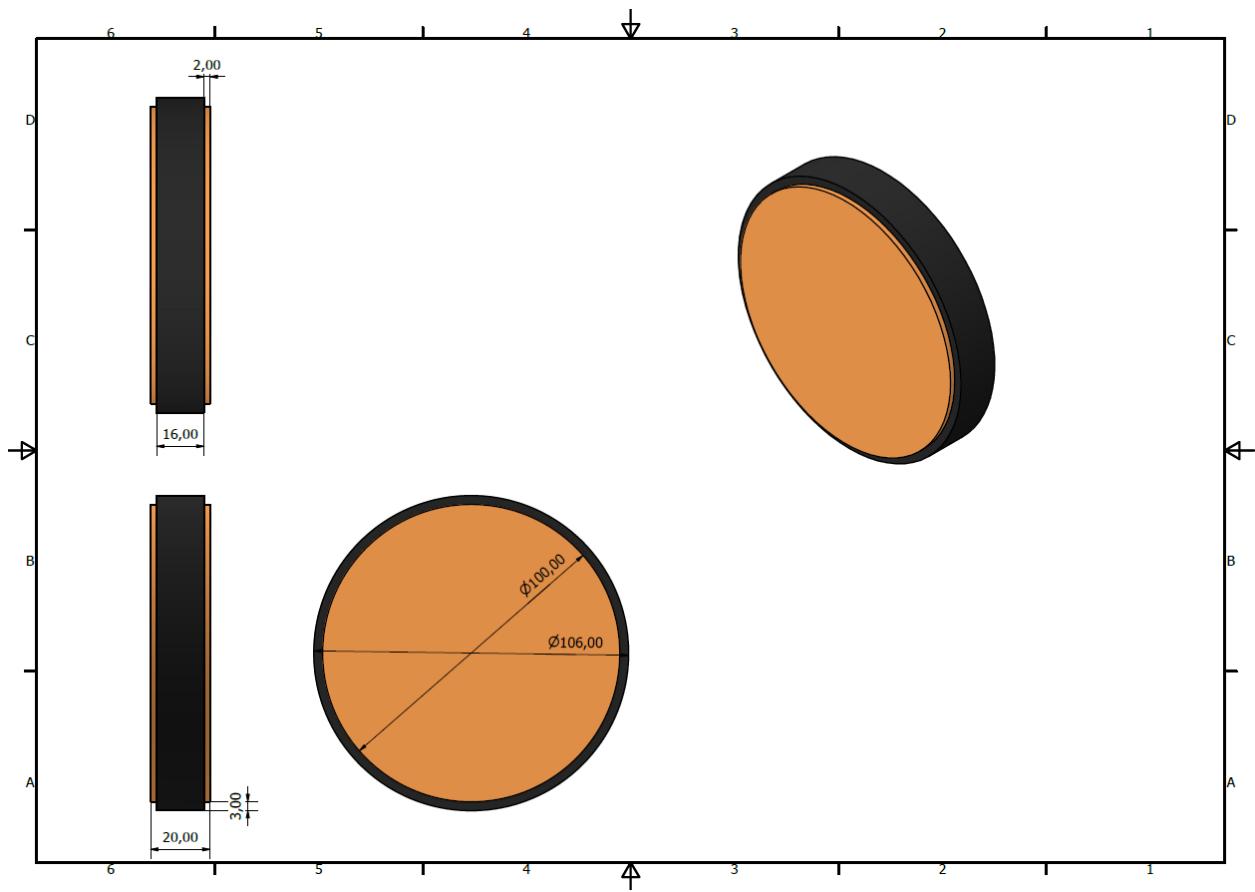


Figura 7 Roda

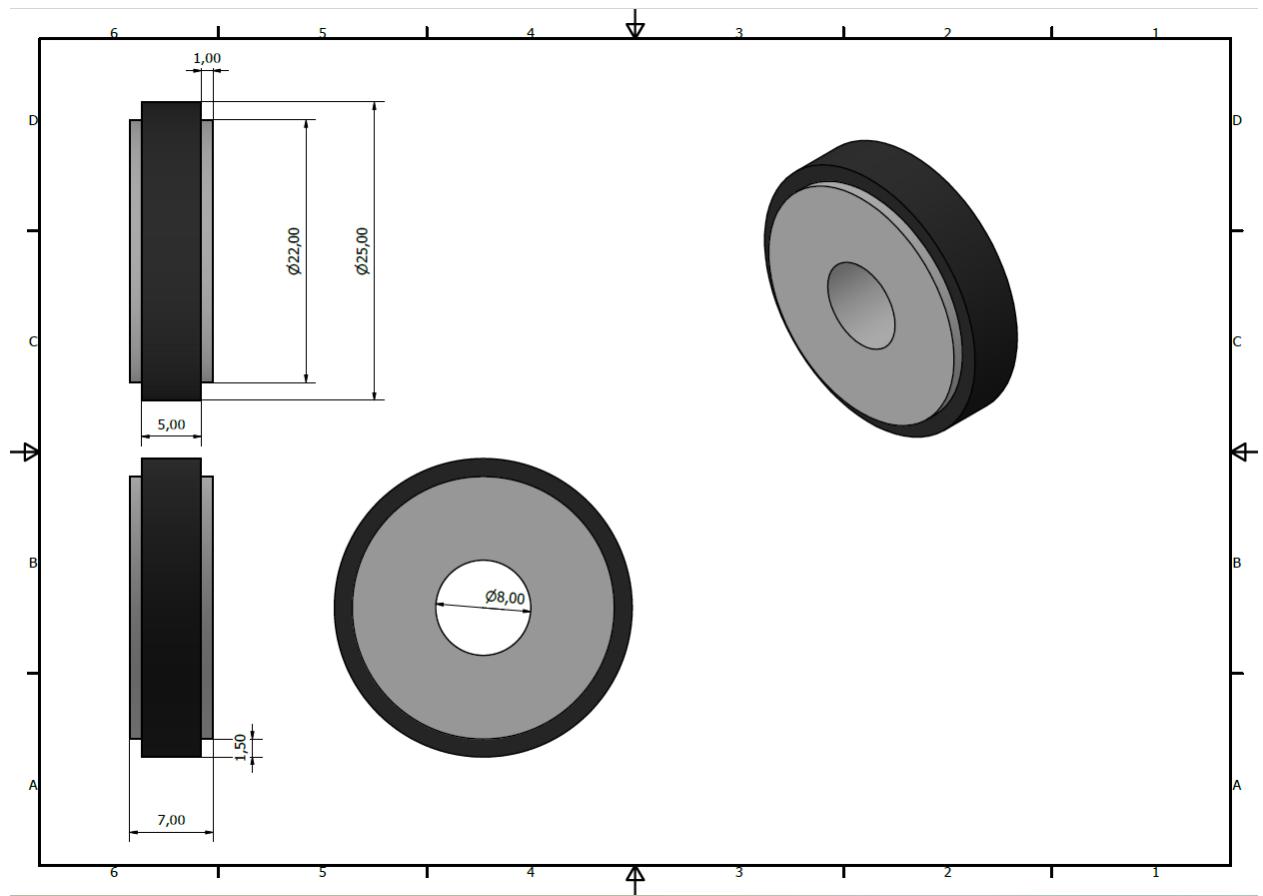


Figura 8 Rolamento

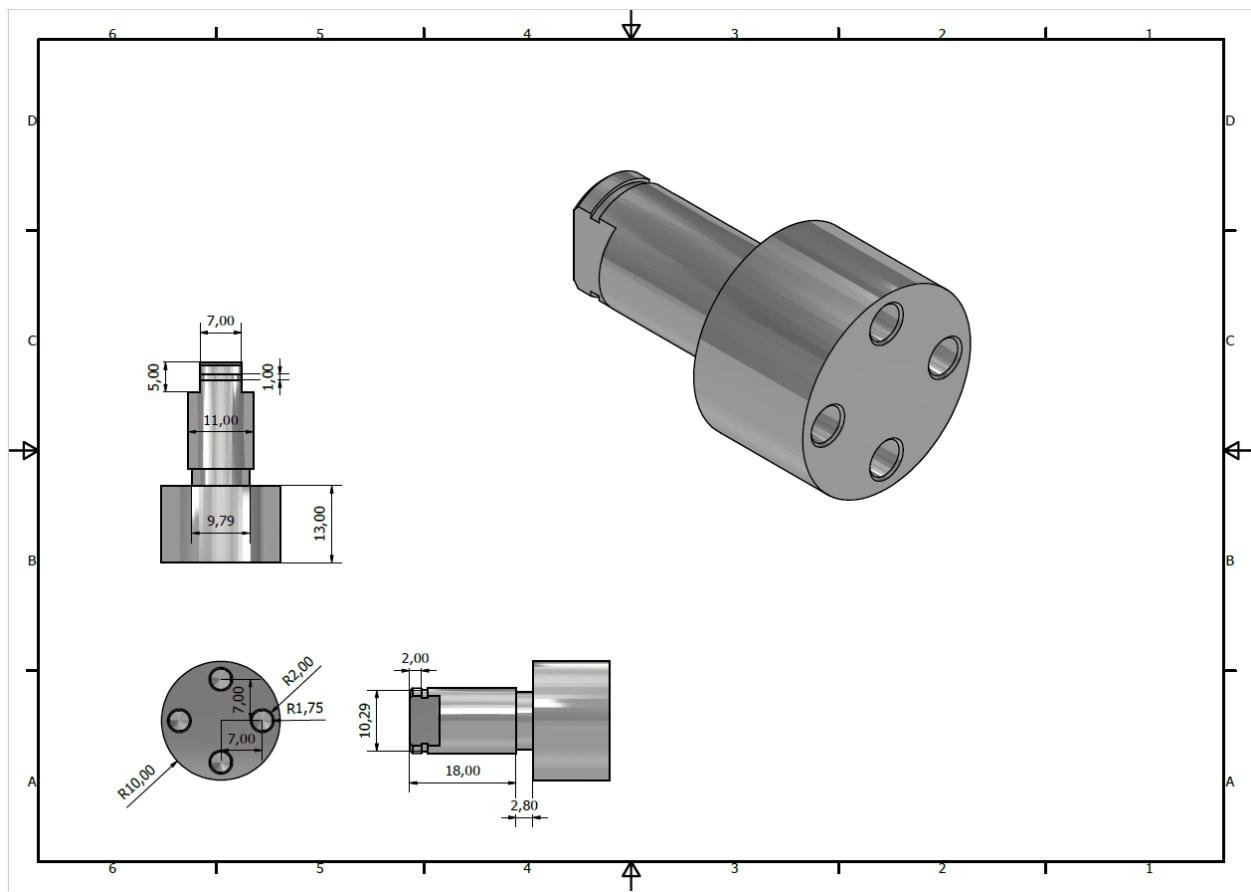


Figura 9 Eixo

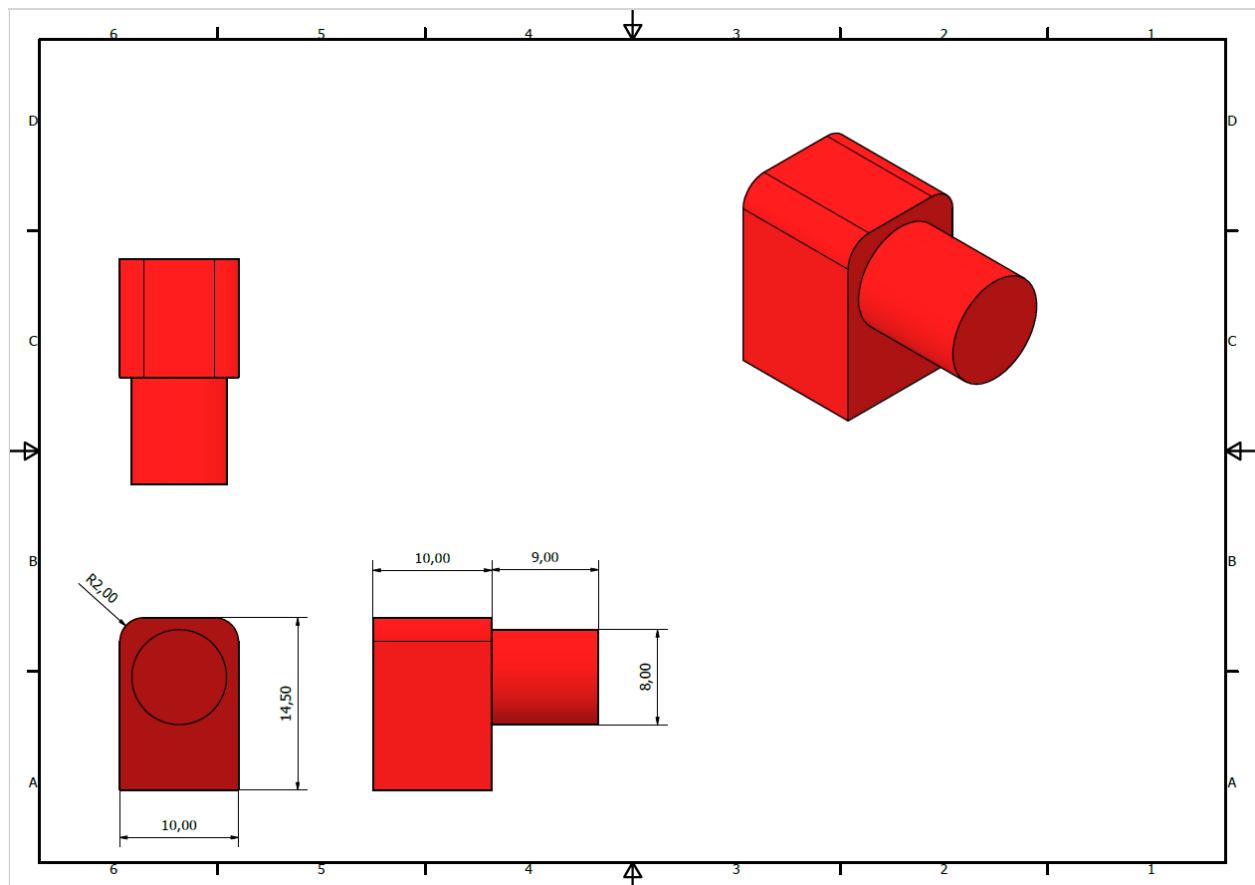


Figura 10 suporte do rolamento

7. CÓDIGO C++ DO ROBÔ

```
// ----- MOTOR A (Direito) -----  
  
#define A_RPWM 25  
  
#define A_LPWM 26  
  
#define A_REN 18  
  
#define A_LEN 5  
  
  
// ----- MOTOR B (Esquerdo) -----  
  
#define B_RPWM 27  
  
#define B_LPWM 22  
  
#define B_REN 32  
  
#define B_LEN 33  
  
  
// Bluetooth  
  
#include "BluetoothSerial.h"  
  
BluetoothSerial SerialBT;  
  
  
char comando = 'S';  
  
  
void setup() {  
    SerialBT.begin("Overtorq");  
  
    pinMode(A_RPWM, OUTPUT);  
    pinMode(A_LPWM, OUTPUT);
```

```
pinMode(A_REN, OUTPUT);
pinMode(A_LEN, OUTPUT);

pinMode(B_RPWM, OUTPUT);
pinMode(B_LPWM, OUTPUT);
pinMode(B_REN, OUTPUT);
pinMode(B_LEN, OUTPUT);

digitalWrite(A_REN, HIGH);
digitalWrite(A_LEN, HIGH);
digitalWrite(B_REN, HIGH);
digitalWrite(B_LEN, HIGH);

pararMotores();
}
```

```
void loop() {
    if (SerialBT.available()) {
        comando = SerialBT.read();
    }
}
```

```
switch (comando) {
    case 'F': // Frente
```

```
frenteMotores();  
break;  
  
case 'B': // Trás  
trasMotores();  
break;  
  
case 'L': // Esquerda  
virarEsquerda();  
break;  
  
case 'R': // Direita  
virarDireita();  
break;  
  
case 'l': // Esquerda trás  
virarEsquerdaTras();  
break;  
  
case 'd': // Direita trás  
virarDireitaTras();  
break;  
  
case 'w': // Comemoração
```

```
comemorar();  
break;  
  
case 'S': // Solto o botão → PARA TUDO  
default:  
    pararMotores();  
    break;  
}  
}
```

```
void frenteMotores() {  
    // A para frente  
    digitalWrite(A_RPWM, HIGH);  
    digitalWrite(A_LPWM, LOW);  
  
    // B para frente  
    digitalWrite(B_RPWM, HIGH);  
    digitalWrite(B_LPWM, LOW);  
}
```

```
void trasMotores() {  
    // A trás  
    digitalWrite(A_RPWM, LOW);  
    digitalWrite(A_LPWM, HIGH);
```

```
// B trás  
  
digitalWrite(B_RPWM, LOW);  
  
digitalWrite(B_LPWM, HIGH);  
  
}
```

```
void virarEsquerda() {  
  
// A frente (direito roda)  
  
digitalWrite(A_RPWM, HIGH);  
  
digitalWrite(A_LPWM, LOW);  
  
  
// B parado  
  
digitalWrite(B_RPWM, LOW);  
  
digitalWrite(B_LPWM, LOW);  
  
}
```

```
void virarDireita() {  
  
// B frente (esquierdo roda)  
  
digitalWrite(B_RPWM, HIGH);  
  
digitalWrite(B_LPWM, LOW);  
  
  
// A parado  
  
digitalWrite(A_RPWM, LOW);  
  
digitalWrite(A_LPWM, LOW);
```

```
}
```

```
void virarEsquerdaTras() {  
    // B tras  
    digitalWrite(B_RPWM, LOW);  
    digitalWrite(B_LPWM, LOW);  
  
    // A parado  
    digitalWrite(A_RPWM, LOW);  
    digitalWrite(A_LPWM, HIGH);  
}
```

```
void virarDireitaTras() {  
    // B parado  
    digitalWrite(B_RPWM, LOW);  
    digitalWrite(B_LPWM, HIGH);  
  
    // A tras  
    digitalWrite(A_RPWM, LOW);  
    digitalWrite(A_LPWM, LOW);  
}
```

```
void comemorar() {  
    // B tras
```

```
digitalWrite(B_RPWM, LOW);
digitalWrite(B_LPWM, HIGH);

// A frente
digitalWrite(A_RPWM, HIGH);
digitalWrite(A_LPWM, LOW);
}
```

```
void pararMotores() {
    digitalWrite(A_RPWM, LOW);
    digitalWrite(A_LPWM, LOW);
    digitalWrite(B_RPWM, LOW);
    digitalWrite(B_LPWM, LOW);
}
```