

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

LEONARDO HENRIQUE TÁRTARI

PROTÓTIPO DE UM MEDIDOR DE COMPRIMENTO DE CONDUTORES
ELÉTRICOS ACIONADO POR ARDUINO

Joinville

2021

LEONARDO HENRIQUE TÁRTARI

PROTÓTIPO DE UM MEDIDOR DE COMPRIMENTO DE CONDUTORES
ELÉTRICOS ACIONADO POR ARDUINO

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica no curso de Engenharia Mecatrônica, da Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Orientador: Dr. Anderson Wedderhoff Spengler

Joinville

2021

RESUMO

Este trabalho apresenta o processo de desenvolvimento de um protótipo para medição de comprimento de condutores elétricos, acionado e controlado pela placa de desenvolvimento Arduino Uno. O objetivo é elaborar um projeto e a construção de um protótipo para a medição de comprimento de condutores elétricos para aplicação em uma empresa de automação. Foi utilizado o software Solidworks para o projeto das estruturas mecânicas, componentes eletrônicos de automação industrial foram reutilizados, simulação e validação da programação realizada no Tinkercad e uma interface homem-máquina foi desenvolvida utilizando *shield* para Arduino. Para a coleta dos resultados foram realizados testes em cabos com comprimentos entre 50mm e 2000mm e diâmetros de 6mm a 12,7mm. O protótipo foi capaz de realizar a medição com uma velocidade de 21,78mm/s com um erro médio de 1mm a cada 50mm medidos. Com a reutilização de materiais segregados foi possível economizar 94,23% do valor total considerando a realização do projeto com a compra dos itens.

Palavras-chave: Automação. Arduino. Medidor de comprimento.

ABSTRACT

This work presents the development process of a prototype for measuring the length of electrical conductors, driven and controlled by the development board Arduino Uno. The objective is design and build a prototype for measuring the length of electrical conductors for application in an automation company. Solidworks software was used for the design of mechanical structures, electronic automation components were reused, simulation and validation of programming performed in Tinkercad and a man-machine interface was developed using shield for Arduino. To collect the results, tests were carried out on cables with lengths between 50mm and 2000mm and diameters from 6mm to 12,7mm. The prototype was able to perform the measurement with a speed of 21.78mm/s with an average error of 1mm for every 50mm measured. With the reuse of segregated materials, it was possible to save 94.23% of the total price considering the completion of the project with the purchase of items.

Keywords: Automation. Arduino. Length gauge.

LISTA DE FIGURAS

Figura 1 – Imagem da placa Arduino modelo UNO.....	15
Figura 2 – Exemplo de shield para a placa Arduino.	16
Figura 3 – Exemplo de programação feita na IDE da placa de desenvolvimento Arduino.....	18
Figura 4 – Exemplo de um motor de relutância variável.	20
Figura 5 – Exemplo de um motor de imã permanente.	20
Figura 6 – Exemplo de um motor híbrido.	21
Figura 7 – Modelo construtivo das bobinas de um motor de passo unipolar.....	22
Figura 8 – Modelo construtivo das bobinas de um motor de passo bipolar.....	22
Figura 9 – Circuito elétrico de uma topologia “Ponte H” acionada de forma correta: C1 e C3 ou C2 e C4.....	23
Figura 10 – Circuito elétrico de uma topologia “Ponte H” acionada da forma incorreta (C1 e C2 ou C3 e C4).....	24
Figura 11 – Sequência de acionamento das bobinas de um motor de passo.	24
Figura 12 – Estrutura interna de uma fonte chaveada.	25
Figura 13 – Estrutura comum de engrenagens planetárias.....	26
Figura 14 – Redutor planetário de demonstração.	26
Figura 15 – Comparativo entre polia dentada e polia em V.	27
Figura 16 – Estrutura geral do desenvolvimento do protótipo de medidor de condutor elétrico automático.	30
Figura 17 – Dados de placa fonte chaveada utilizada.....	31
Figura 18 – Driver para motor de passo Leadshine.	32
Figura 19 – Arduino Uno R3 utilizado para o controle.....	33
Figura 20 – LCD Keypad Shield para Arduino Uno.	34
Figura 21 – Esquema de ligação dos componentes eletrônicos utilizados.	35
Figura 22 – Motor de passo Akiyama 52kgf.cm padrão NEMA 34.	37
Figura 23 – Curva de torque motor de passo 52kgf.cm.	38
Figura 24 – Desenho técnico motor de passo 52kgf.cm.	39
Figura 25 – Redutor planetário Apex modelo PE2-070-003.....	40
Figura 26 – Desenho técnico redutor planetário Apex utilizado.	41
Figura 27 – Mola de compressão utilizada.....	42
Figura 28 – Mola de compressão instalada no dispositivo de medição.....	42

Figura 29 – Rolamentos utilizados na mancalização	43
Figura 30 – Suportes para os rolamentos.	43
Figura 31 – Eixo central para a mancalização.....	44
Figura 32 – Polia 1.	44
Figura 33 – Polia 2.	45
Figura 34 – Simulação 3D do sistema mecânico completo.	46
Figura 35 – Dispositivo de medição projeto no software Solidworks.....	47
Figura 36 – Renderização do sistema de acionamento e medição.	47
Figura 37 – Marcação de posição correta.	49
Figura 38 – Disposição legendada de cada item no sistema.	49
Figura 39 – Base de rolagem projetada.	50
Figura 40 – Projeto da mancalização dupla.	51
Figura 41 – Estrutura da base de rolagem dos carretéis feita em madeira e barras de rosca sem fim.	52
Figura 42 – Menu de escolha inicial.	54
Figura 43 – Fluxograma demonstrativo do código desenvolvido.....	55
Figura 44 – Simulação e validação da estrutura lógica no software Tinkercad.	56
Figura 45 – Montagem final do sistema de medição.	57
Figura 46 – Barra de rosca sem fim adicionada para otimização do sistema mecânico.	58
Figura 47 – Base giratória para carretéis finalizada.	59
Figura 48 – Mancalização da base giratória finalizada.....	59

LISTA DE QUADROS

Quadro 1 – Especificações técnicas do motor de passo AK34/52F4CN1.8.....	38
Quadro 2 – Identificação dos fios do motor de passo utilizado	39
Quadro 3 – Especificações técnicas do redutor utilizado modelo PE2-070-003.	41
Quadro 4 – Lista de materiais especificados para a montagem do dispositivo de medição.....	48

LISTA DE TABELAS

Tabela 1 – Informações sobre os cabos e carretéis utilizados para medição.	29
Tabela 2 – Entradas e resultados obtidos na realização do segundo teste.	60
Tabela 3 – Terceiro teste de medição de cabos.....	61
Tabela 4 – Custo dos itens e total considerando a possibilidade de compra de todos os itens do projeto.	62
Tabela 5 – Custo dos itens e total aplicado no projeto.....	63

LISTA DE ABREVIATURAS E SIGLAS

AWG – American Wire Gage

CAD – Computer-Aided Design

CLP – Controlador Lógico Programável

CNC – Computer Numerical Control

EEPROM – Electrically-Erasable Programmable Read-Only Memory

IDE – Integrated Development Environment

RAM – Random Access Memory

RISC – Reduced Instruction Set Computer

PPR – Pulsos Por Revolução

LISTA DE SÍMBOLOS

UNIDADES

A - Ampere

V – Volts

Vac – Volts (tensão alternada)

Vdc – Volts (tensão contínua)

W – Watt

m - Metros

mm – Milímetros

rpm – Rotações por minuto

N – Newton

Nm – Newton metro

kg – Quilograma

m/s – Metros por segundo

m/min – Metros por minuto

Hz - Hertz

PREFIXOS

giga (G) – 10^9

mega (M) – 10^6

kilo (k) – 10^3

mili (m) – 10^{-3}

micro (μ) – 10^{-6}

nano (n) – 10^{-9}

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVOS	13
1.1.1 Objetivo geral	13
1.1.2 Objetivos específicos.....	13
2 REVISÃO TEÓRICA	14
2.1 ARDUINO.....	14
2.1.1 Microcontrolador	15
2.1.2 Shields.....	16
2.1.3 Interface de desenvolvimento	17
2.2 MOTOR DE PASSO.....	19
2.2.1 Modelos de motores de passo	19
2.2.2 Tipos de construções	21
2.3 DRIVER DE ACIONAMENTO	23
2.4 FONTE CHAVEADA	25
2.5 ENGRENAGENS PLANETÁRIAS.....	25
2.6 POLIAS	27
3 MATERIAIS E MÉTODOS	29
3.1 ESTRUTURA ELETRÔNICA.....	30
3.1.1 Fonte Chaveada.....	31
3.1.2 Driver	32
3.1.3 Microcontrolador	33
3.1.4 Ligações eletrônicas	35
3.2 ESTRUTURA MECÂNICA.....	35
3.2.1 Motor de passo	36
3.2.2 Redutor planetário.....	39
3.2.3 Mola	42
3.2.4 Mancais	43
3.2.5 Polias.....	44
3.2.6 Projeto Solidworks	45
3.3 ESTRUTURA LÓGICA.....	52
3.3.1 Lógica de medição	52
3.3.2 Programação	53

4 RESULTADOS.....	57
4.1 PROTÓTIPO	57
4.2 MEDIÇÕES	60
4.3 CUSTO.....	62
4.4 LIMITANTE ENCONTRADA.....	63
5 CONCLUSÃO	65
REFERÊNCIAS.....	67
APÊNDICE A – DIMENSÕES POLIA 1	70
APÊNDICE B – DIMENSÕES POLIA 2.....	71
APÊNDICE C – HASTE DE MANCALIZAÇÃO DA POLIA 2 E FIXAÇÃO DA MOLA DE TRACIONAMENTO	72
APÊNDICE D – SUPORTE FIXADOR POLIA 1	73
APÊNDICE E – SUPORTE FRONTAL 1	74
APÊNDICE F – SUPORTE FRONTAL 2	75
APÊNDICE G – CÓDIGO COMPLETO NA LINGUAGEM C.....	76

1 INTRODUÇÃO

O uso de microcontroladores está crescendo cada vez mais, com isso, podemos encontrar algumas situações no nosso cotidiano como por exemplo: em relógios inteligentes, patinetes elétricos, projetos universitários visando um baixo investimento e sensoriamento de atividades, afim de reduzir custos e ter uma maior facilidade em seu uso mantendo um ótimo desempenho.

Segundo Cunha (2021) os sistemas automáticos, por exemplo, os micro-ondas residenciais possuem uma capacidade computacional maior do que tinha o projeto Apolo, que levou o homem à lua, portanto, uma tendência que se observa é que cada vez mais inteligência será adicionada aos equipamentos do dia-a-dia.

Em automatização de sistemas, as vantagens segundo Pazos (2002) são: melhoria da produtividade devido a efetividade na realização da tarefa, melhoria da qualidade do produto devido a precisão de sua tarefa e redução de desperdícios de matéria prima.

Os microcontroladores propiciam inovações tecnológicas que antes do processo de miniaturização do processamento não eram possíveis. Stevan Jr. E Silva (2015) afirmam que existem diferentes microcontroladores no mercado e requisitos de cada projeto que diferenciam famílias e aplicações. Visando reduzir justamente o tempo de projeto de um sistema de automação, busca-se plataformas padronizadas que permitem a utilização simples e rápida de entradas e saídas, através de conexões comuns, como por exemplo o microcontrolador Arduino amplamente difundido.

Para este estudo de aplicação, será realizado um medidor de comprimento de condutores elétricos utilizando o microcontrolador Arduino, automatizando o trabalho manual existente.

A dúvida que surge é: por que realizar um protótipo de medidor de comprimento de cabos para substituir o trabalho manual que já era feito antes? Sendo assim, foi desenvolvido o projeto e construção de um protótipo visando sanar as dúvidas mais pertinentes sobre este assunto.

Portanto, desejando reduzir o desperdício de cabos, a problemática a ser respondida busca mostrar como construir um medidor de condutores elétricos acionado por Arduino em uma empresa de automação, reutilizando itens segregados.

1.1 OBJETIVOS

Para resolver a problemática de como construir um medidor de condutores elétricos acionado por Arduino em uma empresa de automação, propõe-se neste trabalho os seguintes objetivos:

1.1.1 Objetivo geral

Projeto e construção de um protótipo de medidor de condutores elétricos para aplicação em uma empresa de automação.

1.1.2 Objetivos específicos

- Projetar um dispositivo mecânico para a medição de condutores elétricos;
- Projetar uma base para os carretéis de cabos a serem medidos;
- Realizar a programação do controle na placa de desenvolvimento Arduino;
- Realizar a interface homem-máquina de forma amigável, simples e objetiva;
- Analisar a precisão e desempenho do protótipo.

2 REVISÃO TEÓRICA

Nesta seção, serão destacados os principais temas e autores relacionados a aplicação do medidor de condutores elétricos acionado por Arduino, como por exemplo: o que é um microcontrolador Arduino e qual sua finalidade; o que é um motor de passo acionado por um driver e porque são amplamente utilizados nos dias atuais; quais são os condutores elétricos utilizados para a execução deste trabalho; e por fim, o que é um redutor planetário e por que ele é um item mecânico que revolucionou o acionamento das máquinas.

2.1 ARDUINO

O Arduino foi criado em 2005 na Itália, é composto de um protótipo de hardware e software, de fonte aberta (*open source*), possibilitando que qualquer pessoa possa criar projetos sem precisar pagar direitos autorais aos criadores (JUNIOR, SILVA, 2015).

Segundo Monk (2013) o Arduino permite uma conexão entre circuitos eletrônicos aos seus terminais, de modo que o mesmo consiga controlar dispositivos como ligar e desligar lâmpadas e motores, ou medir luz e temperatura.

Figura 1 – Imagem da placa Arduino modelo UNO.



Fonte: Schmidt (2011).

Portanto, Arduino é uma plataforma de prototipagem eletrônica baseada em flexibilidade, onde hardware e software são fáceis de serem utilizados e adaptados a diferentes aplicações (JUNIOR, SILVA, 2015).

Existem muitas variações do Arduino, sendo a versão mais recente e também mais popular, o Arduino Uno (MCROBERTS, 2011). Junior e Silva (2015) também afirmam que o UNO é a versão mais disseminada da família Arduino, por contar com um microcontrolador de entrada e bons recursos juntamente com um número de interfaces suficiente para muitos projetos de baixa complexidade.

2.1.1 Microcontrolador

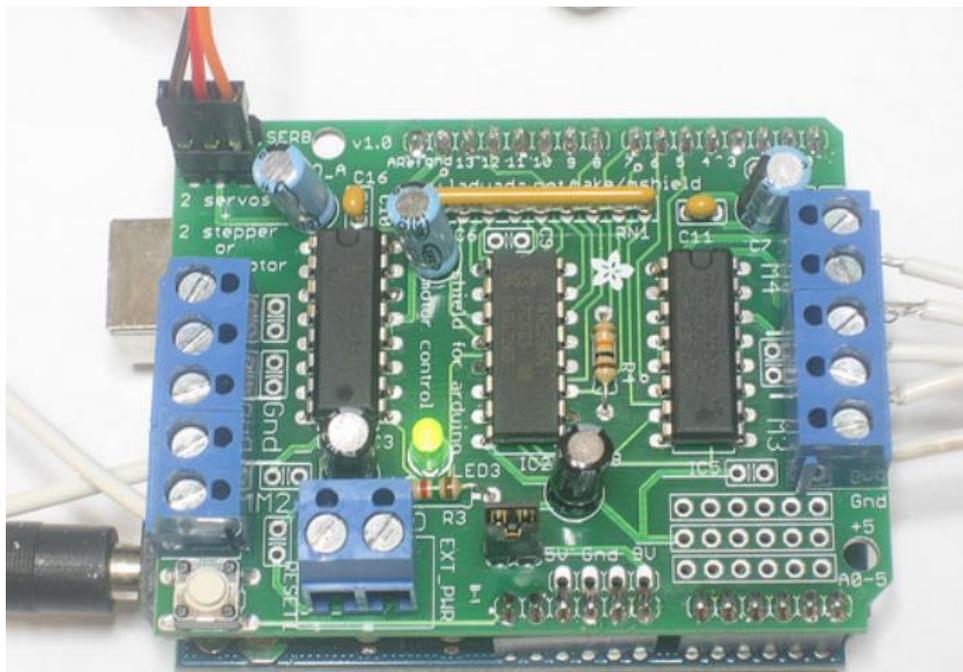
O microcontrolador de uma placa Arduino é o circuito integrado de 28 pinos que é encontrado no centro da placa, este chip contém a memória, o processador e toda eletrônica necessária para os pinos de entrada e saída (MONK, 2013). Ainda segundo Monk (2013) este circuito é fabricado pela empresa Atmel, que é uma das maiores fabricantes de microcontroladores para diferentes áreas.

O componente principal de uma placa Arduino UNO é seu microcontrolador ATMEGA328, que é um microcontrolador de 8 bits com arquitetura RISC (*Reduced Instruction Set Computer*), este que pode ser encontrado em diferentes encapsulamentos, possui 32KB de memória flash para programas, 2KB de dados do tipo RAM (*Random Access Memory*), e 1KB de memória de dados do tipo EEPROM (*Electrically-Erasable Programmable Read-Only Memory*), 14 entradas e saídas digitais para propósito geral que operam com nível lógico alto de 5V e baixo de 0V e 6 canais analógicos que operam com um conversor analógico-digital de 10 bits (JUNIOR; SILVA, 2015).

2.1.2 Shields

Segundo Monk (2013), outra razão para o sucesso do Arduino em diversas aplicações é que ele não se limita apenas a placas com o microcontrolador, existem diversos acessórios para Arduino que são denominados *Shields*, onde pode ser encaixada diretamente sobre a placa.

Figura 2 – Exemplo de *shield* para a placa Arduino.



Fonte: Evans et al. (2013)

Estes periféricos chamados de Shields, são placas de circuitos impresso que utilizam a padronização tanto geométrica quanto de pinos para se comunicar com os periféricos adicionados, aumentando as funcionalidades (JUNIOR; SILVA, 2015). Considerando os componentes da Figura 2, pode-se observar que a placa de desenvolvimento Arduino se encontra na parte de baixo e logo acima, encaixado sobre o microcontrolador, se encontra o *shield*, neste caso expandindo a funcionalidade do Arduino como uma placa para controlar motores de corrente contínua, motores de passo e servo motores.

2.1.3 Interface de desenvolvimento

O *software* utilizado para realizar a programação da placa Arduino, também chamado de IDE (*Integrated Development Environment*), em primeira atribuição é uma ferramenta de edição de linguagem de alto nível (linguagem C), em outros termos, é quem compila o código na linguagem C e o converte para a linguagem Assembly, assim convertendo para código binário e então embarcar o código no Arduino (JUNIOR; SILVA, 2015).

Segundo McRoberts (2011), o Arduino executará as instruções programadas via sua IDE interagindo com o que estiver conectado a ele.

Junior e Silva (2015) mencionam que o arquivo onde é desenvolvido o código na linguagem C é conhecido como *sketch* e de forma simples, a estrutura de um sketch é formada por três partes:

- Declaração de variáveis;
- Função setup, com declarações, inicializações ou funções que apenas são executadas uma vez;
- Função loop, com funções que se repetirão.

Figura 3 – Exemplo de programação feita na IDE da placa de desenvolvimento Arduino.

```

Fade §
int led = 9;           // porta utilizada para o LED
int brilho = 0;       // qual o brilho do LED
int velocidade = 5;   // quao rapido sera a transicao

void setup() {
  // declara que o pino 9 eh saida
  pinMode(led, OUTPUT);
}

// loop infinito - programa
void loop() {
  // seta pino 9 como brilho
  analogWrite(led, brilho);

  // muda o brilho para o proximo loop
  brilho = brilho + velocidade;

  // inverte o brilho quando chega no max ou min
  if (brilho <= 0 || brilho >= 255) {
    velocidade = -velocidade;
  }
  // delay de 30milisegundos para visulalizar o efeito
  delay(30);
}

```

Declaração de variáveis

Setup do programa

Desenvolvimento do Programa (loop)

7 Arduino Uno em COM3

Fonte: Autor (2021).

Na Figura 3 se encontra um programa na interface de programação do Arduino onde a função é aumentar e diminuir o brilho de um LED. Primeiramente é declarado as variáveis utilizadas posteriormente no programa, por exemplo: variável brilho e velocidade. No *setup* do programa é declarado o pino onde o LED se encontra (pino 9) como uma saída analógica. Por fim, no desenvolvimento do programa, primeiramente é acionado o LED na intensidade que consta na variável brilho, seguido dos ajustes para o próximo *loop* da função: ajuste do brilho e a condição para verificar se chegou no máximo ou mínimo do brilho para então inverter a intensidade.

2.2 MOTOR DE PASSO

Os motores de passo são dispositivos eletromecânicos que convertem pulsos elétricos em movimentos mecânicos que geram variações angulares discretas. Seu rotor também denominado de eixo do motor de passo, é rotacionado em pequenos incrementos angulares ou “passos” quando pulsos elétricos são aplicados em uma determinada sequência nos terminais de alimentação do motor (BRITES; SANTOS, 2008).

Segundo Evans et al. (2013) os motores de passo são uma boa escolha para projetos que requerem um controle e posicionamento preciso, pré-requisitos típicos de impressoras 3D, posicionamento de telescópios e máquinas CNC (*Computer Numerical Control*). Brites e Santos (2008) mencionam que o ponto forte de um motor de passo não é o torque que ele pode gerar, tampouco sua capacidade de desenvolver altas velocidades, mas sim a possibilidade de controlar seus movimentos de forma precisa.

Umans (2014) também afirma que uma outra característica importante do motor de passo é sua compatibilidade com os sistemas eletrônicos digitais, tornando-se comuns em uma variedade de aplicações.

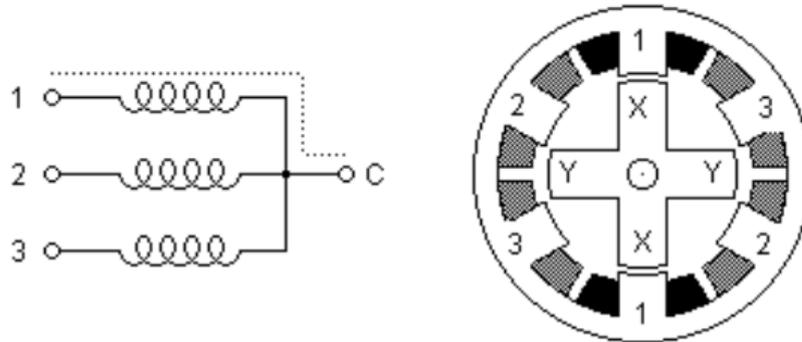
2.2.1 Modelos de motores de passo

Atualmente existem três topologias de motor de passo, divididos em modelos diferentes, são eles: relutância variável, ímã permanente e híbrido (Condit et al., 2014).

2.2.1.1 Relutância Variável

Este modelo de motor de passo possui de três a cinco bobinas de fios em seu estator (parte estacionária do motor), todas conectadas a um ponto comum. Conforme a Figura 4, o rotor (parte móvel) possui quatro dentes e seu estator seis polos, sendo cada enrolamento disposto em sentido opostos e a 30° do próximo, logo, quando o polo 1 é energizado, o dente X é atraído, então é desativado o polo 1 e energizado o polo 2 onde o dente Y será atraído, de forma contínua esta operação é feita para todas as bobinas, onde haverá atração e conseqüentemente a rotação do rotor (Condit et al., 2014).

Figura 4 – Exemplo de um motor de relutância variável.

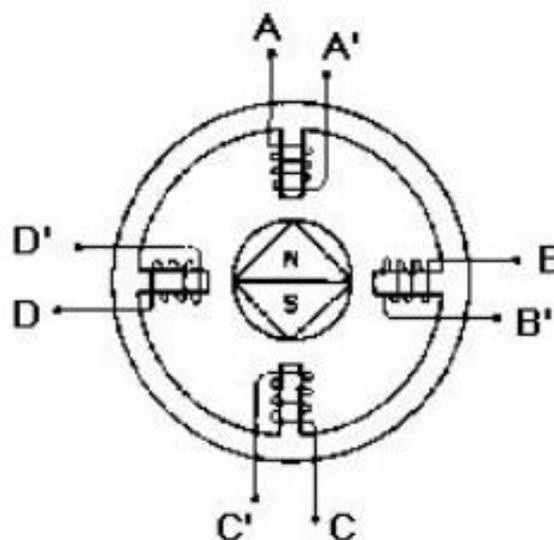


Fonte: Condit et al. (2014).

2.2.1.2 Imã permanente

Segundo Brites e Santos (2008), os motores de passo de imã permanente possuem baixo custo e baixa resolução, sendo seu rotor construído de imãs sem a presença de dentes. Este modelo possui uma maior intensidade de fluxo magnético resultando numa melhor geração de torque se comparado com o motor de relutância variável. Sua resolução varia de 7,5 a 15°.

Figura 5 – Exemplo de um motor de imã permanente.

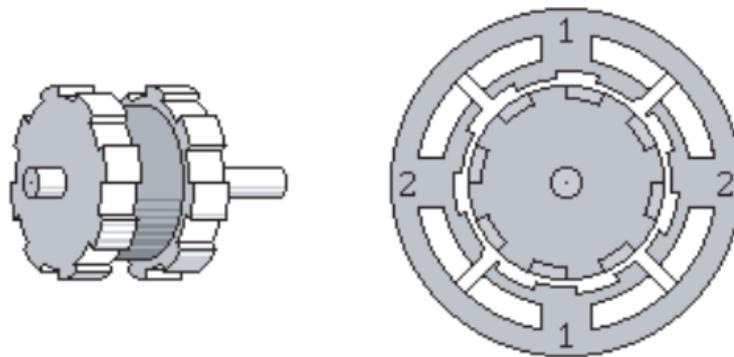


Fonte: Brites e Santos (2008).

2.2.1.3 Híbrido

Segundo Condit et al. (2014), o motor de passo híbrido possui as melhores características dos motores de passo de relutância variável e ímãs permanentes. O rotor do motor de passo híbrido contém vários dentes como um motor de passo de relutância variável e possui ímãs dispostos em torno de seu rotor conforme a Figura 6.

Figura 6 – Exemplo de um motor híbrido.



Fonte: Condit et al. (2014).

Ainda segundo Condit et al. (2014) os múltiplos dentes do rotor ajudam a fornecer um melhor caminho para o fluxo magnético no entreferro do motor de passo. Em relação aos ímãs, os mesmos ajudam a manter o torque dinâmico.

2.2.2 Tipos de construções

Segundo Evans (2013), existem dois tipos de construções internas possíveis para motores de passos: a forma unipolar e bipolar.

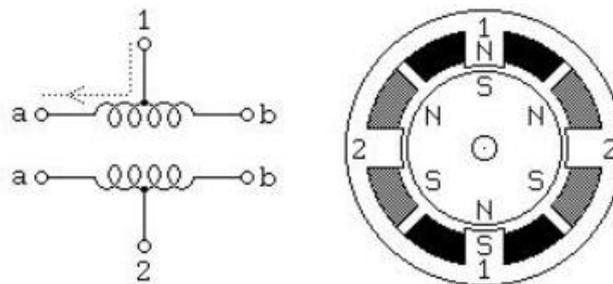
2.2.2.1 Motor de passo unipolar

Motor de passo unipolar é composto por dois enrolamentos no estator, cada um com uma derivação em seu centro (conforme a Figura 7), estas derivações são

trazidas para fora do motor como dois fios separados, logo, um motor de passo unipolar possui seis fios (Condit et al., 2014).

Evans (2013) também observa que um motor de passo unipolar pode ser controlado da mesma forma do que um motor bipolar, se feito as conexões da maneira correta.

Figura 7 – Modelo construtivo das bobinas de um motor de passo unipolar.

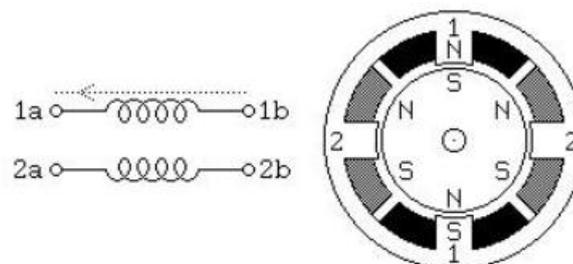


Fonte: Condit et al. (2014).

2.2.2.2 Motor de passo bipolar

Segundo Condit et al. (2014), os motores de passo bipolares possuem dois enrolamentos e quatro fios. Diferentemente do motor unipolar, o bipolar não possui derivações centrais, a vantagem dessa estrutura é que a corrente elétrica percorre por todo o enrolamento, e não somente metade como o unipolar, como consequência, os motores bipolares produzem mais torque.

Figura 8 – Modelo construtivo das bobinas de um motor de passo bipolar.



Fonte: Condit et al., (2008)

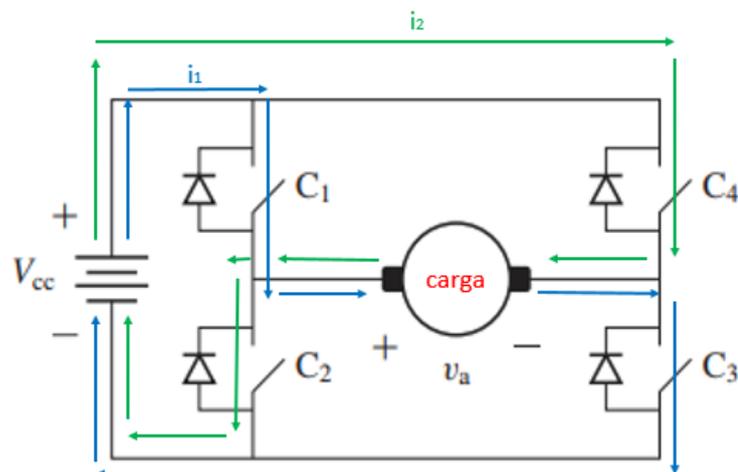
É importante destacar que independentemente do tipo de construção do motor de passo, é adotada uma nomenclatura para suas bobinas, normalmente adotadas como A+, A-, B+ e B-.

2.3 DRIVER DE ACIONAMENTO

Segundo Pazos (2002), um driver é um circuito digital que amplifica um sinal digital de baixa potência. Isso permite ativar ou desativar um atuador de maior potência. Para o acionamento de forma correta da carga, é utilizado a topologia de circuito chamada Ponte H.

Ponte H é uma estrutura eletrônica difundida, que permite a rotação de um motor tanto para um sentido quanto para outro. Ele é constituído por quatro chaves acionadas intercaladamente (C1 e C3 ou C2 e S4) como é mostrado na Figura 9 (BRITES; SANTOS, 2008). Foi representado o sentido das correntes elétricas no caso do acionamento na sequencia correta indicada.

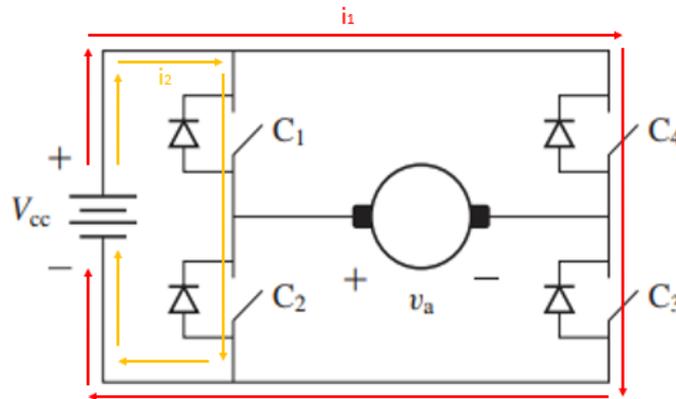
Figura 9 – Circuito elétrico de uma topologia “Ponte H” acionada de forma correta: C1 e C3 ou C2 e C4.



Fonte: Adaptado de Umans (2014).

Caso as chaves C1 e C2 ou C3 e C4 forem acionadas ao mesmo tempo, haverá um curto circuito na fonte de alimentação (BRITES; SANTOS, 2008). Na Figura 10 é representado o sentido das correntes onde pode-se verificar o motivo pelo qual ocorre curto circuito.

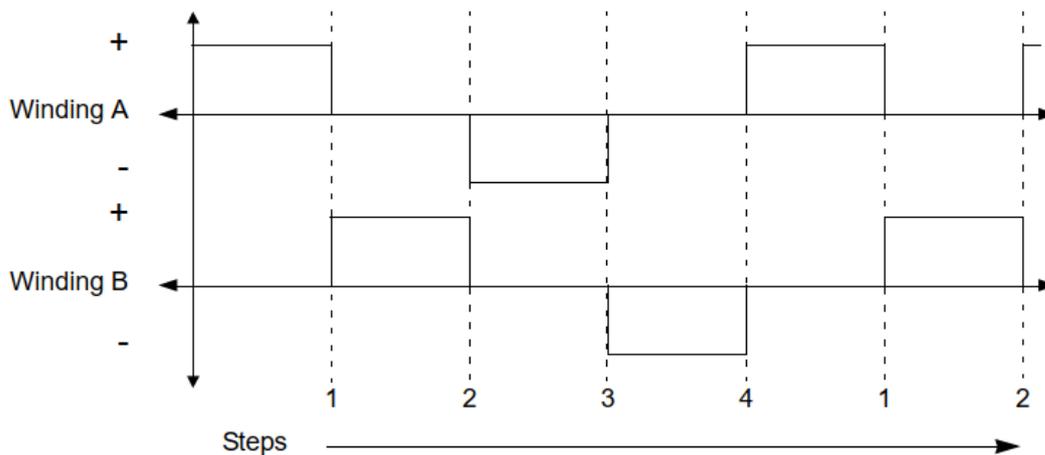
Figura 10 – Circuito elétrico de uma topologia “Ponte H” acionada da forma incorreta (C1 e C2 ou C3 e C4).



Fonte: Adaptado de Umans (2014).

Yedamale et al. (2002) afirma que drivers unipolares podem ser usados unicamente com motores de passo unipolares. Drivers bipolares conseguem acionar ambos os tipos de motores de passo (unipolar ou bipolar), ao realizar a ação de apenas ignorar os fios centrais que dividem as bobinas, um motor unipolar se torna bipolar.

Figura 11 – Sequência de acionamento das bobinas de um motor de passo.



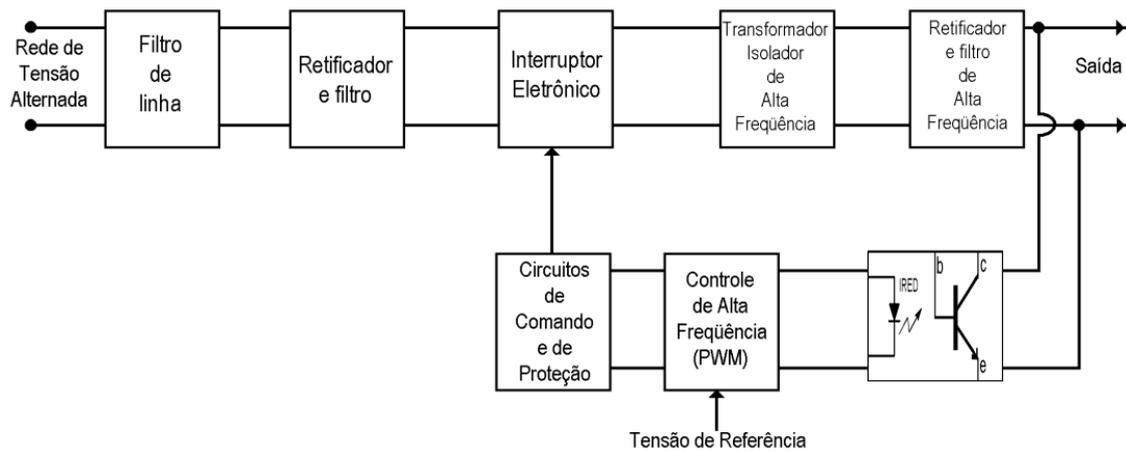
Fonte: Yedamale et al. (2002).

Yedamale et al. (2002) afirma que são utilizadas duas pontes H para o acionamento dos motores de passo, sendo uma ponte para cada bobina do motor de passo, conforme é possível verificar na Figura 11 onde os dois primeiros passos são A+ e B+ seguidos pelo acionamento inverso A- e B-.

2.4 FONTE CHAVEADA

Segundo Barbi (2001) as fontes chaveadas começaram a ser desenvolvidas na década de 60 para uso em programas espaciais, cujo objetivo era substituir as fontes lineares, que são volumosas e pesadas, por fontes compactas e de alto rendimento.

Figura 12 – Estrutura interna de uma fonte chaveada.



Fonte: Mehl (2021).

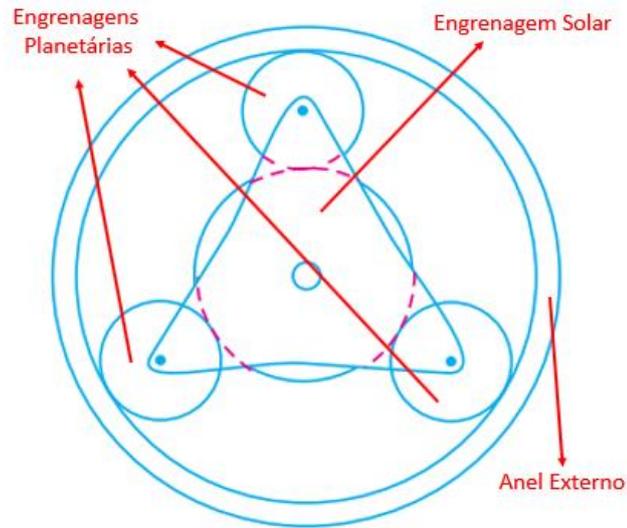
Uma fonte chaveada é basicamente um conversor, que por meio do chaveamento de um dispositivo semicondutor em alta frequência, consegue regular a tensão de saída em corrente contínua, possuindo a seguinte estrutura conforme a Figura 12 (MEHL, 2021).

2.5 ENGRENAGENS PLANETÁRIAS

Um conjunto de engrenagens planetárias segundo Collins et al. (2010), é uma sequência de várias engrenagens acopladas de tal forma que a velocidade, torque e sentido de rotação da saída desejados são obtidos utilizando uma entrada específica.

Os conjuntos de engrenagens planetárias apresentam vantagens como: as relações de redução são maiores em tamanhos menores e dispõem de entradas e saídas concêntricas e bidirecionais. (Norton, 2013).

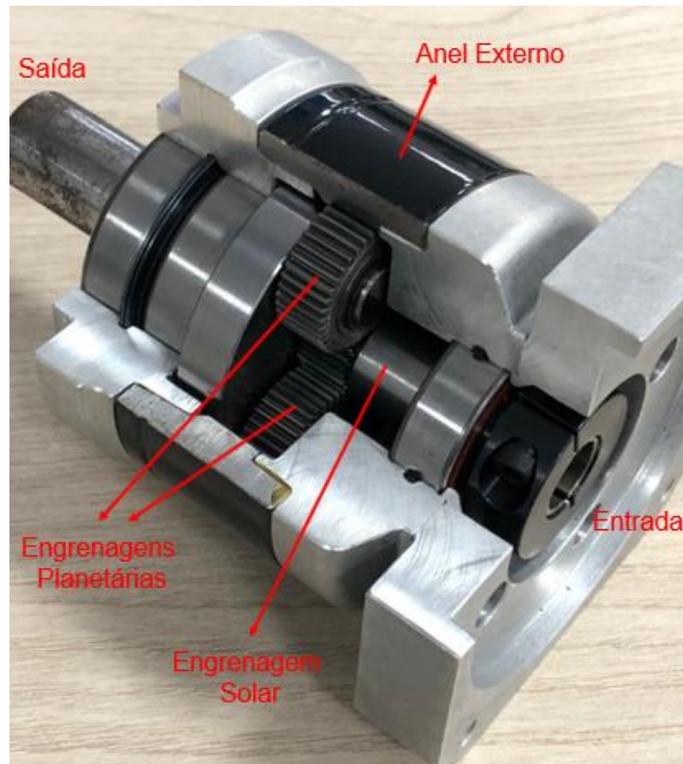
Figura 13 – Estrutura comum de engrenagens planetárias.



Fonte: Adaptado de Khurmi (2005).

Müller (1982) afirma que as engrenagens planetárias consistem de alguns itens importantes como por exemplo uma engrenagem solar, três engrenagens planetárias, um suporte de sincronismo e um anel externo de acordo com a Figura 13.

Figura 14 – Redutor planetário de demonstração.



Fonte: Autor (2021)

Segundo Höhn et al (2013), a potência mecânica transmitida pela engrenagem solar é distribuída para as três engrenagens planetárias, resultando na redução de carga e em menor esforço aplicado em seus dentes.

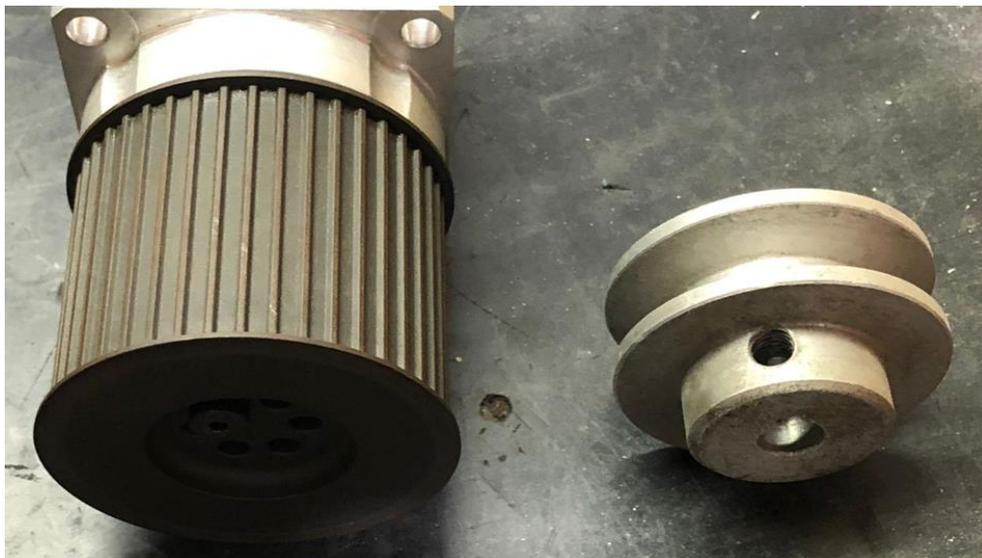
2.6 POLIAS

Como caracteriza Essel (2021), as polias são dispositivos mecânicos circulares, que podem ou não possuir canais, acopladas em motores e movidas por equipamentos. Ainda na visão de Essel (2021), para o funcionamento correto das polias, as mesmas precisam da presença de algum outro dispositivo ou material como correias ou até mesmo cordas, fios ou cabos.

Do ponto de vista de Lino (2013), a polia que é conectada ao motor ou algum dispositivo que origina o movimento, é chamada de polia motora, já a polia que recebe o movimento de uma polia motora é chamada de polia movida ou conduzida.

Na Figura 15 pode-se visualizar a diferença entre uma polia dentada visando a operação com correia também dentada em relação a uma polia em V.

Figura 15 – Comparativo entre polia dentada e polia em V.



Fonte: Autor (2021).

Essel (2021) alega que para o funcionamento adequado das polias, deve-se tomar algumas precauções:

- Não existir desgastes nos canais da polia;
- Não existir bordas quebradas, trincadas, amassadas ou oxidadas;
- Garantir que os canais das polias estejam livre de graxas, óleos ou qualquer outro lubrificante.

3 MATERIAIS E MÉTODOS

Nesta seção será explicado quais os materiais e métodos utilizados para o desenvolvimento do projeto protótipo do medidor de condutores elétricos automático acionado por Arduino e qual a estrutura geral do sistema foi adotada. É interessante destacar que a maioria dos materiais utilizados foram reaproveitados da área de segregação, onde possuem pequenos defeitos, que tornam o produto não comercializável.

Para o trabalho em questão foi adotada uma abordagem qualitativa. A pesquisa a ser desenvolvida para o cumprimento dos objetivos específicos é de caráter exploratório, pois de acordo com Gil (2002), uma pesquisa exploratória tem como objetivo construir uma maior familiaridade e o aprimoramento de ideias. A metodologia adotada para a realização dos procedimentos foi de nível experimental, com o intuito de cumprir com o objetivo geral levando em consideração a geração de conhecimento sobre o assunto e suas ferramentas.

O projeto do protótipo foi separado em três estruturas, sendo elas: eletrônica, mecânica e lógica. Conforme a Figura 16 a seguir, em cada estrutura será explicado quais foram os materiais e métodos utilizados demonstrando validações, simulações e projetos de componentes mecânicos.

Os cabos que possuem menor e maior espessura utilizados pela empresa são de 4 milímetros e 12,7 milímetros respectivamente, portanto, foi utilizado para a medição três tipos de cabos com valor de espessura variados. A Tabela 1 abaixo mostra as informações de cada carretel.

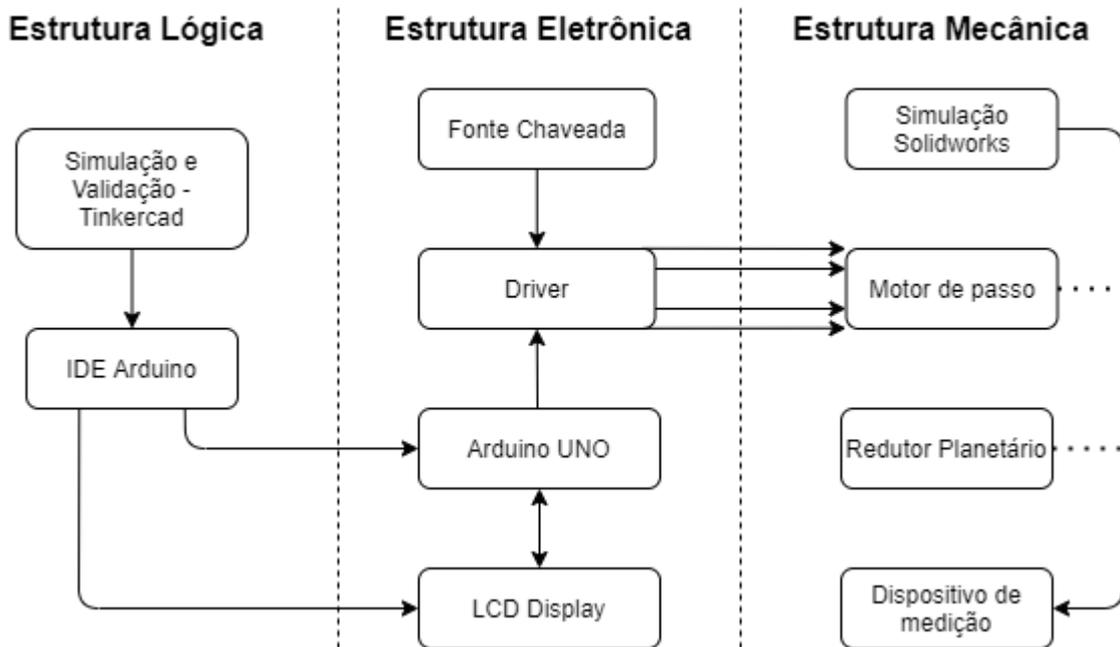
Tabela 1 – Informações sobre os cabos e carretéis utilizados para medição.

Espessura do cabo [mm]	Peso do carretel [kg]
6	6
9,5	8,5
12,7	15

Fonte: Autor (2021).

Todos os carretéis possuem aproximadamente 20 metros de cabos enrolados. O valor que pode variar devido a utilização mais frequente de um modelo específico de cabo.

Figura 16 – Estrutura geral do desenvolvimento do protótipo de medidor de condutor elétrico automático.



Fonte: Autor (2021).

Foi definido alguns requisitos para a realização dos cálculos dos componentes utilizados e padronização de variáveis, são eles:

- Velocidade de 50rpm na saída;
- Rampa de aceleração de 0,1s;
- Raio médio dos carretéis de 250mm;
- Fator de segurança para os cálculos de 30%;
- Peso máximo do carretel de cabo de 20kg;
- Quantidade máxima de medição de 20 metros;

3.1 ESTRUTURA ELETRÔNICA

Nesta seção é apresentado os componentes eletrônicos utilizados, suas características e suas respectivas ligações.

3.1.1 Fonte Chaveada

Foi realizado o cálculo da potência mecânica necessária do sistema utilizando a equação de potência segundo Chapman (2013).

$$P = \frac{Tn}{9550} \text{ [kW]} \quad (1)$$

Sendo T o torque necessário do sistema em Nm, n a velocidade angular do motor em rpm e por fim, P a potência mecânica consumida pelo sistema em kW.

Considerando alguns resultados obtidos na seção 3.2.1, temos uma potência mecânica consumida de aproximadamente 70W.

Figura 17 – Dados de placa fonte chaveada utilizada.



Fonte: Autor (2021)

Para o acionamento do conjunto motor de passo e driver, foi utilizado uma fonte chaveada com alimentação 110V/220Vac e saída 24Vdc. Esta fonte possui uma potência de 200W e corrente máxima de 8,3A.

3.1.2 Driver

O driver para motor de passo utilizado é da marca *Leadshine* (Figura 18), modelo DMA860E. Este dispositivo possui diferentes tipos de resoluções para acionamento do motor, também chamado de PPR ou Pulsos Por Revolução e diferentes correntes máximas dependendo da força do motor de passo e da aplicação.

Neste projeto foi configurado o driver para 400 pulsos por revolução, portanto, é obtido uma volta completa no motor de passo caso seja enviado 400 pulsos para o driver. O motivo pelo qual foi escolhido esta configuração se deve ao fato de alcançar a maior velocidade de saída que o driver pode entregar, pois assim é necessário gerar o mínimo de pulsos para se obter uma volta completa na saída do motor.

Em relação a corrente do driver, não foi necessário limitar a corrente por se tratar de um acionamento simples, logo, a corrente foi configurada para o máximo de 7,2A de pico.

Figura 18 – Driver para motor de passo Leadshine.



Fonte: Autor (2021).

As portas ENA- e ENA+ do driver não foram utilizadas. A porta que recebe os pulsos enviados pelo microcontrolador Arduino é a PUL+, sendo assim foi alimentado a porta DIR+ com 5V pois é esta entrada que é responsável pelo sentido de rotação do motor. As portas DIR- e PUL- não foram utilizadas, logo, foram ligadas no terminal terra (GND) do microcontrolador.

Para a alimentação do motor de passo, o driver utiliza as nomenclaturas A+, A-, B+ e B-.

3.1.3 Microcontrolador

Foi necessário um controlador para receber a informação do usuário da quantidade de cabo desejada, realizar os cálculos da quantidade de pulsos necessários para o acionamento correto do motor de passo, e finalmente o envio dos pulsos para o drive. Para isso foi utilizado o microcontrolador Arduino Uno R3 como é mostrado na Figura 19.

Figura 19 – Arduino Uno R3 utilizado para o controle.

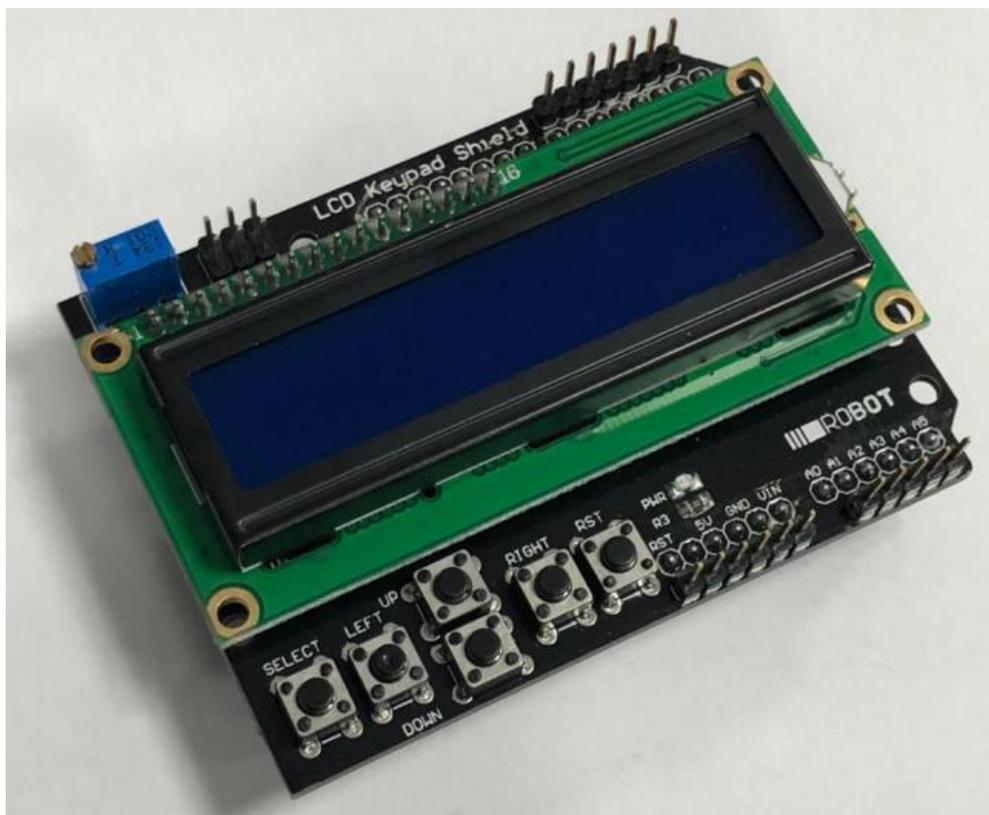


Foram utilizadas seis saídas digitais para o controle do *Display Shield* sendo as portas: 4, 5, 6, 7, 8 e 9. Para realizar o envio dos pulsos para o driver, foi utilizado a porta digital 11. A porta digital 12 foi utilizada para controlar o sentido de rotação do motor de passo. Por fim, a porta analógica A0 foi utilizada para verificar o estado dos botões do *Display Shield*.

3.1.3.1 Display Shield

Para realizar a Interface Homem-Máquina (IHM) levando em consideração a simplicidade, foi utilizado o *LCD Keypad Shield* mostrado na Figura 20. Este *shield* desenvolvido para Arduino possui um display 16x2 (16 colunas por 2 linhas) para exposição de informações relevantes como seleção de comprimento desejado e comprimento atual de medição, juntamente com cinco botões configuráveis para a obtenção de informações.

Figura 20 – *LCD Keypad Shield* para Arduino Uno.

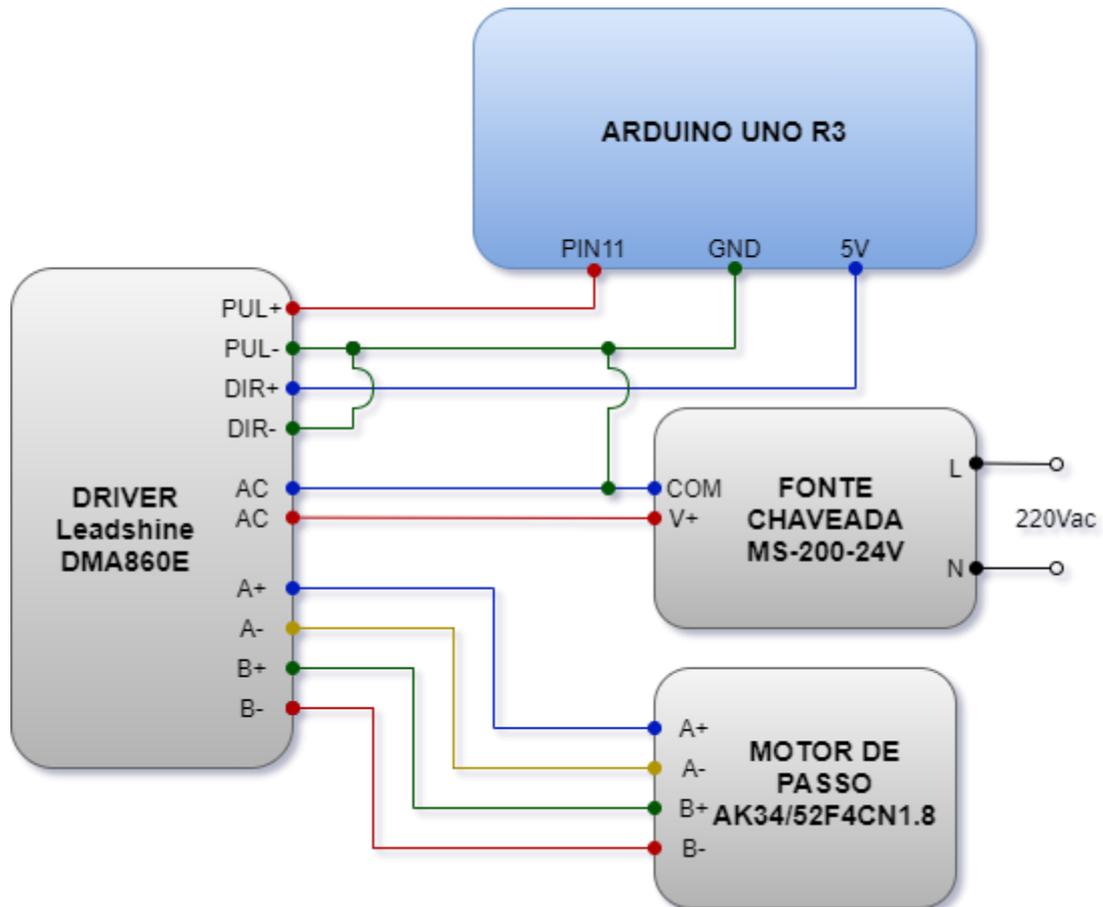


Fonte: Autor (2021).

3.1.4 Ligações eletrônicas

Um demonstrativo de todas as ligações eletrônicas realizadas entre os componentes pode ser conferido na Figura 21.

Figura 21 – Esquema de ligação dos componentes eletrônicos utilizados.



Fonte: Autor (2021).

3.2 ESTRUTURA MECÂNICA

Nesta seção será explicado quais foram os componentes mecânicos utilizados, suas características e projetos realizados.

3.2.1 Motor de passo

Apesar de o motor de passo ser um item eletromecânico, a consideração deste dispositivo na seção mecânica é para a realização dos cálculos da força necessária que o sistema mecânico vai exigir.

Para a escolha do motor de passo que melhor atende ao protótipo proposto foi necessário realizar o cálculo da força para o tracionamento e medição de condutor, para isto, foi utilizado a Equação (2) conforme APEX DYNAMICS (2021).

$$F = (ma)fs \quad (2)$$

Onde F é a força necessária para tirar o sistema da inércia; m é a massa do carretel de cabo; a é a aceleração linear do sistema; por fim fs é o fator de segurança.

Foi preciso calcular a aceleração linear do sistema, para tal, foi considerado a equação de cálculo de aceleração linear em função da velocidade linear conforme a equação (Chapman, 2013).

$$a = \frac{V_l}{t_a} \quad (3)$$

Onde V_l é a velocidade linear e t_a é o tempo de aceleração ou rampa de aceleração do sistema.

Para a resolução da Equação (3), foi preciso calcular qual a velocidade linear com base no requisito de velocidade angular de saída de medição em 50RPM. Utilizando a Equação (4) foi possível encontrar a velocidade linear do sistema segundo Autor (2021).

$$V_l = nP_r \quad (4)$$

Onde n é a velocidade angular e P_r é o perímetro da polia motora do dispositivo de medição.

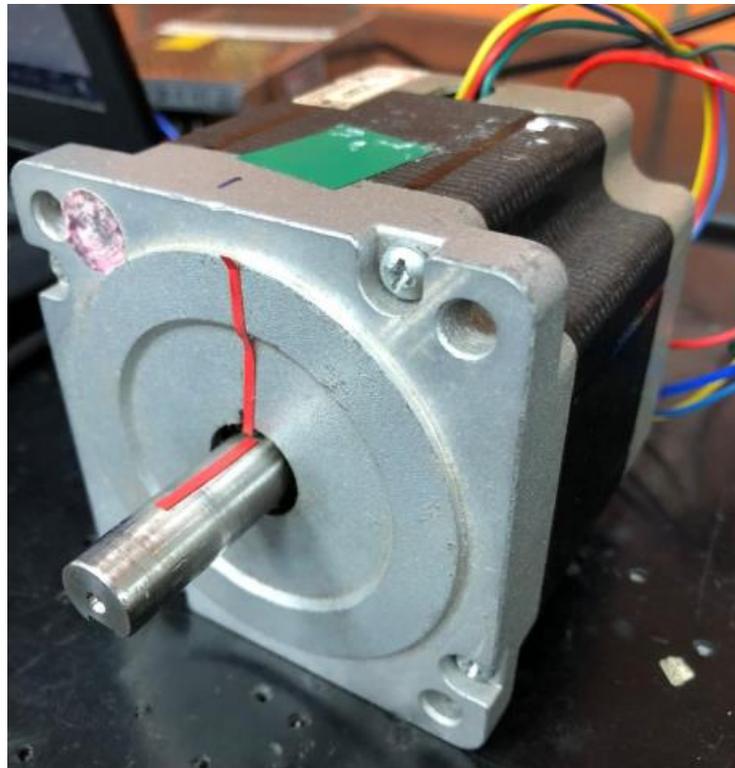
Finalmente com a realização dos cálculos temos uma força de 17,77N já com o fator de segurança de 30%. Para o cálculo do torque necessário foi utilizado a equação de momento conforme Chapman (2013).

$$T = Fd \quad (5)$$

Onde T é o torque e d é a distância do braço de alavanca do momento realizado.

Considerando os requisitos e a aplicação das equações, o torque necessário para a aplicação é de 4,44Nm, portanto, o motor utilizado foi o AK34/52F4CN1.8 da marca Akiyama (Figura 22). Por se tratar de um motor disponível para reutilização e atender ao requisito de torque.

Figura 22 – Motor de passo Akiyama 52kgf.cm padrão NEMA 34.



Fonte: Autor (2021).

Este motor de passo possui o padrão de medidas NEMA 34 e suas características técnicas são mencionadas no Quadro 1.

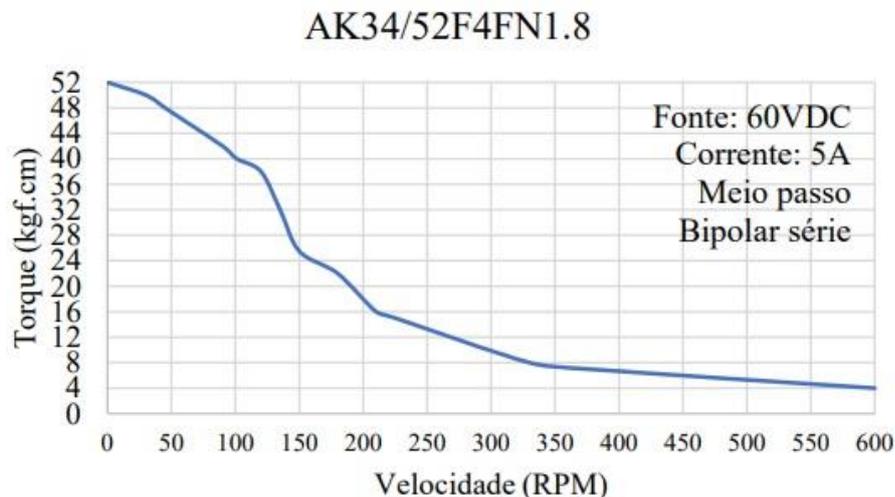
Quadro 1 – Especificações técnicas do motor de passo AK34/52F4CN1.8.

Dados	Valores
Torque máximo	52kgf.cm / 5,1Nm
Corrente por fase	5A
Conexão	Bipolar
Quantidade de fios	4
Ângulo por passo	1,8
Peso	2,3kg

Fonte: Neomotion (2021).

Uma informação indispensável para qualquer motor é sua curva de torque em relação a velocidade. Para o motor de passo em questão, a curva que descreve sua performance é mostrada na Figura 23.

Figura 23 – Curva de torque motor de passo 52kgf.cm.



Fonte: Neomotion (2021).

Considerando uma redução de 3 para 1 devido ao redutor planetário que será explicado na próxima subseção, o motor de passo operou numa velocidade de 150rpm, em 0,06833m/s ou 4,1m/min em sua velocidade linear, atendendo assim o requisito de velocidade de saída de 50rpm.

O motor de passo em questão possui 4 fios de alimentação, a identificação dos fios é importante para a ligação correta no driver acionador, caso as ligações não

estejam de acordo o motor de passo não é acionado, portanto, os fios possuem as seguintes identificações conforme o Quadro 2.

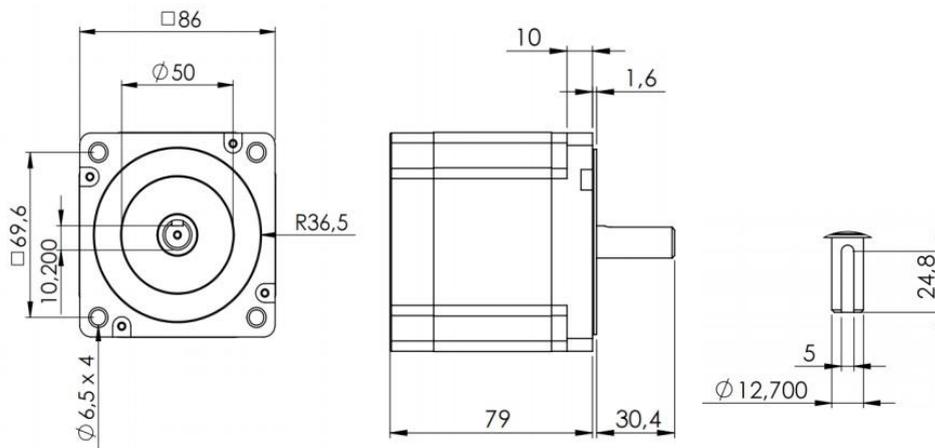
Quadro 2 – Identificação dos fios do motor de passo utilizado

Fio	Terminal Driver
Vermelho	A+
Verde	A-
Amarelo	B+
Azul	B-

Fonte: Neomotion (2021).

Como mencionado, o motor de passo de 52kgf.cm da Akiyama utiliza as dimensões padrões NEMA 34. Seu eixo tem 12,7mm de diâmetro e sua furação de fixação com 98,425mm de diâmetro conforme Figura 24.

Figura 24 – Desenho técnico motor de passo 52kgf.cm.



Fonte: Neomotion (2021).

3.2.2 Redutor planetário

O redutor planetário possui diversas funções em uma aplicação, além de reduzir a velocidade, ele aumenta na mesma proporção o torque gerado pelo motor com uma alta eficiência. Optou-se por utilizar um redutor planetário pelas seguintes questões:

- Aliviar o torque gerado no motor de passo e conseqüentemente melhorar a relação de inércia do sistema;
- Aumentar a resistência estrutural do sistema justamente pelo fato de o redutor suportar maiores esforços radiais e axiais;

O modelo de redutor planetário utilizado foi o PE2-070-003 da marca *Apex Dynamics* (Figura 25). Este redutor que é compatível com o motor de passo que foi utilizado, possui uma redução de 3 para 1. Portanto, o torque necessário da aplicação que era de 4,44Nm, com o uso do redutor cai para 1,48Nm. A velocidade também é afetada, um dos requisitos é a velocidade de saída à 50RPM, logo, o motor de passo teve de operar em 150RPM.

Figura 25 – Redutor planetário Apex modelo PE2-070-003.



Fonte: Autor (2021).

As características técnicas do redutor planetário utilizado podem ser verificadas no quadro:

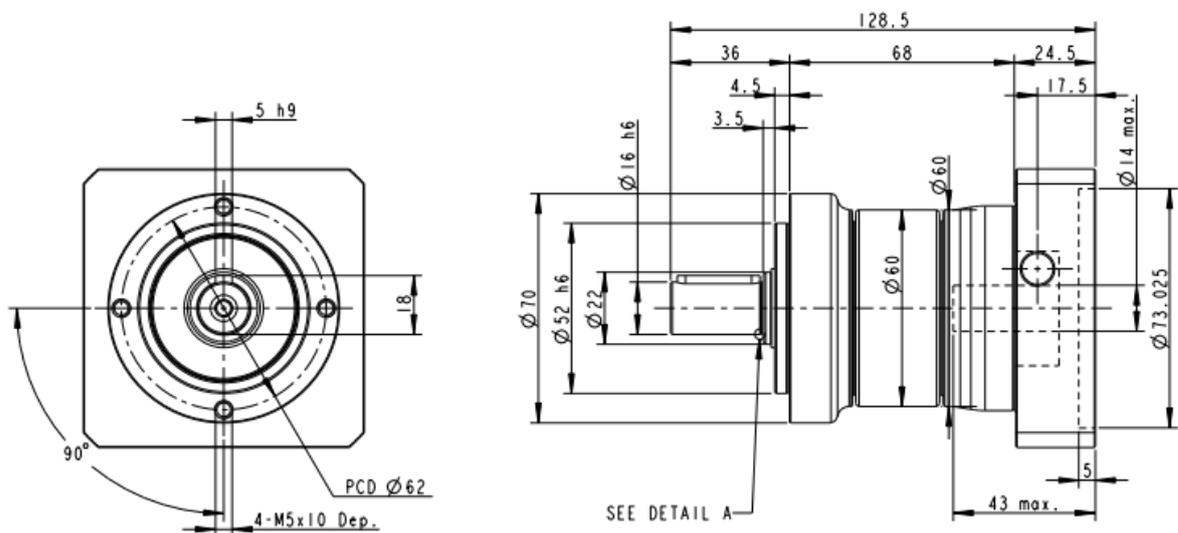
Quadro 3 – Especificações técnicas do redutor utilizado modelo PE2-070-003.

Informações técnicas	Valores
Torque nominal de trabalho	42Nm
Torque máximo de trabalho	75,6Nm
Força máxima radial no eixo	1150N
Força máxima axial no eixo	575N
Eficiência	97%
Peso	1,2kg

Fonte: Apex Dynamics (2021)

Utilizando as furações de saída do redutor para fixar o dispositivo de medição, é possível garantir uma estrutura rígida o suficiente para a operação. O redutor planetário utilizado possui um eixo de saída de 16mm de diâmetro com furação de fixação padrão M5 de 62mm entre furos detalhado na Figura 26.

Figura 26 – Desenho técnico redutor planetário Apex utilizado.



Fonte: Autor (2021).

3.2.3 Mola

A mola utilizada para realizar a compressão entre as duas polias, gerando assim atrito suficiente para o tracionamento do cabo desejado, possui um tamanho aproximado de 54 milímetros, conforme a Figura 27. A aplicação desta mola é no sistema de freio dianteiro de uma motocicleta convencional.

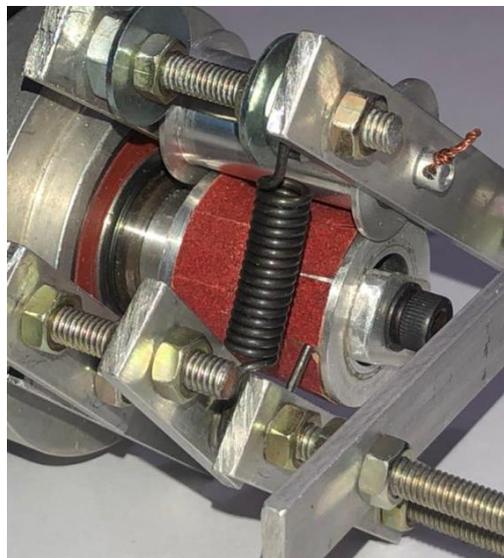
Figura 27 – Mola de compressão utilizada.



Fonte: Autor (2021).

Esta mola realiza a conexão entre as hastes da Polia 1 e a estrutura do medidor de cabo que é fixada na carcaça do redutor planetário, que pode ser verificada na Figura 28.

Figura 28 – Mola de compressão instalada no dispositivo de medição.



Fonte: Autor (2021).

3.2.4 Mancais

Os materiais utilizados para a montagem dos mancais da base de rolagem do carretel de cabo foram dois rolamentos, dois suportes de alumínio que possuem alojamento para rolamentos e um eixo central para interligação dos rolamentos.

As dimensões dos rolamentos são de 35mm de diâmetro externo, 17mm de diâmetro interno com uma espessura de 10mm, tal qual pode ser visualizado na Figura 29.

Figura 29 – Rolamentos utilizados na mancalização



Fonte: Autor (2021).

Os suportes para realizar o alojamento dos rolamentos possuem 75mm de comprimento, 43mm de largura e 8mm de espessura sem considerar o ressalto do alojamento, conforme a Figura 30.

Figura 30 – Suportes para os rolamentos.



Fonte: Autor (2021).

Para a interligação entre os rolamentos, utilizou-se um eixo manufaturado com 17mm de diâmetro em suas duas extremidades para o encaixe no interior dos rolamentos, e um ressalto em seu corpo servindo como limitador. Sua estrutura pode ser conferida na Figura 31.

Figura 31 – Eixo central para a mancalização.



Fonte: Autor (2021).

3.2.5 Polias

Como parte do desenvolvimento, foram manufaturadas polias em alumínio. A Polia 1 possui um furo interno de 16 milímetros para o encaixe no redutor planetário, com um diâmetro externo de 28mm sendo este também seu comprimento.

Figura 32 – Polia 1.



Fonte: Autor (2021).

A Polia 2 foi projetada com um diâmetro dos discos externos de 22mm e um comprimento de 31mm, o ponto importante desta polia são seus dois alojamentos para rolamentos em suas laterais para reduzir o atrito entre a polia e o cabo a ser medido.

Figura 33 – Polia 2.

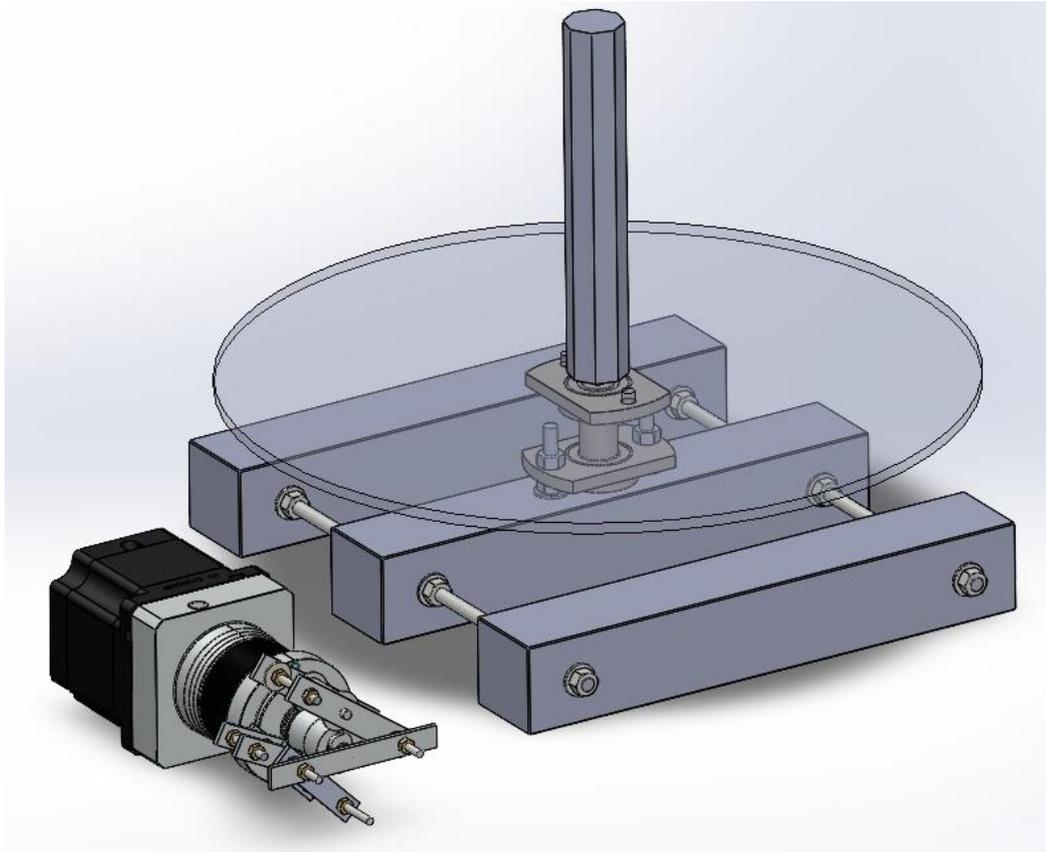


Fonte: Autor (2021).

3.2.6 Projeto Solidworks

Para a realização da montagem da parte mecânica do dispositivo de medição e também da base de rolagem do carretel de cabos, as peças que formam estes dois sistemas foram projetadas utilizando o *software* Solidworks. Desta forma temos prévia de como ficaria o projeto final sem custo monetário. Na Figura 34, é mostrado uma vista completa de todo o sistema mecânico.

Figura 34 – Simulação 3D do sistema mecânico completo.

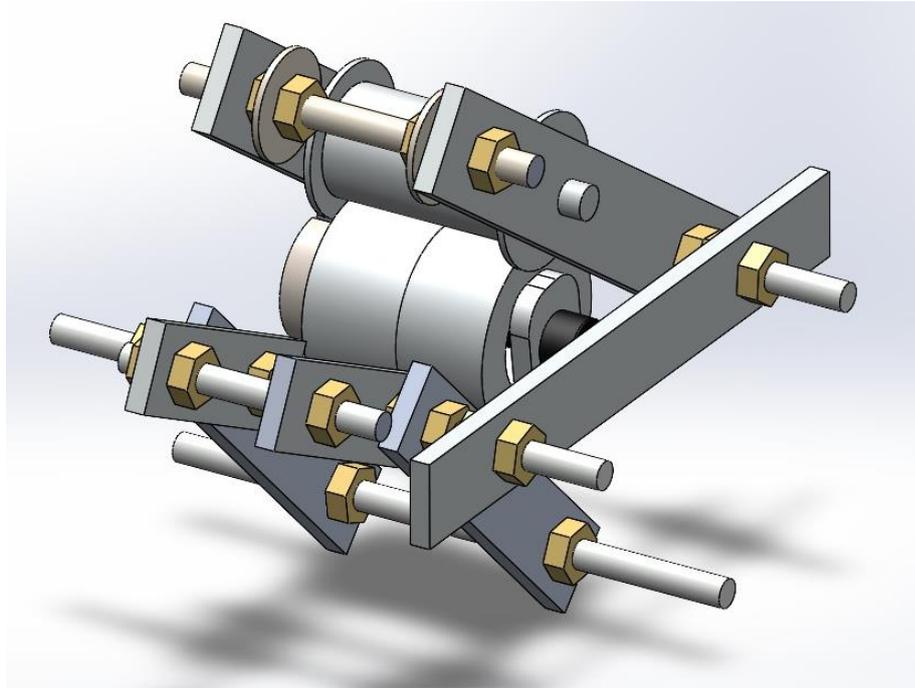


Fonte: Autor (2021).

3.2.6.1 Dispositivo de medição

Para realizar a medição dos condutores elétricos foi tomado como um requisito, uma estrutura simples, fácil de ser montada e com um baixo custo. A metodologia utilizada para cumprir os requisitos mencionados foi por meio de duas polias atuando como cilindros de alimentação. A polia motora, chamada de “Polia 1”, é conectada e mancalizada no próprio eixo do redutor planetário. A “Polia 2” ou polia movida, é mancalizada numa haste móvel, onde por meio da mola, esta Polia 2 é pressionada contra a Polia 1. Com o acionamento do motor de passo, o tracionamento do cabo é efetuado e devido a força da mola, logo, o condutor é mantido em contato com a Polia motora. O dispositivo de medição projetado foi fixado na saída do redutor planetário por meio de duas barras de rosca sem fim. Na Figura 35 se encontra o projeto em Solidworks do dispositivo de medição.

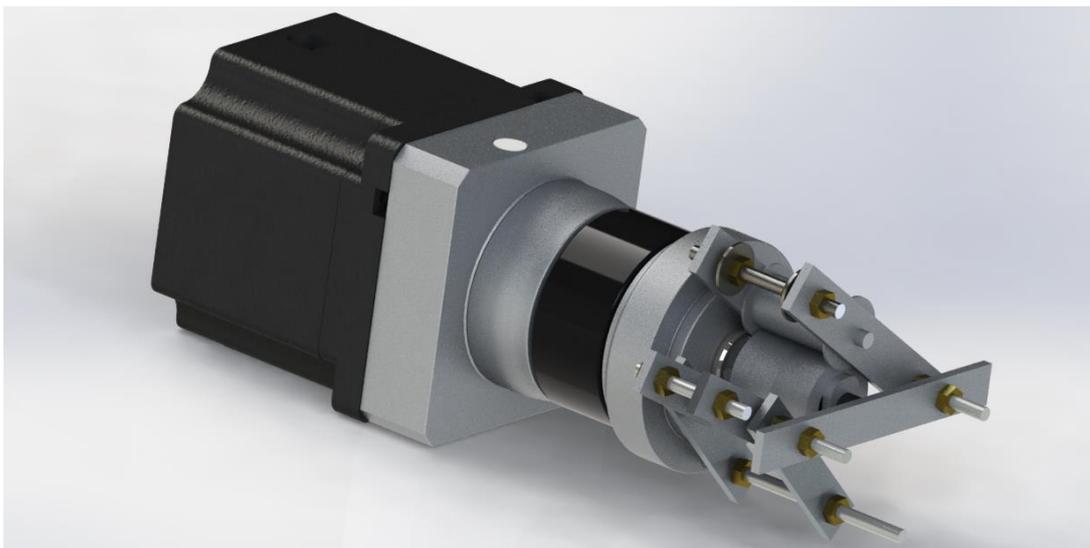
Figura 35 – Dispositivo de medição projeto no *software* Solidworks.



Fonte: Autor (2021).

Foi utilizado a ferramenta de renderização do Solidworks para verificar como ficaria o sistema montado. A aparência final do dispositivo, mostrado na Figura 36 a seguir.

Figura 36 – Renderização do sistema de acionamento e medição.



Fonte: Autor (2021).

Os materiais que foram utilizados para a montagem deste sistema de medição estão listados no Quadro 4.

Quadro 4 – Lista de materiais especificados para a montagem do dispositivo de medição.

MATERIAL	QUANTIDADE
Polia 1	1 unidade
Polia 2	1 unidade
Barra Chata Alumínio 3mm x 15mm	500mm
Barra de rosca sem fim M5	250mm
Porca M5	25 unidades
Eixo Polia 2	1 unidade
Rolamento Polia 2	2 unidades
Suporte limitador Polia 1	1 unidade
Mola	2 unidades

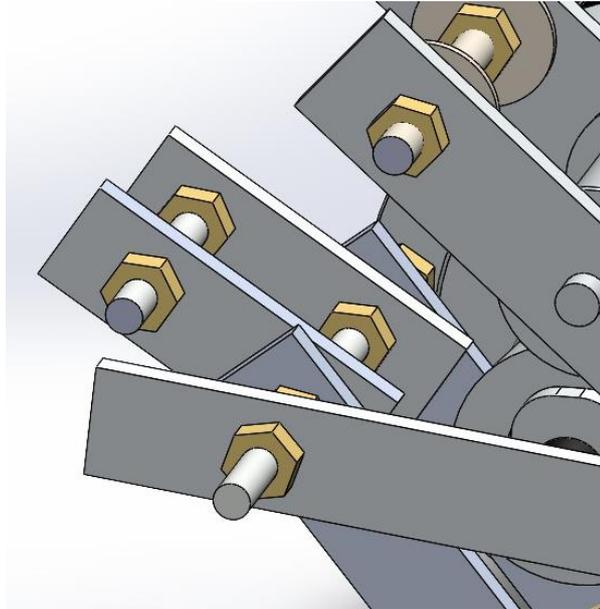
Fonte: Autor (2021).

A Polia 1 foi projetada pensando da praticidade. Esta possui uma cavidade em torno, pois quando acionada consegue manter sempre o cabo em seu centro. O diâmetro onde o fio vai passar pela polia é de 26mm, resultando em um perímetro de 81,68mm. Seu formato pode ser verificado na Figura 32 e suas dimensões no APÊNDICE A – DIMENSÕES POLIA 1. Foi utilizado um suporte para limitar o encaixe da polia no eixo do redutor e garantir que este opere numa posição fixa.

Em relação a Polia 2, foi necessário a mancalização por meio de dois rolamentos para diminuir o atrito e fazer com que a polia movida possa rotacionar o mais livre possível. Sua função no sistema é justamente manter o fio pressionado sobre a Polia 1 por meio das hastes mancalizadas.

Para indicar a posição correta do cabo antes de iniciar a medição, foi projetado uma marcação. Ao inserir o cabo para medição, o mesmo é posicionado de forma perpendicular a esta indicação. Esta marcação de posição correta pode ser conferida da Figura 37.

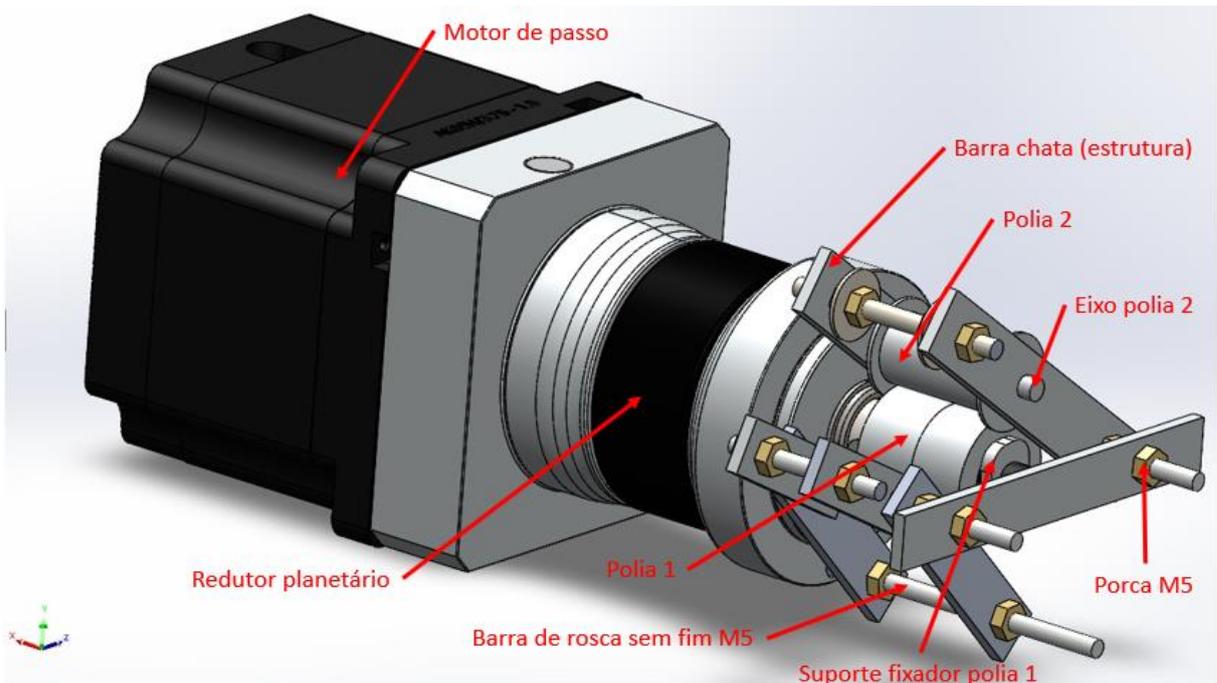
Figura 37 – Marcação de posição correta.



Fonte: Autor (2021).

Foi montado um demonstrativo da disposição de cada item no sistema de medição de condutores, que pode ser verificado na Figura 38 a seguir.

Figura 38 – Disposição legendada de cada item no sistema.



Fonte: Autor (2021).

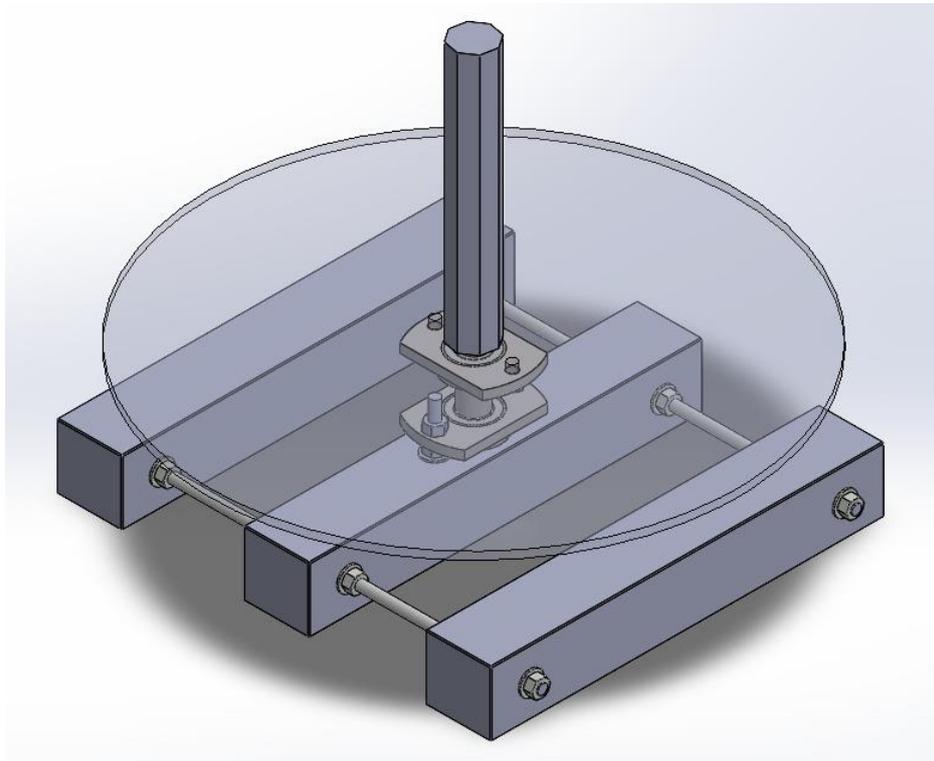
Os desenhos dimensionais dos componentes principais do sistema de medição se encontram no Apêndice A, Apêndice B, Apêndice C, Apêndice D, Apêndice E e Apêndice F.

3.2.6.2 Base de rolagem do carretel

Partindo do mesmo princípio do dispositivo de medição, a base de rolagem do carretel foi projetada de uma forma simples e com uma alta rigidez para suportar o peso disposto sobre.

Para garantir que esta base não tenha resistência para o tracionamento, foi utilizada uma dupla mancalização na base, na Figura 39 é mostrado como ficou o projeto da base completa.

Figura 39 – Base de rolagem projetada.

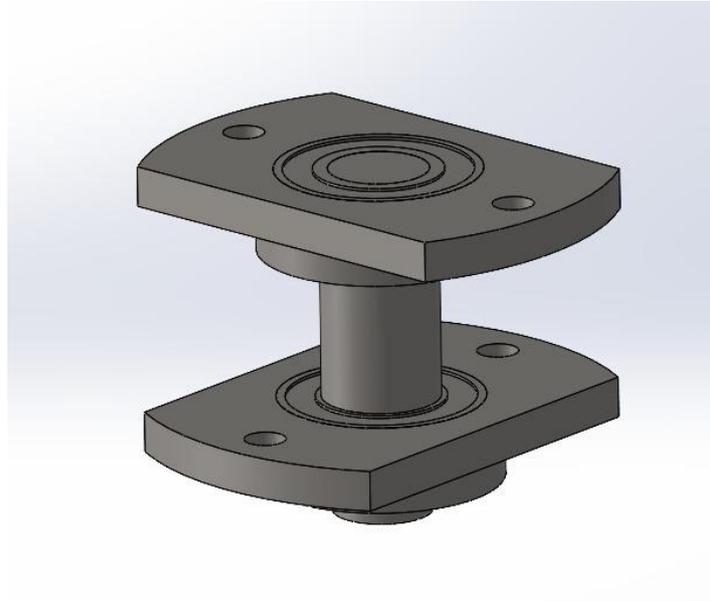


Fonte: Autor (2021).

A dupla mancalização projetada foi realizada a partir de dois suportes para dois rolamentos e um eixo central para realizar o movimento rotacional. O suporte inferior

fica fixo na estrutura do sistema enquanto o suporte superior consegue se movimentar livremente.

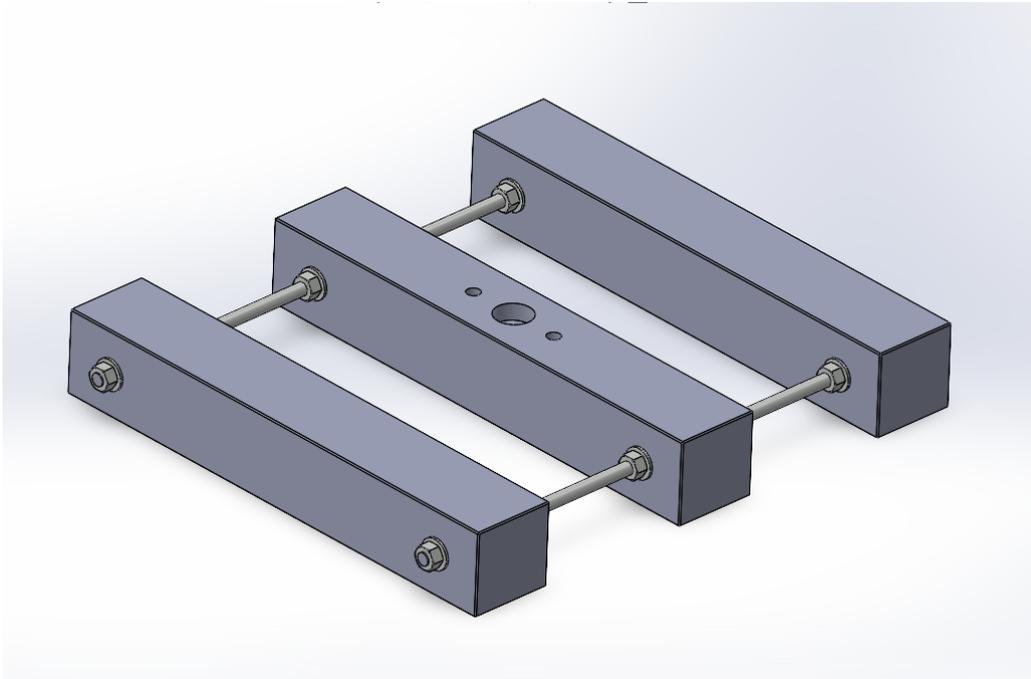
Figura 40 – Projeto da mancalização dupla.



Fonte: Autor (2021).

Em um primeiro momento a construção da base com perfis de aço foi considerada. Porém, com o alto preço do metal durante o período de execução do projeto, se tornou inviável a utilização desse material. Buscando um material com menor custo e facilmente customizável, a estrutura deste sistema foi construída utilizando madeira maciça de 50x50 milímetros, barras de rosca sem fim foram utilizadas para a interligação e criar rigidez na estrutura, uma forma simples, funcional e de baixo custo conforme a Figura 41.

Figura 41 – Estrutura da base de rolagem dos carretéis feita em madeira e barras de rosca sem fim.



Fonte: Autor (2021).

3.3 ESTRUTURA LÓGICA

Nesta seção será explicado qual a lógica utilizada para a realização da medição dos condutores elétricos, e o código desenvolvido para IDE Arduino.

3.3.1 Lógica de medição

A lógica adotada para a medição dos cabos é simples, dependendo apenas do diâmetro de trabalho da Polia 1 e da resolução configurada no driver de motor de passo.

Foi explicado nas subseções anteriores, que a resolução (PPR) adotada no driver de motor de passo foi de 400 pulsos por revolução e o diâmetro de trabalho da Polia 1 que foi fixada no eixo do redutor planetário de 26mm de diâmetro.

Conhecendo estas informações, primeiramente foi calculado o perímetro de trabalho da Polia 1 utilizando a Equação (6) para cálculo de perímetro de um círculo. Desta forma, o perímetro de trabalho obtido é de 81,68mm.

$$p_{polia1} = 2\pi r \quad (6)$$

Sendo p_{polia1} o perímetro de trabalho da Polia 1 em milímetros; e r sendo o raio de trabalho da Polia 1 em milímetros.

Com o perímetro conhecido e por meio de uma regra de três simples, podemos calcular a quantidade de pulsos necessários para uma determinada quantidade de fio requerida. A Equação (7) determina esta relação das variáveis.

$$P_{total} = \frac{x_{mm} PPR i}{p_{polia1}} \quad (7)$$

Sendo P_{total} a quantidade de pulsos total para a quantidade de cabo desejada; x_{mm} a quantidade de cabo desejada pelo usuário em milímetros; PPR a resolução do driver ou pulsos por revolução, e i sendo a relação de redução a ser utilizada.

A quantidade de cabo desejada é setada pelo usuário pela interface disponibilizada pelo LCD *Keypad Shield*, logo, temos todos os parâmetros necessários para a obtenção da quantidade de pulsos relacionados a quantidade de cabo requerida.

3.3.2 Programação

Nesta subseção será explicado quais foram as funções implementadas, destacando seus pontos importantes do código e da Interface Homem-Máquina adotada.

Foi desenvolvido dois modos para este projeto que podem ser escolhidos ao iniciar o Arduino, são eles:

- Modo COMPRIMENTO: onde o usuário pode selecionar a quantidade de cabo desejada;
- Modo LIVRE: onde o sistema é acionado livremente, mostrando a quantidade de cabo medida em tempo real e ao pressionar o botão *select*, a tarefa é finalizada.

Na Figura 42 pode ser verificado o menu de seleção de modo diretamente implementado no Arduino Uno com o LCD *Keypad Shield*.

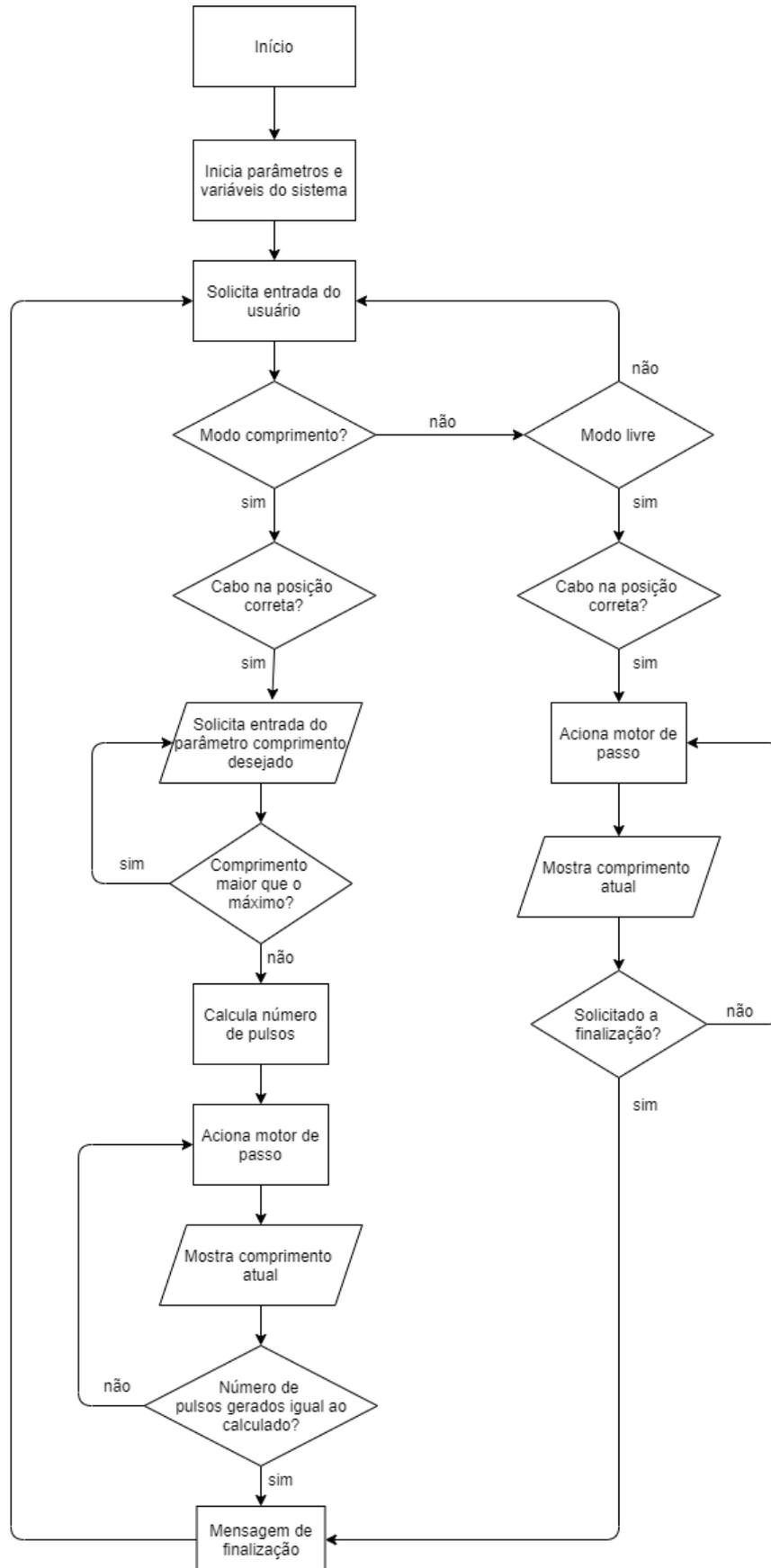
Figura 42 – Menu de escolha inicial.



Fonte: Autor (2021).

Um fluxograma foi desenvolvido para facilitar a compreensão do código desenvolvido. Este que pode ser verificado na Figura 43.

Figura 43 – Fluxograma demonstrativo do código desenvolvido.



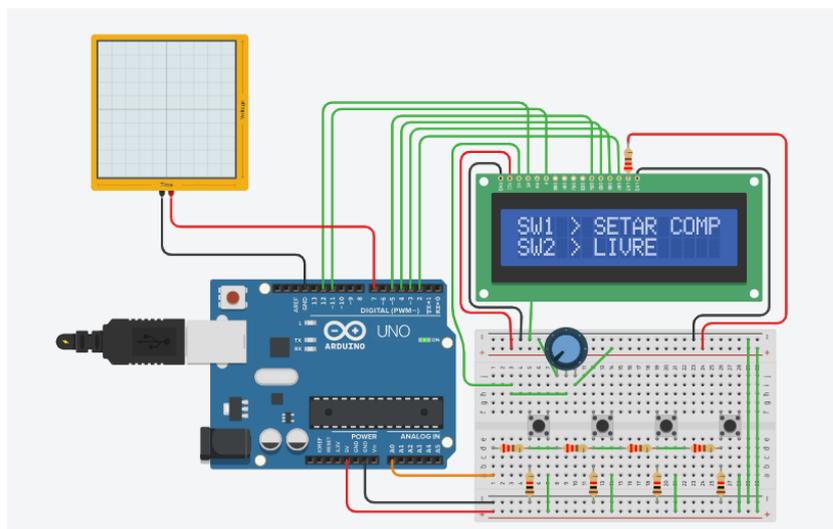
Fonte: Autor (2021).

O código completo do projeto desenvolvido para a IDE Arduino pode ser conferido no Apêndice G.

3.3.2.1 Simulação Tinkercad

Para a validação da programação desenvolvida nas primeiras versões, foi utilizado a ferramenta Tinkercad para a simulação do Arduino já com o display 16x2. Esta ferramenta permite o acionamento do Arduino com o código feito, testando as entradas e saídas com a ajuda de osciloscópio, tudo de forma virtual sem riscos de faltar ou queimar componentes. Tinkercad foi uma ferramenta essencial para o desenvolvimento preciso e sem custos adicionais do protótipo.

Figura 44 – Simulação e validação da estrutura lógica no *software* Tinkercad.



Fonte: Autor (2021).

Ambas as funções funcionaram conforme o esperado, gerando os pulsos corretos para o driver de motor de passo.

4 RESULTADOS

Esta seção tratará dos resultados obtidos no projeto do protótipo, levando em consideração os objetivos específicos citados na introdução.

4.1 PROTÓTIPO

A montagem do protótipo e da base giratória para carretéis seguiu fielmente o projeto desenvolvido via *software* Solidworks. Na Figura 45 é possível verificar como ficou o estado final do sistema de medição.

Figura 45 – Montagem final do sistema de medição.



Fonte: Autor (2021).

Algumas otimizações foram necessárias para um melhor desempenho do protótipo como por exemplo a inclusão de uma barra de rosca sem fim na estrutura

da polia 2, desta forma foi necessário a utilização de apenas uma mola de compressão para o tracionamento do cabo.

Figura 46 – Barra de rosca sem fim adicionada para otimização do sistema mecânico.



Fonte: Autor (2021).

A base de rolagem para os carretéis cumpriu com sua finalidade de aliviar o atrito no tracionamento dos cabos diretamente do carretel onde o cabo se encontra e sua estrutura final pode ser conferida da Figura 47 a seguir.

Figura 47 – Base giratória para carretéis finalizada.



Fonte: Autor (2021).

Sua estrutura de mancalização por meio de dois rolamentos conectados por um eixo central, pode ser verificado na Figura 48.

Figura 48 – Mancalização da base giratória finalizada.



Fonte: Autor (2021).

4.2 MEDIÇÕES

O primeiro teste de medição dos cabos foi realizado utilizando uma velocidade no motor de passo de 51rpm, resultando numa velocidade de 17rpm na saída do redutor.

Foi considerando como entrada cinco comprimentos de cabos, sendo os seguintes tamanhos indicados:

- 50mm;
- 100mm
- 200mm;
- 220mm;
- 300mm.

Em um segundo teste utilizando o sistema completo (motor de passo, redutor planetário e sistema de medição), os seguintes comprimentos foram dados como entrada para o sistema, conforme a Tabela 2 onde também é mostrado os resultados obtidos.

Tabela 2 – Entradas e resultados obtidos na realização do segundo teste.

Entrada [mm]	Resultado [mm]
50	51
75	76
100	101
125	126
167	169
201	203
235	238
260	262
300	304
350	354

Fonte: Autor (2021).

Foi realizado um terceiro teste de medição utilizando como entrada valores maiores, verificando assim como o sistema se comporta para um comprimento maior. Para este teste quatro entradas foram selecionadas conforme a Tabela 3 abaixo, a coleta dos dados foi feita medindo o tamanho resultante do cabo e também o tempo de medição para cada caso.

Tabela 3 – Terceiro teste de medição de cabos.

Entrada [mm]	Resultado [mm]	Tempo [s]
400	405	35
500	510	44
1000	1030	89
2000	2040	176

Fonte: Autor (2021).

Com base nos resultados obtidos nos testes para cabos de até 350mm de comprimento o erro encontrado foi em média 2,5 mm. Já para cabos com comprimentos de até 2000mm o erro encontrado foi em média 27mm. Esta informação de erro de medição acaba não afetando no resultado final pois é importante que o cabo esteja com o comprimento acima do solicitado. Ao realizar a montagem dos conectores em ambas as extremidades do cabo é perdido uma parte do cabo ao desencapá-lo.

A princípio pensou-se que a origem do erro de medição encontrado foi devido ao cálculo dos pulsos necessários em relação ao comprimento de cabo desejado, verificando este cálculo observou-se que a diferença no número de pulsos obtido entre o cálculo manual e código do programa desenvolvido no Arduino foi apenas na casa decimal. O programa arredondou o valor de pulsos para um número inteiro, desconsiderando a casa decimal. Portanto, a origem do erro no comprimento do cabo não foi devido ao cálculo realizado no código. Esta divergência na medição possui um erro em média de 1mm a cada 50mm medidos.

Conclui-se, portanto, que o erro no comprimento dos cabos é devido a limitação de hardware encontrada e que será explicada na subseção 4.4.

4.3 CUSTO

Em uma primeira avaliação sobre o custo do projeto, podemos avaliar qual seria o valor total caso existisse a necessidade da compra de todos os itens utilizados no projeto em questão.

Conforme a Tabela 4, é possível observar que o custo total previsto considerando a compra de todos os itens do projeto seria de R\$4.504,72, um valor consideravelmente alto levando em conta que um dos objetivos específicos é visar o menor custo utilizando materiais segregados.

Tabela 4 – Custo dos itens e total considerando a possibilidade de compra de todos os itens do projeto.

Item	Custo
Polia 1	R\$50,00
Polia 2	R\$50,00
Eixo Polia 2	R\$20,00
Suporte Polia 1	R30,00
Eixo base giratória	R\$30,00
Motor de passo	R\$512,33
Driver motor de passo	R\$750,00
Fonte chaveada	R\$370,00
Redutor planetário	R\$2.367,39
Componentes estruturais	R\$30,00
Arduino Uno	R\$35,00
Fonte Arduino	R\$20,00
LCD <i>Keypad Shield</i>	R\$30,00
Mola	R\$10,00
Mancais rolamentados	R\$200,00
TOTAL	R\$4.504,72

Fonte: Autor (2021).

Em uma segunda avaliação, pode-se observar na Tabela 5 qual foi o valor total investido no projeto como um todo. Vários itens, foram reutilizados do estoque de segregação da empresa de automação.

Tabela 5 – Custo dos itens e total aplicado no projeto.

Item	Custo
Polia 1	R\$50,00
Polia 2	R\$50,00
Eixo Polia 2	R\$20,00
Suporte Polia 1	R30,00
Eixo base giratória	R\$30,00
Motor de passo	R\$0,00
Driver motor de passo	R\$0,00
Fonte chaveada	R\$0,00
Redutor planetário	R\$0,00
Componentes estruturais	R\$30,00
Arduino Uno	R\$0,00
Fonte Arduino	R\$20,00
LCD <i>Keypad Shield</i>	R\$30,00
Mola	R\$0,00
Mancais rolamentados	R\$0,00
TOTAL	R\$260,00

Fonte: Autor (2021).

O valor final foi de R\$260,00, ou seja, foi economizado 94,23% do valor total previsto considerando a compra de todos os itens em questão.

4.4 LIMITANTE ENCONTRADA

Uma limitante encontrada foi em relação ao microcontrolador Arduino Uno utilizado para o acionamento do motor de passo. A saída digital utilizada para enviar os pulsos para o driver de motor de passo possui uma frequência máxima de 490Hz. Portanto, esta frequência se torna insuficiente para alcançar a velocidade desejada do motor de passo de 150 rpm. Resultando em uma velocidade máxima de aproximadamente 50RPM no eixo do motor de passo, e na saída do redutor uma velocidade máxima de aproximadamente 16rpm.

Apesar da limitante de velocidade encontrada, o protótipo não deixou de cumprir seus objetivos.

Uma possível solução para resolver a limitação de velocidade é a utilização de um CLP (Controlador Lógico Programável), cuja frequência do sinal de saída é configurável. Desta forma, é possível a utilização do motor de passo em sua velocidade máxima de 600rpm, conforme sua curva de torque x velocidade (Figura 23). Outra vantagem ao utilizar este *hardware* é a disponibilidade do mesmo, que é encontrado como item de segregação na empresa de automação citada.

5 CONCLUSÃO

Pode-se concluir com este experimento a importância de um projeto voltado à área da automação de uma atividade anteriormente realizada de forma manual, por meio da programação do Arduino Uno juntamente com o desenvolvimento da estrutura mecânica e eletrônica, foi possível alcançar os objetivos propostos.

O desenvolvimento das estruturas deste projeto forneceu uma base sólida dos principais assuntos relacionados à automação de atividades e também um conhecimento aprofundado do *hardware* em que a empresa de automação comercializa.

O objetivo específico de projetar um dispositivo mecânico para a medição de comprimento de condutores elétricos foi alcançado após a realização de várias versões do projeto deste dispositivo, até ser encontrado uma estrutura mais compacta e com melhor desempenho.

A base giratória para os carretéis atendeu as expectativas esperadas conforme sua função de reduzir os esforços de tracionamento dos condutores elétricos. Este dispositivo atua como um acessório para o dispositivo de medição.

Conclui-se que devido a programação do controle da placa de desenvolvimento Arduino ser realizada em etapas, resultou em testes e validações mais coerentes influenciando no desenvolvimento da interface homem-máquina, facilitando seu manuseio.

Em relação a precisão do dispositivo, percebe-se que houve uma divergência nas medições de aproximadamente 1mm a cada 50mm medidos, porém, este valor atendeu de forma satisfatória nos resultados.

Com este experimento foi possível encontrar uma característica limitante por parte da placa de desenvolvimento Arduino Uno, devido à baixa frequência em sua saída digital observou-se que a velocidade angular resultante do motor de passo ficou abaixo do esperado, conseqüentemente o tempo de medição ficou relativamente longo, o que acaba influenciando no tempo de medição para cabos com uma metragem superior.

Também é recomendado a melhoria da base giratória para carretéis, utilizando como elemento de estrutura principal tubos de aço resultando em um melhor alinhamento da mancalização.

Como proposta de melhoria e trabalhos futuros, se propõe a utilização de um Controlador Lógico Programável – CLP que possui saídas digitais com frequências de até 50kHz e conseqüentemente maior velocidade de medição. A utilização de outros *hardwares* disponíveis também é interessante, como por exemplo um *encoder* externo para a realização das medições com maior precisão, fechando a malha do sistema.

REFERÊNCIAS

APEX DYNAMICS (Taiwan). **High Precision Rack and Pinion**. Disponível em: <http://www.apexdyna.com/download/catalog/Catalog%20Rack%20and%20Pinion.pdf>. Acesso em: 12 set. 2021.

APEX DYNAMICS (Taiwan). **PLANETARY GEARBOX NEW GENERATION P-SERIES**. 2019. Disponível em: <http://www.apexdyna.com/download/catalog/PII&PIIR-Eng.pdf>. Acesso em: 06 maio 2021.

BARBI, Ivo. **Eletrônica de potência**: projetos de fontes chaveadas. Edição do autor. Florianópolis: [S.I.], 2001.

BRASIL. INSTITUTO DE AVIAÇÃO CIVIL. **Aviônicos I**: instrumentos e sistemas elétricos. Rio de Janeiro: [S.I.], 2002.

Brites, Felipe Gonçalves; Santos, Vinicius Puga de Almeida. **Motor de Passo**. 2008. Disponível em: <http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>. Acesso em: 24 mar. 2021.

CHAPMAN, Stephen J.. **Fundamentos de Máquinas Elétricas**. 5. ed. Nova Iorque: Bookman, 2013.

Collins, Jack A.; Busby, Henry; Staab, George. **Projeto Mecânico de Elementos de Máquinas**. 2. ed. Rio de Janeiro: LTC, 2010. 1726 p.

CONDIT, Reston; INC., Microchip Technology; Jones, Dr. Douglas W.; Iowa, University Of. **Stepping Motors Fundamentals**. 2014. Disponível em: <http://ww1.microchip.com/downloads/en/appnotes/00907a.pdf>. Acesso em: 13 abr. 2021.

CUNHA, Alessandro. **O que são sistemas embarcados?** Disponível em: https://files.comunidades.net/mutcom/ARTIGO_SIST_EMB.pdf. Acesso em: 18 mar. 2021.

DASSAULT SYSTEMES. **INTRODUÇÃO AO SOLIDWORKS**. 2014. Disponível em: <http://joinville.ifsc.edu.br/~antonio.dias/201702%20DES16304%20Integrado/Apostila%20Solidworks%202014.pdf>. Acesso em: 08 abr. 2021.

ELETROTÉCNICA Geral. Disponível em: https://edisciplinas.usp.br/pluginfile.php/5073796/course/section/5917953/teo_condutores.pdf. Acesso em: 30 mar. 2021.

ESSEL (Brasil). **Polias e correias I**. Disponível em: <https://essel.com.br/cursos/material/01/Manutencao/23manu2.pdf>. Acesso em: 06 maio 2021.

EVANS, Martins; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino em Ação**. Shelter Island: Manning, 2013.

GIL, A.C. **Como Elaborar Projetos de Pesquisa**. 4. Ed. São Paulo: Atlas, 2002. 175 p.

LINO, Paulo Sérgio Costa. **Polias, Correias e Transmissão de Potência**. 2013. Blog Fatos Matemáticos. Disponível em: <https://engenharia360.com/wp-content/uploads/2013/05/PoliaseCorreias.pdf>. Acesso em: 06 maio 2021.

KHURMI, R. S.; GUPTA, J. K.. **Theory of Machines**. Nova Delhi: S. Chand, 2005

MCROBERTS, Michael. **Arduino Básico**. São Paulo: Novatec, 2011.

MEHL, Ewaldo L. M.. **Fontes Chaveadas**. Disponível em: <http://www.eletrica.ufpr.br/mehl/fonteschaveadas.pdf>. Acesso em: 25 mar. 2021.

MONK, Simon. **Programação com Arduino: começando com sketches**. Porto Alegre: Bookman, 2013. 147 p.

MÜLLER, Herbert W.. **EPICYCLIC DRIVE TRAINS: analysis, synthesis, and applications**. Detroit: Wayne State University Press, 1982.

NEOMOTION (Santa Catarina). **DATASHEET DE PRODUTO: motores de passo. MOTORES DE PASSO**. Disponível em: <https://neomotion.com.br/wp-content/uploads/2020/08/AK34-52F4CN1.8.pdf>. Acesso em: 03 maio 2021.

NORTON, Robert L.. **PROJETOS DE MÁQUINAS: uma abordagem integrada**. 4. ed. Porto Alegre: Bookman, 2013.

PAZOS, Fernando. **Automação de Sistemas & Robótica**. Rio de Janeiro: Axcel Books, 2002.

SCHMIDT, Maik. **Arduino - A Quick-Start Guide**. Dallas: Pragmatic Bookshelf, 2011.

SHIGLEY, Joseph E.; MISCHKE, Charles R.; BUDYNAS, Richard G.. **Projeto de Engenharia Mecânica**. 7. ed. Porto Alegre: Bookman, 2005.

STEVAN JR., Sergio Luiz; SILVA, Rodrigo Adamshuk. **AUTOMAÇÃO E INSTRUMENTAÇÃO INDUSTRIAL COM ARDUINO**: teoria e projetos. São Paulo: Érica, 2015. 296 p.

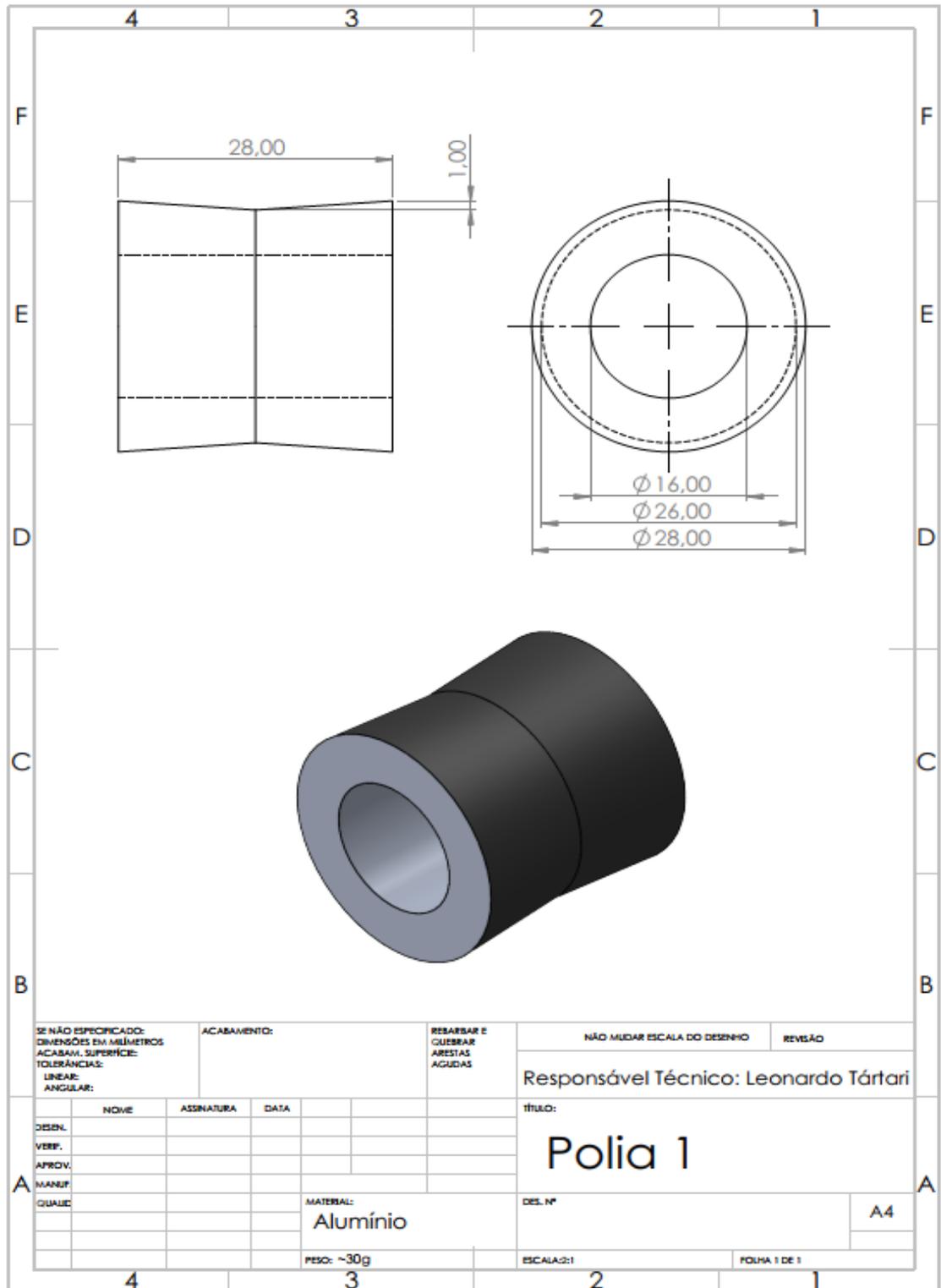
UMANS, Stephen D.. **Máquinas Elétricas de Fitzgerald e Kingsley**: de fitzgerald e kingsley. 7. ed. Porto Alegre: Bookman, 2014.

VERMA, Gaurav; WEBER, Matt. **SolidWorks 2017 Black Book**. Taoru: Cadcamcae, 2016.

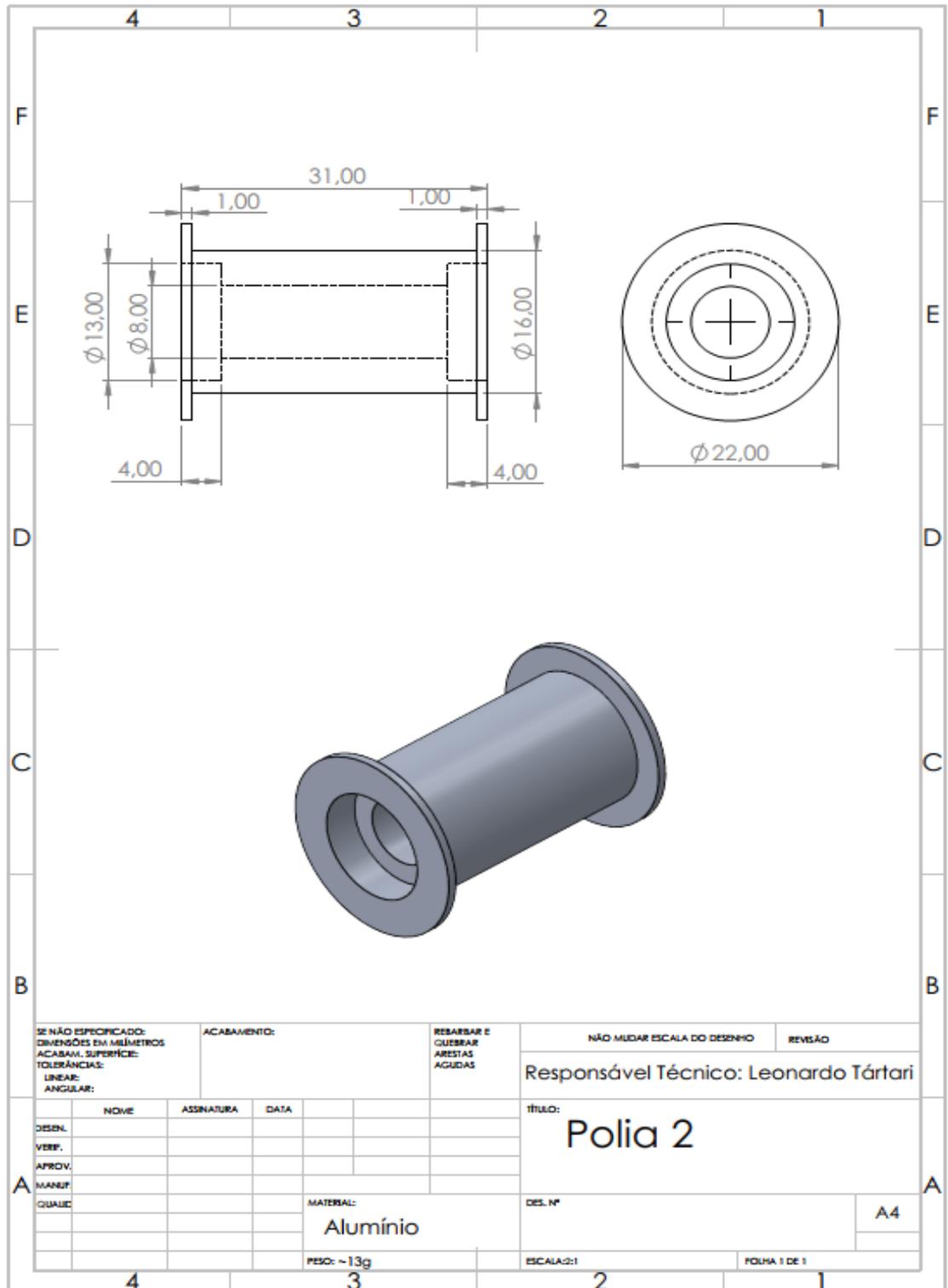
WINDSOR MOLD GROUP (Califórnia). **LEARN HOW TO USE TINKERCAD**. 2020. Disponível em: <https://www.epicentreuwindsor.ca/wp-content/uploads/2020/07/TinkerCad.pdf>. Acesso em: 08 abr. 2021.

YEDAMALE, Padmaraja; CHATTOPADHYAY, Sandip; INC., Microchip Technology. **AN822**. 2002. Disponível em: <http://ww1.microchip.com/downloads/en/appnotes/00822a.pdf>. Acesso em: 25 mar. 2021.

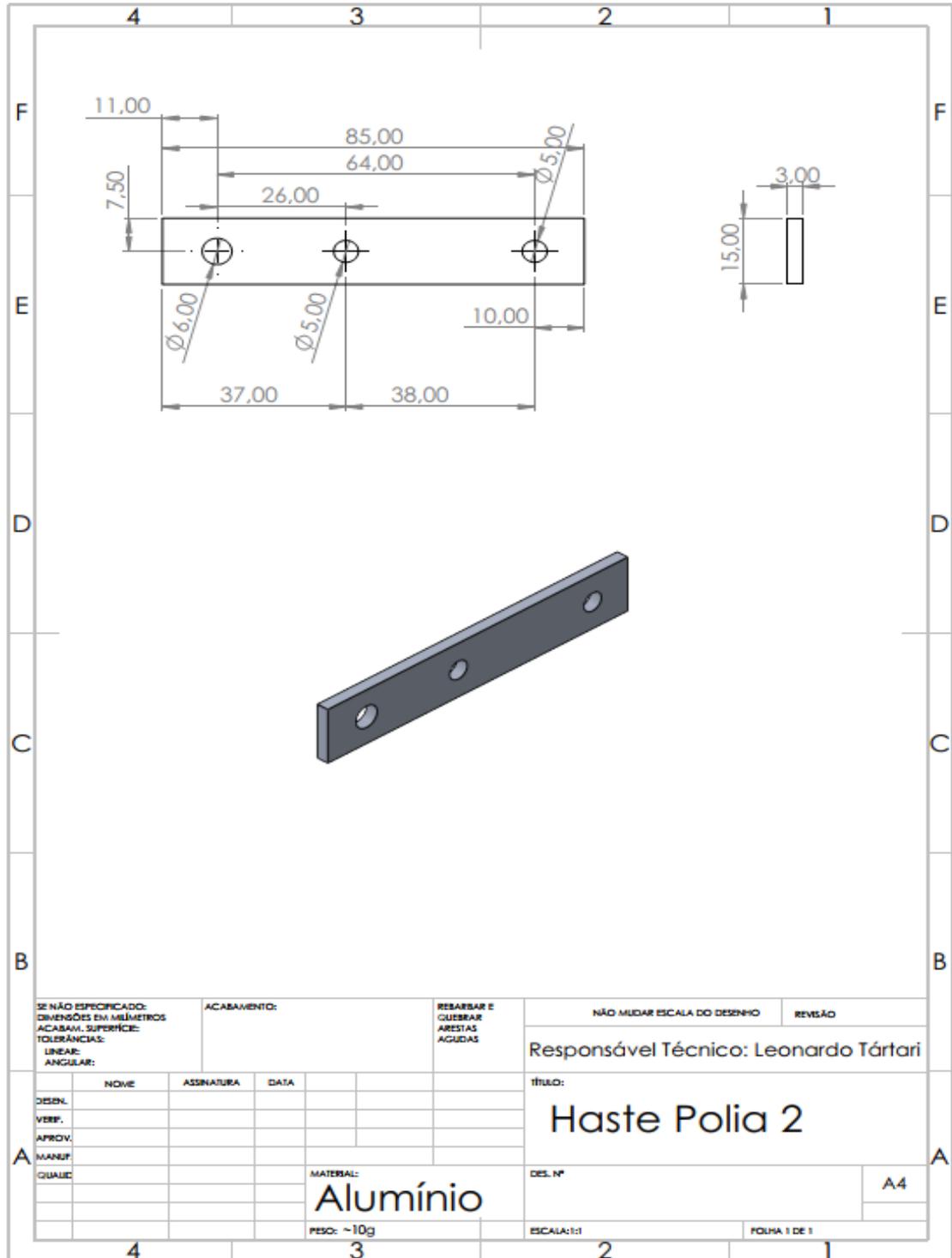
APÊNDICE A – DIMENSÕES POLIA 1



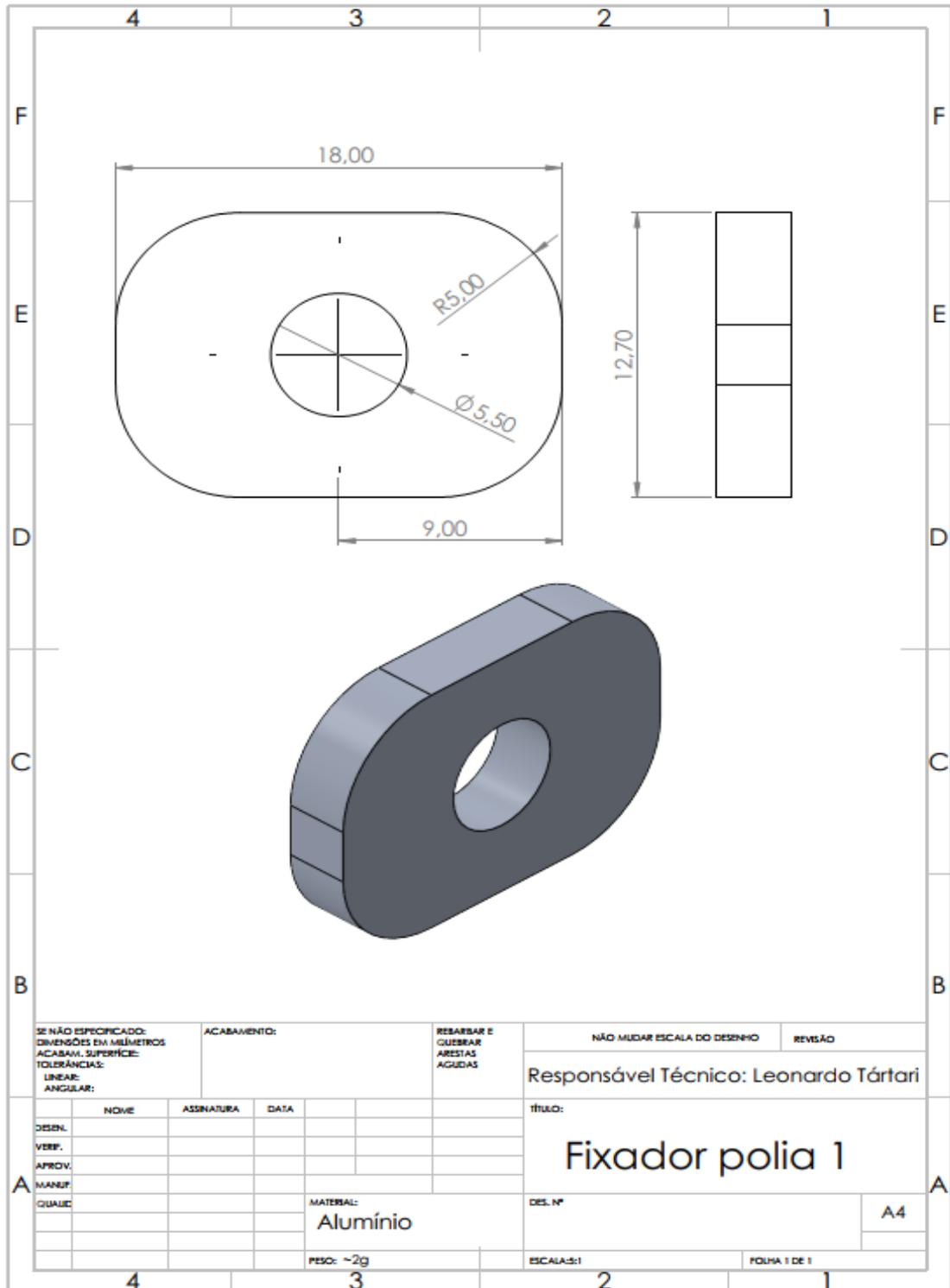
APÊNDICE B – DIMENSÕES POLIA 2



APÊNDICE C – HASTE DE MANCALIZAÇÃO DA POLIA 2 E FIXAÇÃO DA MOLA DE TRACIONAMENTO

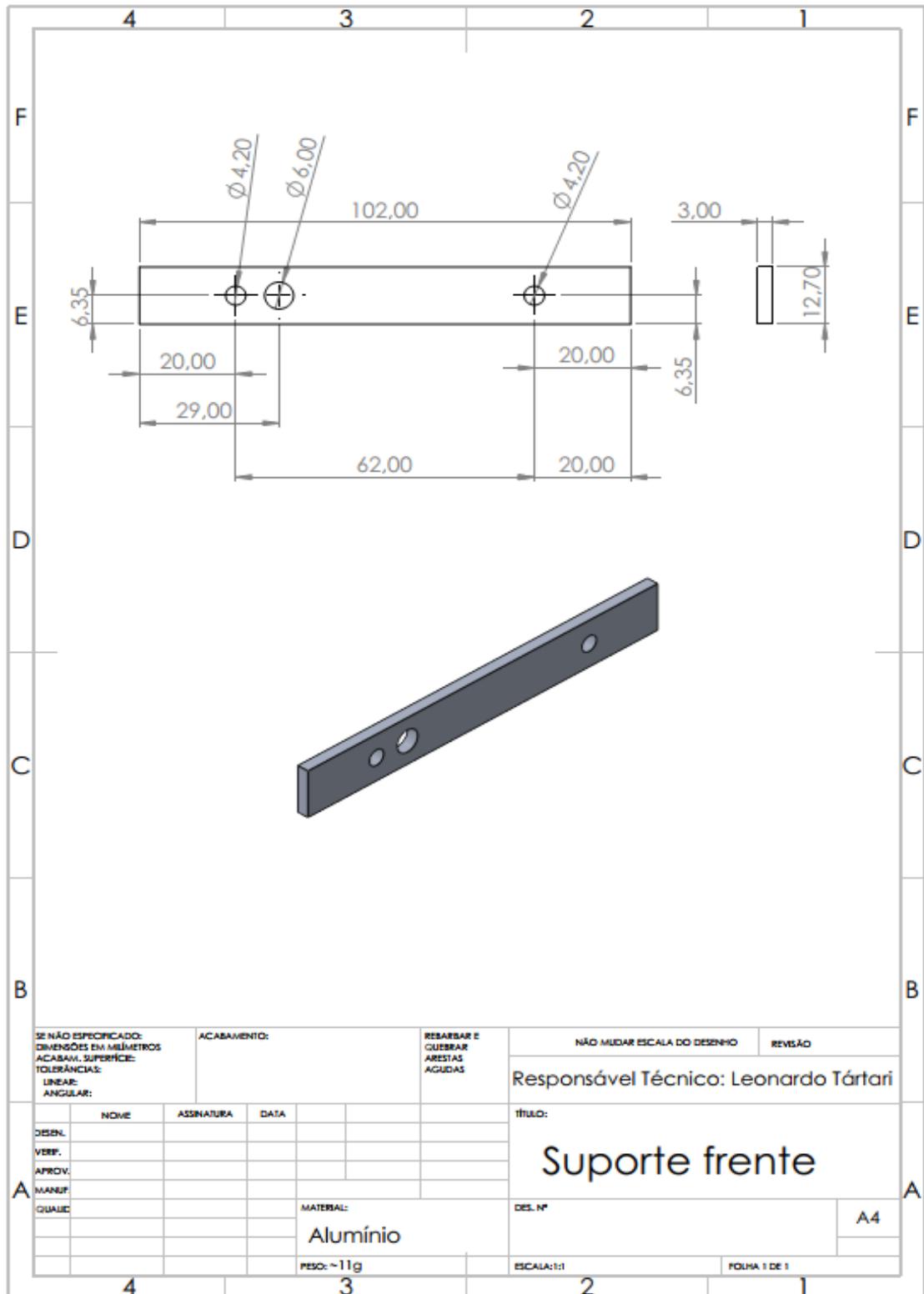


APÊNDICE D – SUPORTE FIXADOR POLIA 1



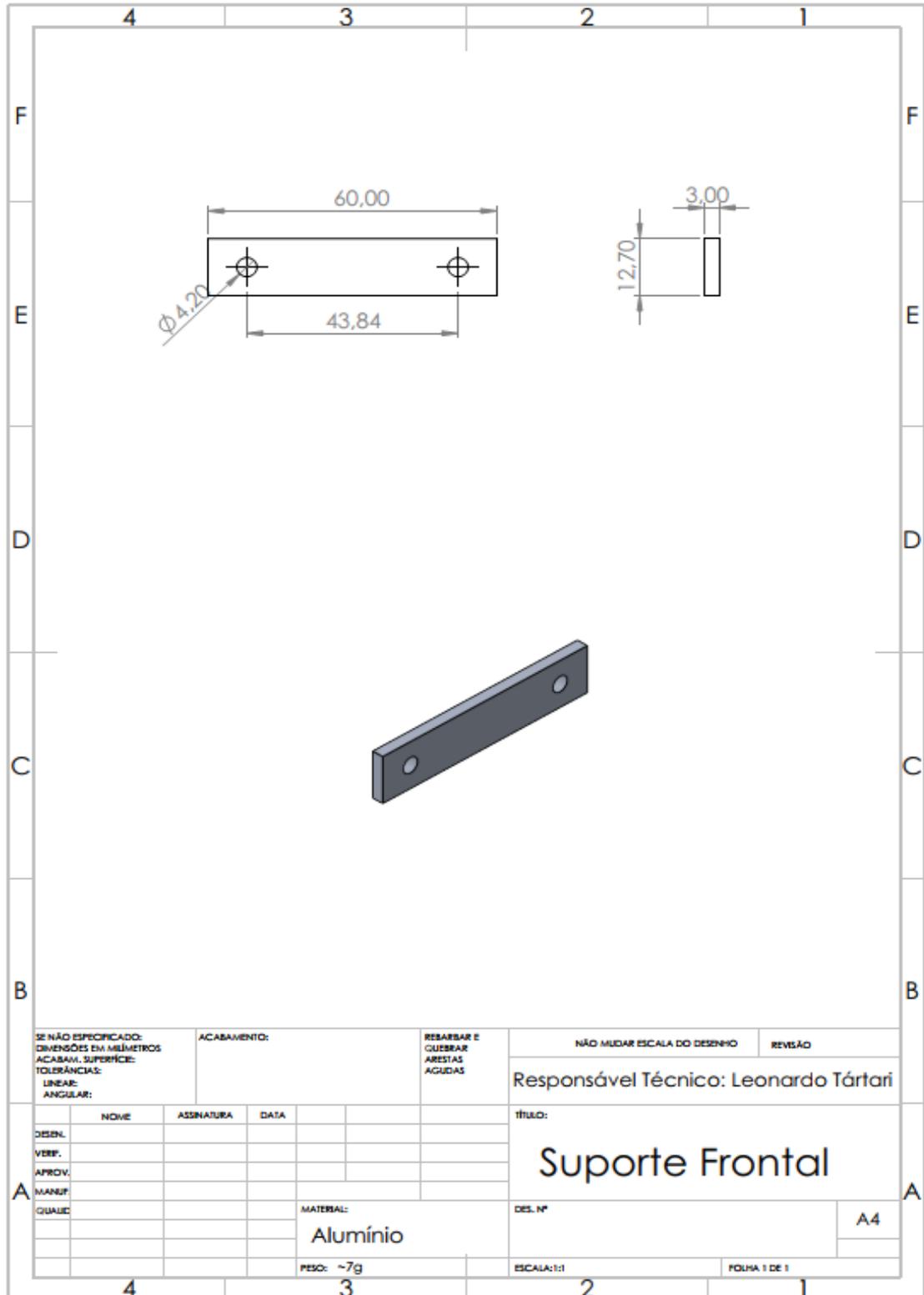
SE NÃO ESPECIFICADO: DIMENSÕES EM MILÍMETROS ACABAM. SUPERFÍCIE: TOLERÂNCIAS: LINEAR: ANGULAR:		ACABAMENTO:	REBARBAR E QUEBRAR ARESTAS AGUDAS	NÃO MUDAR ESCALA DO DESENHO	REVISÃO
				Responsável Técnico: Leonardo Tártari	
DESEN.	NOME	ASSINATURA	DATA	TÍTULO:	
VERIF.				Fixador polia 1	
APROV.					
MANUF.				DES. Nº	A4
QUALIF.				MATERIAL: Alumínio	
				PESO: ~2g	ESCALA: 3:1
					FOLHA 1 DE 1

APÊNCICE E – SUPORTE FRONTAL 1



SE NÃO ESPECIFICADO: DIMENSÕES EM MILÍMETROS ACABAM. SUPERFÍCIE: TOLERÂNCIAS: LINEAR: ANGULAR:		ACABAMENTO:		RESBARBAR E QUEBRAR ARESTAS AGUDAS		NÃO MUDAR ESCALA DO DESENHO		REVISÃO	
						Responsável Técnico: Leonardo Tártari			
						TÍTULO: Suporte frente			
DISEN.		ASSINATURA		DATA		DES. Nº		A4	
VERIF.									
APROV.									
MANUF.									
QUALID.									
				MATERIAL: Alumínio		ESCALA:1:1		FOLHA 1 DE 1	
				PESO: ~11g					

APÊNDICE F – SUPORTE FRONTAL 2



APÊNDICE G – CÓDIGO COMPLETO NA LINGUAGEM C

```

1 #include <LiquidCrystal.h>
2
3
4 #define PUL 11
5 #define DIR 12
6 #define PPR 400
7 #define DIAM_POLIA1 26
8
9
10 int AnalogIn = A0; //porta analogica utilizada para os botoes
11
12
13 //***** INTERFACE *****
14 int sw1 = 640; //SELECT //valor do botao 1
15 int sw2 = 408; //ESQUERDA //valor do botao 2
16 int sw3 = 255; //BAIXO //valor do botao 3
17 int sw4 = 99; //CIMA //valor do botao 4
18 int sw5 = 0; //DIREITA //valor do botao 5
19
20 int status = 0; //estados dos botoes
21
22 bool option1 = 0; //opcao do MODO 1, se = 1
23 bool option2 = 0; //opcao do MODO 2, se = 1
24
25 int pos_correta = 0; //posicionamento do cabo
26
27 int select = 0; //variavel para selecao do modo
28
29
30 //***** DADOS MOTOR DE PASSO *****
31 float comprimento = 0; //variavel com o comprimento desejado
32
33 bool fim = false; //variavel de fim de operacao
34
35 int reducao = 3;
36
37 //***** ACIONAMENTO MOTOR DE PASSO *****
38 long int pulsos = 0; //quantidades de pulsos necessarios
39 int intervalo = 1000; //intervalo em microsegundos dos pulsos
40
41
42 //***** LCD DISPLAY *****
43 LiquidCrystal lcd(8, 9, 4, 5, 6, 7); //seta pinos para o display
44
45
46
47 void setup()
48 {
49     pinMode(PUL, OUTPUT); //pino PUL como saida digital
50     pinMode(DIR, OUTPUT); //pino DIR como saida digital
51
52     Serial.begin(9600); //inicializa o monitor serial

```

```

53   lcd.begin(16, 2);           //inicializa as funções do LCD Display
54
55   digitalWrite(PUL, LOW);    //saida PUL em nivel logico baixo
56   digitalWrite(DIR, LOW);   //saida DIR em nivel logico baixo
57 }
58
59 void loop()
60 {
61   status = analogRead(AnalogIn); //leitura da entrada analogica
62
63   Serial.println(status);      //monitoramento painel serial
64
65   //***** TELA 1 *****
66   if(option1 == false && option2 == false) //TELA DE INICIO
67   {
68       lcd.clear();           //limpa o LCD
69       lcd.setCursor(2, 0);   //cursor na posicao desejada
70       lcd.print("COMPRIMENTO"); //titulo do modo 1
71       lcd.setCursor(2, 1);   //cursor na posicao desejada
72       lcd.print("LIVRE");    //titulo do modo 2
73   }
74
75
76   //***** SELECIONA MODOS *****
77   Func_SelectOption();      //funcao de selecao de modo
78
79
80   //***** TELA 2 *****
81   if(option1 == true) //MODO 1 - SETAR COMPRIMENTO
82   {
83       delay(550);
84       Func_SetComprimento(); //funcao entrada de comprimento
85       delay(600);
86       Func_Run();           //funcao para acionar o motor
87
88       pos_correta = 2;      //finalização
89       fim = true;           //finalização
90   }
91
92
93   if(option2 == true) //MODO 2 - LIVRE
94   {
95       delay(1000);
96       option1 = false;      //seta modo 1 como falso
97
98       Func_ModoLivre();     //funcao para execucao do modo livre
99   }
100
101
102   if(fim == true)          //se fim dos modos
103   {
104       lcd.clear();           //limpa display
105       lcd.setCursor(0, 0);   //reset local do cursor
106       lcd.print("VOLTANDO AO MENU"); //mostra mensagem
107       pos_correta = 0; option1 = false; //reset parametros
108       option2 = false; fim = false; //reset parametros

```

```

109     delay(1000);
110 }
111 delay(50);
112 }
113
114
115
116 /*
117 * Executa a funcao de selecionar o modo desejado
118 * (modo 1 - comprimento ou modo 2 - livre)
119 * Pre: nenhum
120 * Pos: retorna o modo desejado para execucao
121 */
122 void Func_SelectOption(void)
123 {
124     if(status > sw3-10 && status < sw3+10)    //botao para baixo
125     {
126         select = 1;    //valor 1 para indicar o modo selecionado
127     }
128
129     else if(status > sw4-10 && status < sw4+10)    //botao para baixo
130     {
131         select = 0;    //valor 0 para indicar o modo nao selecionado
132     }
133
134     //Caso a variavel auxiliar da interface for 1
135     if(select == 1)
136     {
137         lcd.setCursor(0, 1);
138         lcd.print(">>");
139
140         //printa >> no modo LIVRE
141         lcd.setCursor(0, 0);
142         lcd.print(" ");
143     }
144     else
145     {
146         lcd.setCursor(0, 0);
147         lcd.print(">>");
148
149         //printa >> no modo COMPRIMENTO
150         lcd.setCursor(0, 1);
151         lcd.print(" ");
152     }
153
154     if(status > sw1-10 && status < sw1+10) //selecionado o modo >>
155     {
156         if(select == 0)    //se select for 0,
157         {
158             option1 = true;    //comprimento = option1 = true
159         }
160         else if(select = 1)    //se select for 1,
161         {
162             option2 = true;    //livre = option2 = true
163         }
164     }

```

```

165 }
166
167
168 /*
169 * Executa a funcao de setar o comprimento desejado pelo usuário.
170 * Pre: mostra no display qual o valor atual estabelecido
171 *     para a metragem do cabo.
172 * Pos: confirmação para execução da tarefa de medir o cabo.
173 */
174 int Func_SetComprimento (void)
175 {
176     Func_PosCorreta();           //cabo na posicao correta?
177     delay(500);
178
179     lcd.clear();                 //limpa display
180     lcd.setCursor(0, 0);        //cursor na posicao desejada
181     lcd.print("COMPRIM. EM mm:"); //print descricao da funcao
182
183     int vetor[5] = {0};         //vetor para comprimento do cabo
184     int posicaoCursor = 0;       //controle da posicao do cursor
185
186     lcd.setCursor(0, 1);        //cursor na posicao desejada
187     lcd.cursor();               //underline sob o numero sub o cursor
188
189
190     while(1) //loop para a selecao do comprimento
191     {
192         lcd.setCursor(0, 0);    //cursor na posicao desejada
193         lcd.print("COMPRIM. EM mm:"); //print descricao da funcao
194
195         Func_PrintVetor(vetor, posicaoCursor); //mostra vetor compr.
196         delay(70);
197
198         status = analogRead(AnalogIn); //verifica status analogico
199         Serial.println(status);        //print no monitor serial
200
201         //SELECIONA A POSIÇÃO DO CURSOR - DIREITA ESQUERDA
202         //direita - pressionado botao right - cursor para a direita
203         if(status > sw5-10 && status < sw5+10)
204         {
205             posicaoCursor++;
206
207             if(posicaoCursor == 5) //se esta na ultima posicao
208             {
209                 posicaoCursor = 0; //volta para o inicio
210             }
211             delay(300);
212             lcd.setCursor(posicaoCursor, 1); //move o cursor
213         }
214
215         //esquerda - pressionado botao left - cursor para a esquerda
216         if(status > sw2-10 && status < sw2+10)
217         {
218             posicaoCursor--;
219
220             if(posicaoCursor == -1) //se esta na primeira posicao

```

```

221     {
222         posicaoCursor = 4;    //volta para o ultimo
223     }
224     delay(300);
225     lcd.setCursor(posicaoCursor, 1);    //move o cursor
226 }
227
228
229 //SELECIONA O NUMERO DESEJADO - CIMA BAIIXO
230 //cima - adiciona um se pressionado para cima
231 if(status > sw4-10 && status < sw4+10)
232 {
233     vetor[posicaoCursor] = vetor[posicaoCursor] + 1;//dir.
234
235     if(vetor[posicaoCursor] == 10)    //se ja esta no maximo
236     {
237         vetor[posicaoCursor] = 0;    //volta para 0
238     }
239     delay(300);
240 }
241 //baixo - subtrai um se pressionado para baixo
242 if(status > sw3-15 && status < sw3+15)
243 {
244     vetor[posicaoCursor] = vetor[posicaoCursor] - 1;//esq.
245
246     if(vetor[posicaoCursor] == -1)    //se ja esta no minimo
247     {
248         vetor[posicaoCursor] = 9;    //volta para 9
249     }
250     delay(300);
251 }
252
253
254 //SELECT - continuar/confirmar
255 if(status > sw1-15 && status < sw1+15)
256 {
257     if(vetor[0]*10000 + vetor[1]*1000 +
258     vetor[2]*100 + vetor[3]*10 + vetor[4] > 20000)
259     {
260         for(int i=0; i<5; i++)
261         {
262             vetor[i] = 0;
263         }
264         lcd.clear();    //limpa display
265         lcd.setCursor(0, 0); //cursor na posicao desejada
266         lcd.print("ACIMA DO MAXIMO");
267         lcd.setCursor(0, 1);
268         lcd.print("MAXIMO: 20000");
269         delay(1000);
270         lcd.clear();
271     }
272     else
273     {
274         break;    //sai do loop de selecao de comprimento
275     }
276

```

```

277     }
278
279     }
280     lcd.noCursor(); //desabilita o underline do cursor
281     comprimento = vetor[0]*10000 + vetor[1]*1000 +
282     vetor[2]*100 + vetor[3]*10 + vetor[4]; //vetor para inteiro
283     return comprimento; //retorna o comprimento desejado
284 }
285
286
287
288 /*
289 * Executa a funcao printar o vetor com o tamanho atual desejado.
290 * Pre: O vetor e a posição devem ser válidos
291 * Pos: mostra na tela o estado atual do vetor comprimento.
292 */
293 void Func_PrintVetor(int *vetor, int posicao)
294 {
295     lcd.noCursor(); //desabilita o underline do cursor
296     int i = 0; //variavel auxiliar
297
298     for(i=0; i<5; i++) //print do vetor comprimento
299     {
300         lcd.setCursor(i, 1);
301         lcd.print(vetor[i]);
302     }
303
304     lcd.setCursor(posicao, 1); //seta o cursor na posicao desejada
305     lcd.cursor(); //ativa o underline do cursor
306     return; //retorna
307 }
308
309
310
311 /*
312 * Calcula a quantidade de pulsos que deve ser gerado para o MP
313 * Pre: os parametros devem ser validos e diferentes de 0
314 * Pos: quantidade de pulsos para o acionamento do MP é conhecida
315 */
316 void Func_CalculaPulsos(void)
317 {
318     //int ppr = 400;
319     //int DIAM_POLIA1 = 40;
320     //float pi = 3.1415;
321
322     float perimetro = 3.1415*DIAM_POLIA1; //perimetro da polia
323     float voltas = 0; //variavel voltas necessarias
324     if(comprimento > 50)
325     {
326         voltas = (comprimento-50)/perimetro; //quantidade de voltas
327     }
328     else
329     {
330         voltas = comprimento/perimetro; //quantidade de voltas
331     }
332

```

```

333     pulsos = voltas*PPR*reducao; //pulsos necessarios
334
335     //Serial.println(perimetro);
336     Serial.println(voltas); //print no monitor serial
337     Serial.println(pulsos); //print no monitor serial
338 }
339
340
341 /*
342 * Executa a funcao de acionamento do motor de passo
343 * Pre: a quantidade de pulsos necessária precisa ser valida
344 * Pos: o motor de passo deve ter girado a quantiade de voltas
345 */
346 void Func_Run (void)
347 {
348     //Anti-horario - DIR = LOW
349     //horario - DIR = HIGH
350     int cont = 0; //variavel de controle
351
352     digitalWrite(DIR, HIGH); //SETADO PARA HORARIO
353
354     Func_CalculaPulsos(); //calcula pulsos
355
356     lcd.clear(); //limpa o display
357     lcd.setCursor(0, 0); //move cursor
358
359     while(pulsos > cont) //loop geracao dos pulsos
360     {
361         digitalWrite(PUL, LOW); //PUL em nivel logico baixo
362         delayMicroseconds(intervalo/2); //delay
363         digitalWrite(PUL, HIGH); //PUL em nivel logico alto
364         delayMicroseconds(intervalo/2); //delay
365         cont++; //+1 na variavel de controle
366
367         lcd.setCursor(0, 0); //move cursor
368         lcd.print(cont); //mostra pulsos
369         lcd.print("/");
370         lcd.print(pulsos); //mostra pulsos total
371     }
372     cont = 0; //reseta variavel p/controle
373     pulsos = 0; //reseta pulsos
374     digitalWrite(DIR, LOW); //desativa saida digital DIR
375     digitalWrite(PUL, LOW); //desativa saida digital PUL
376     lcd.clear(); //limpa o display
377     lcd.print("FIM"); //printa a mensagem FIM
378     delay(1000);
379 }
380
381
382
383 int Func_PosCorreta(void)
384 {
385     lcd.clear(); //limpa display
386     lcd.setCursor(0, 0); //seta cursor na posicao desejada
387     lcd.print("POSICAO CORRETA?"); //printa a descricao da funcao
388     lcd.setCursor(0, 1); //seta cursor na posicao desejada

```

```

389     lcd.print("SELECT >> SIM!"); //printa a descricao da funcao
390     delay(200);
391
392     //loop para aguardar a confirmacao pelo usuario
393     while(1)
394     {
395         status = analogRead(AnalogIn); //verifica status analogico
396         if(status > swl-15 && status < swl+15) //acao apertar select
397         {
398             return 1; //sai do loop
399             break;
400         }
401     }
402 }
403
404
405 void Func_ModoLivre (void)
406 {
407     float cont = 0; //variavel de controle
408     float distancia = 0; //variavel distancia
409     float perimetro = 3.141592*DIAM_POLIA1; //perimetro polia 1
410
411     while(1)
412     {
413         if (pos_correta == 0) //verifica posicao
414         {
415             pos_correta = Func_PosCorreta(); //funcao confirmacao pos.
416         }
417
418         if(pos_correta == 1) //se posicao esta correta
419         {
420             lcd.clear(); //limpa display
421             lcd.setCursor(0, 0); //move cursor
422             lcd.print("DISTANCIA [mm]:"); //mostra comprimento
423             digitalWrite(DIR, HIGH); //SETADO PARA HORARIO
424
425             delay(600);
426
427             while(1) //loop modo livre
428             {
429                 digitalWrite(PUL, LOW); //saida PUL em baixa
430                 delayMicroseconds(intervalo/2); //delay
431                 digitalWrite(PUL, HIGH); //saida PUL em alta
432                 delayMicroseconds(intervalo/2); //delay
433                 cont++; //+1 contador
434
435                 distancia = (cont/PPR)*(perimetro); //calcula dist.
436                 lcd.setCursor(0, 1); //move cursor
437                 lcd.print(distancia); //mostra dist.
438
439                 status = analogRead(AnalogIn); //verifica status
440                 if(status > swl-15 && status < swl+15)//status select
441                 {
442                     break; //fim da funcao modo livre
443                 }
444                 else if(distancia > 20000)

```

```
445         {
446             break;
447         }
448     }
449     digitalWrite(DIR, LOW); //desativa saida digital DIR
450     digitalWrite(PUL, LOW); //desativa saida digital PUL
451
452     pos_correta = 2; //posicao final do cabo
453 }
454
455 else if(pos_correta == 2) //se a posicao e a final
456 {
457     lcd.clear(); //limpa display
458     lcd.setCursor(0, 0); //move cursor
459     lcd.print("FIM"); //printa a mensagem FIM
460     delay(1000);
461     fim = true; //fim da funcao = true
462     break; //sai do loop do modo livre
463 }
464 }
465 }
```