



Universidade de Taubaté
Autarquia Municipal de Regime Especial
pelo Dec. Fed. nº 78.924/76
Recredenciada Reconhecida pelo CEE/SP
CNPJ 45.176.153/0001-22

Departamento de Engenharia Elétrica
Rua Daniel Danelli s/nº Jardim Morumbi
Taubaté-Sp 12060-440
Tel.: (12) 3625-4190

LUIZ FERNANDO SIQUEIRA BONIFÁCIO
MURILLO IMAKAWA LISSONE

ELEVADOR INTELIGENTE USANDO ARDUINO

Taubaté
2018

LUIZ FERNANDO SIQUEIRA BONIFÁCIO
MURILLO IMAKAWA LISSONE

ELEVADOR INTELIGENTE USANDO ARDUINO

Trabalho de Graduação apresentado ao Departamento de Engenharia Elétrica da Universidade de Taubaté, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Elétrica.

Orientador(a): Prof. Me. Marcelo Pinheiro Werneck

Taubaté
2018



Universidade de Taubaté
Autarquia Municipal de Regime Especial
pelo Dec. Fed. nº 78.924/76
Recredenciada Reconhecida pelo CEE/SP
CNPJ 45.176.153/0001-22

Departamento de Engenharia Elétrica
Rua Daniel Danelli s/nº Jardim Morumbi
Taubaté-Sp 12060-440
Tel.: (12) 3625-4190

ELEVADOR INTELIGENTE USANDO ARDUINO

LUIZ FERNANDO SIQUEIRA BONIFÁCIO

MURILLO IMAKAWA LISSONE

**ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO PARTE DO
REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE “GRADUADO EM
ENGENHARIA ELÉTRICA”**

BANCA EXAMINADORA:

Prof. Me. Marcelo Pinheiro Werneck
Orientador

Prof. Rubens Castilho Junior
UNITAU

Prof. Me. Seide da Cunha Filho
UNITAU

Dezembro de 2018

Resumo

Elevadores são imprescindíveis para o dia a dia das pessoas e para as indústrias, tão importante quanto o elevador, é o seu controle. Neste trabalho de Graduação aborda-se os principais tipos de comandos utilizados em elevadores e suas características. Observa-se a possibilidade de montagem de uma maquete através da utilização de Arduino. Tem-se como objetivo ressaltar uma programação de comando diferente onde haverá seletividade entre duas torres, para que a mais próxima seja enviada para o andar pedido.

PALAVRAS-CHAVE: Elevadores, Arduino

Lista de figuras

Figura 1 - Componentes do elevador.....	9
Figura 2- Exemplos de Arduino	10
Figura 3 - Reed switch.....	11
Figura 4 - Exemplo de botoeira	12
Figura 5 - Fluxograma	15
Figura 6 - Reed Switch	17
Figura 7 - Arduino Mega 2560.....	17
Figura 8 - Ponte H L293D	18
Figura 9 - Circuito básico Ponte H.....	18
Figura 10 - Funcionamento Ponte H	19
Figura 11 - Programa Arduino	20
Figura 12- Programa Arduino seleção da torre	21
Figura 13 - Programa Arduino identificação de andar	21
Figura 14 - Programa Arduino Controle do Motor	22
Figura 15 - Ligação motores DC	22
Figura 16 - Motor DC.....	23
Figura 17- Projeto.....	24

Sumário

1. INTRODUÇÃO.....	7
2. DESENVOLVIMENTO.....	8
2.1 - REVISÃO BIBLIOGRÁFICA	8
2.1.1 - Elevadores.....	8
2.1.2 - Componentes do elevador.....	8
2.1.3 - Arduino	9
2.1.4 - Hardware.....	10
2.1.5 - Software	11
2.1.6 - Sensor Reed switch.....	11
2.1.7 - Botões.....	12
2.1.8 - Motores	12
2.2 - TIPOS DE COMANDOS	13
2.2.1 - Comando automático coletivo	13
2.2.2 - Comando automático coletivo seletivo na descida	13
2.2.3 - Comando automático coletivo seletivo na subida e na descida	13
2.2.4 - Comando em grupo.....	14
3. MÉTODO	16
3.1 - MONTAGEM.....	16
3.2 - SENSORES	16
3.3 - CONTROLE.....	17
3.3.1- Hardware.....	17
3.3.2- Software	19
3.4 - ACIONAMENTO.....	22
4. RESULTADOS	24
5. CONCLUSÃO.....	25
REFERÊNCIAS	26

1. INTRODUÇÃO

Elevadores são equipamentos de transportes utilizados para mover pessoas ou bens verticalmente ou diagonalmente, ele é imprescindível para a indústria e no dia a dia das pessoas que utilizam como meio de transporte em geral. Tão importante quanto o elevador, é o seu controle. Desde a parte elétrica quanto a parte de monitoração, devem apresentar robustez e transmitir máxima segurança ao seu operador. O primeiro elevador foi construído em Roma no século I a.C., neste modelo de elevador, o mecanismo era um sistema de transporte de carga vertical composto por um conjunto de roldanas movidas por força humana (muitas vezes escrava), animal, ou água.

A partir da revolução industrial as formas de trações dos elevadores foram sendo substituídas pela energia a vapor, mas mesmo assim o uso dos elevadores por pessoas ainda era muito pouco, devido ao medo e pela frequência de acidentes. Em 1853 ocorreu um marco para a evolução dos elevadores, Elis Graves Otis desenvolveu um sistema de trilhos e travas capaz de sustentar a cabine mesmo no caso de rompimento dos cabos. O primeiro elevador elétrico foi construído por Werner von Siemens em 1880.

Atualmente temos vários tipos de elevadores:

- Elevador a vácuo - Equipamento usado em escritórios, clínicas, casas.
- Elevadores elétricos - Equipamentos usados pra qualquer tipo de atividade, residencial, comercial, construção civil.
- Elevadores Hidráulicos - Equipamento utilizado para levantamento de carga.

2. DESENVOLVIMENTO

2.1 - REVISÃO BIBLIOGRÁFICA

2.1.1 - Elevadores

Segundo o site como isso funciona “O princípio básico de funcionamento de um elevador é o mesmo desde que esta máquina foi inventada, há mais de 150 anos atrás. O elevador fica ligado a um contrapeso por meio de cabos e polias (espécies de roldanas presas a um eixo), movidas através de um motor, assim dando o movimento vertical de sobe e desce. Os elevadores modernos, todavia, contam com complementos que visam garantir a segurança a rapidez deste veículo utilizado por milhões de pessoas, dentre eles estão os freios de emergência, os sensores que evitam o fechamento das portas havendo alguém no caminho e computadores que calculam com precisão o percurso que o elevador percorrerá com a finalidade de não esquecer ninguém em algum dos andares. Com a sua invenção se tornou possível a construção dos imensos arranha-céus com rapidez jamais vista, os elevadores facilitam o transporte de materiais extremamente pesados numa construção de um prédio, por exemplo. Os elevadores também viabilizaram a construção de grandes edifícios residenciais, o qual necessitava de um meio de transporte para a locomoção dos moradores para os andares mais elevados.”

2.1.2 - Componentes do elevador

O elevador tradicional é constituído por quatro componentes principais: cabine, cabos de tração, máquina e contrapeso, outros componentes também são fundamentais para garantir a segurança e o bem-estar dos usuários.

Na imagem a seguir estão listados alguns dos componentes mais comuns em elevadores.

- 01- Quadro de Comando
- 02- Máquina de Tração
- 03- Freio Eletromagnético
- 04- Motor de Tração
- 05- Polia de Tração
- 06- Aparelho Seletor *
- 07- Limitador de Velocidade
- 08- Polia Deslocadora
- 09- Fita Seletora *
- 10- Cabos de Tração
- 11- Cabos do Limitador
- 12- Cabina
- 13- Corrediça de Cabina
- 14- Armação de Cabina
- 15- Guia de Cabina
- 16- Freio de Segurança
- 17- Contrapeso
- 18- Corrediça de Contrapeso
- 19- Armação de Contrapeso
- 20- Guia de Contrapeso
- 21- Mola (para choque)

* não aplicados em novos elevadores

- 22- Polia Tensora
- 23- Limite de Curso
- 24- Soleira de Pavimento
- 25- Amortecedor de Porta
- 26- Botoeira de Andar
- 27- Fecho Eletromecânico
- 28- Indicador de Posição
- 29- Tapa Vista
- 30- Soleira de Cabina
- 31- Botoeira de Cabina
- 32- Operador de Porta
- 33- Cabos de Manobra
- 34- Sapatas de Freio
- 35- Impulsores (sensor)
- 36- Acoplamento do motor

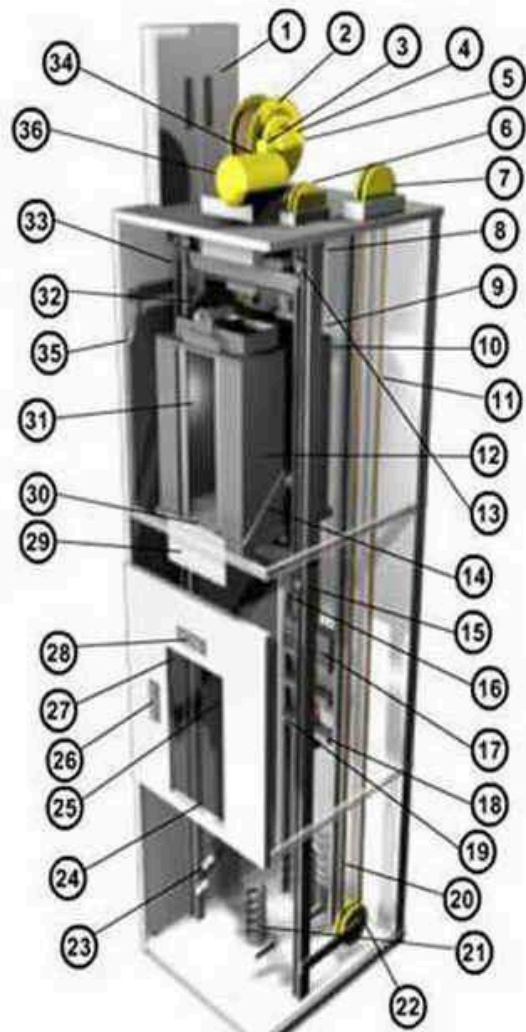


Figura 1 - Componentes do elevador

2.1.3 - Arduino

Arduino começou a ser criado na Itália em 2005, é uma plataforma de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido, uma linguagem de programação padrão, a qual tem origem em Wiring, e é essencialmente C/C++. O objetivo do projeto é criar ferramentas que são acessíveis, com baixo custo, flexíveis e fáceis de se usar por novatos e profissionais. Principalmente para aqueles que não teriam alcance aos controladores mais sofisticados e de ferramentas mais complicadas.

A maior vantagem do Arduino sobre outras plataformas de desenvolvimento de microcontroladores é a facilidade de sua utilização, pessoas que não são da área técnica podem rapidamente aprender o básico e criar seus próprios projetos em um intervalo de tempo

relativamente curto. Artistas, mais especificamente, parecem considerá-lo a forma perfeita de criar obras de arte interativas rapidamente, e sem conhecimento especializado em eletrônica.

2.1.4 - Hardware

O componente principal do Arduino é o seu microcontrolador, de acordo com o seu microcontrolador é definido a quantidade de entradas e saídas que o arduino vai possuir, onde podem ser pinos digitais, analógicos, PWM, comunicação serial, interrupção externa e conectores de alimentação.

Os conectores de alimentação são usados para conexão de placas de hardware com funções específicas (Shields) ou módulos na placa, onde temos os pinos IOREF, Reset, 3,3V, 5 V, GND VIN.

O arduino possui também uma comunicação USB por onde é feita a transferência da sua programação

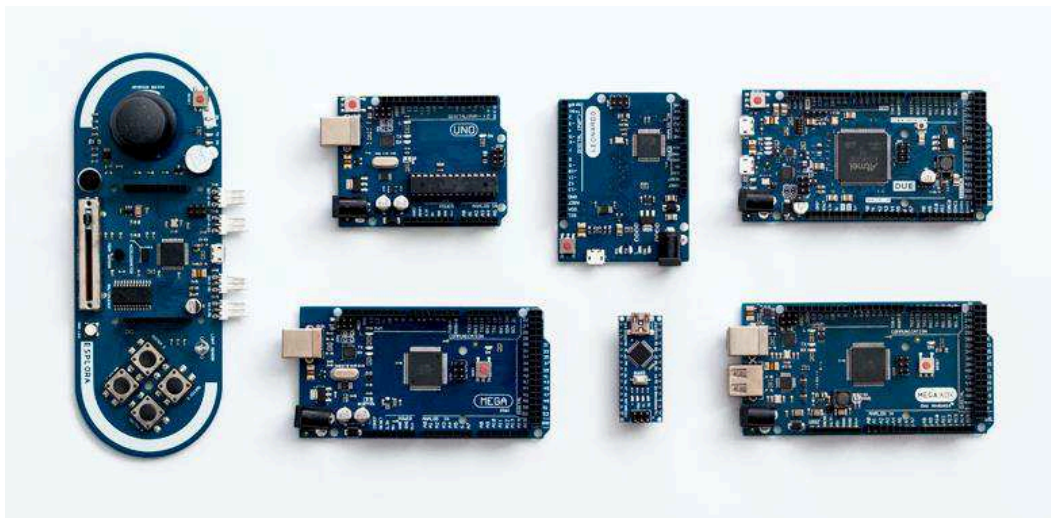


Figura 2- Exemplos de Arduino

Segundo o site ardino.cc “O Arduino Mega 2560 é uma placa microcontroladora baseada no ATmega2560 . Possui 54 pinos de entrada / saída digitais (dos quais 15 podem ser usados como saídas PWM), 16 entradas analógicas, 4 UARTs (portas seriais de hardware), um oscilador de cristal de 16 MHz, uma conexão USB, conector de alimentação, um conector ICSP, e um botão de reset. Ele contém tudo o que é necessário para suportar o microcontrolador. Basta conectá-lo a um computador com um cabo USB ou ligá-lo a um adaptador de CA-CC ou bateria para iniciar. A placa Mega 2560 é compatível com a maioria dos escudos projetados para o Uno e as antigas placas Duemilanove ou Diecimila”

2.1.5 - Software

O ambiente é escrito em Java e baseado em Processing outro software de código aberto. Este software pode ser usado com qualquer placa Arduino. É capaz de compilar e carregar programas para a placa com um único clique.

Tendo uma biblioteca chamada "Wiring", ele possui a capacidade de programar em C/C++. Isto permite criar com facilidade muitas operações de entrada e saída, tendo que definir apenas duas funções no pedido para fazer um programa funcional:

- *setup ()* – Inserida no início, na qual pode ser usada para inicializar configuração.
- *loop ()* – Chamada para repetir um bloco de comandos ou esperar até que seja desligada

2.1.6 - Sensor Reed switch

O reed switch ou interruptor de lâminas, como o nome sugere é um interruptor ou chave que pode ser acionado pelo campo magnético de uma bobina ou de um ímã.

A estrutura básica de um sensor magnético são duas lâminas de material ferromagnético, cada uma ligada a um terminal, envoltas por uma ampola de vidro de alta resistividade. A ampola é vedada e dentro dela existe algum gás inerte para impedir a oxidação das lâminas. Quando um campo magnético se aproxima do sensor, surge uma força magnética de atração entre as lâminas fazendo com que elas se toquem. Dessa forma os terminais do sensor entram em curto, possibilitando a passagem de uma corrente.

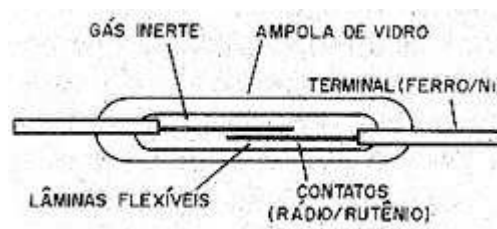


Figura 3 - Reed switch

2.1.7 - Botoeiras

São chaves acionadas manualmente, constituídas por: botão, contato NA (normal aberto) ou NF (normal fechado). Quando seu botão é pressionado, invertem seus contatos, e quando este for solto, devido a ação de uma mola seus contatos voltam à posição inicial.

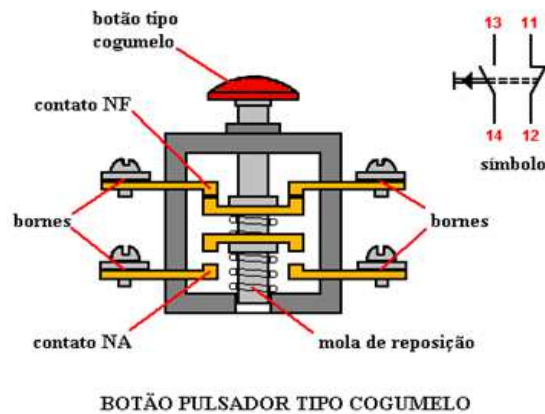


Figura 4 - Exemplo de botoeira

2.1.8 - Motores

Motores elétricos são usados para converter energia elétrica em energia mecânica a fim de produzir trabalho em um sistema. Especificamente, a energia rotacional é produzida a partir da força de campos magnéticos induzidos pela corrente alternada que flui através de bobinas elétricas.

Os motores de corrente contínua devem ser ligados a uma fonte de alimentação para poder funcionar. É devido à polaridade da fonte que o sentido de rotação do eixo do motor é determinado. Quando a polaridade é alterada, o motor girará para o lado contrário do que estava girando anteriormente. Já sua velocidade é determinada pela tensão fornecida pela fonte de alimentação. Essa rotação, por sua vez, tem um movimento suave e contínuo.

Como mencionando anteriormente, a velocidade de um motor é determinada pela tensão fornecida pela fonte de alimentação. O objetivo de um controlador de velocidade do motor é dar um sinal que representa a velocidade exigida, acionando então o motor a velocidade solicitada. Existem vários tipos de motores e a saída do controlador de velocidade dependerá dessas variedades. A grande vantagem de um controlador de velocidade segundo HILLS (2005), é que são facilmente encontrados em brinquedos e tem um custo baixo e podem até

serem adquiridos em qualquer sucata ou lojas de eletrônicos. Estes motores são geralmente enrolados em série, o que significa para revertê-las. (PATRICIA. 2010, Pág. 45/46)

2.2 - TIPOS DE COMANDOS

Todo elevador é controlado por um Controle Lógico Programável (CLP) onde é armazenado a sua programação, o tipo de comando que ele irá executar, cada comando possui um uso específico ou mais apropriado para o local onde foi instalado.

As desvantagens do CLP referente ao Arduino e que eles são mais caros e mais difícil de programar já a vantagem é que o CLP é mais robusto.

A seguir estão descritos os comandos mais usados em elevadores.

2.2.1 - Comando automático coletivo

Comando automático caracterizado por existirem botões de chamada, um para cada pavimento, instalados na cabina, e possuir um único botão de chamada instalado em cada pavimento, todos ligados ao painel central, de tal maneira que todas as chamadas fiquem nele registradas. O carro vai efetuando as paradas em ordem sequencial independentemente da ordem em que as chamadas tenham sido registradas e prossegue no sentido do movimento inicial atendendo a todas as chamadas feitas. Aplica-se a edifícios de poucos andares (de 2 até 3 pavimentos) e pouco movimento, em que o tráfego predominante seja entre andares, como estabelecimentos comerciais e industriais pequenos.

2.2.2 - Comando automático coletivo seletivo na descida

Comando automático coletivo no qual as chamadas de pavimento somente são atendidas quando o elevador se movimenta em sentido descendente, a partir de chamada superior. Aplica-se a edifícios em que o movimento principal é constituído pelo tráfego entre o térreo e os demais pavimentos, sem que haja tráfego apreciável entre os próprios pavimentos. É, portanto, o sistema ideal para edifícios de apartamentos.

2.2.3 - Comando automático coletivo seletivo na subida e na descida

Comando automático coletivo no qual existem nos pavimentos intermediários, dois botões, um de “subida” e um de “descida”, e um botão nos pavimentos extremos. Neste sistema de comando as chamadas de pavimento para subir são selecionadas separadamente

das chamadas de pavimento para descer, sendo atendidas primeiramente todas as chamadas em um dos sentidos para depois serem atendidas as de sentido oposto. Aplica-se a edifícios onde o fluxo predominante seja entre os andares, tais como escritórios em geral ou de uma única entidade, repartições públicas, etc. Em edifícios residenciais se aplica ao pavimento térreo sempre que existirem pavimentos inferiores de garagem.

2.2.4 - Comando em grupo

Comando automático para grupo de dois ou mais elevadores que operam em conjunto e que tenham o mesmo número de paradas, entradas no mesmo hall, somente um pavimento principal de acesso e a mesma destinação de uso (exigências na NBR-5665). Todos os elevadores de passageiros ou todos de serviço, etc., não incluindo elevadores isolados. Nos mais simples, o comando, além de efetuar a seleção de chamadas de descida ou chamadas de subida e descida, seleciona também qual o elevador deverá atender a determinada chamada de pavimento. Estes sistemas são indicados para qualquer tipo de edifício, sempre com melhor rendimento para o fluxo de tráfego. Aplica-se nos casos em que não há divisão no hall de acesso entre os elevadores social e de serviço e sempre que os elevadores estejam próximos, dispostos em grupo (lado a lado ou frente a frente). NOTA: Obedecendo restrições de códigos de obras locais, a distância máxima entre carros de um mesmo grupo tem seu limite fixado em 6m (cidade do Rio de Janeiro). Nos sistemas mais complexos, além das seleções acima descritas, o comando determina, nas horas de pico, quais são as chamadas prioritárias (chamadas de pavimento principal, chamadas de descida chamadas de subida, etc.). Além disso, esses comandos têm extrema flexibilidade, adaptando-se às mais variadas situações de tráfego. São indicados para edifícios com grande fluxo de tráfego.

A seguir um fluxograma que descreve a operação quando um morador do 5º andar sai para o trabalho e considerando a hipótese de novas chamadas quando a cabine chega ao 5º andar, esse elevador usa o comando coletivo seletivo na descida.

Hipóteses:

H1 = a cabine atenderá a chamada no andar superior e após irá ao térreo.

H2 = a cabine irá direto ao térreo (inexistem chamadas de hall ou externas).

H3 = a cabina descera atendendo a chamada do andar inferior e após, irá ao térreo

O objetivo deste TG é fazer uma programação lógica diferente das descritas acima, para controlar duas torres de elevadores aos mesmo tempo e em cada pavimento ter somente um botão de chamada o programa irá verificar qual elevador está mais próximo do andar

solicitado e enviá-lo, com isso podemos ter uma diminuição no consumo de energia pois o elevador andarรก menos e somente um elevador irรก atรก aquele pavimento.

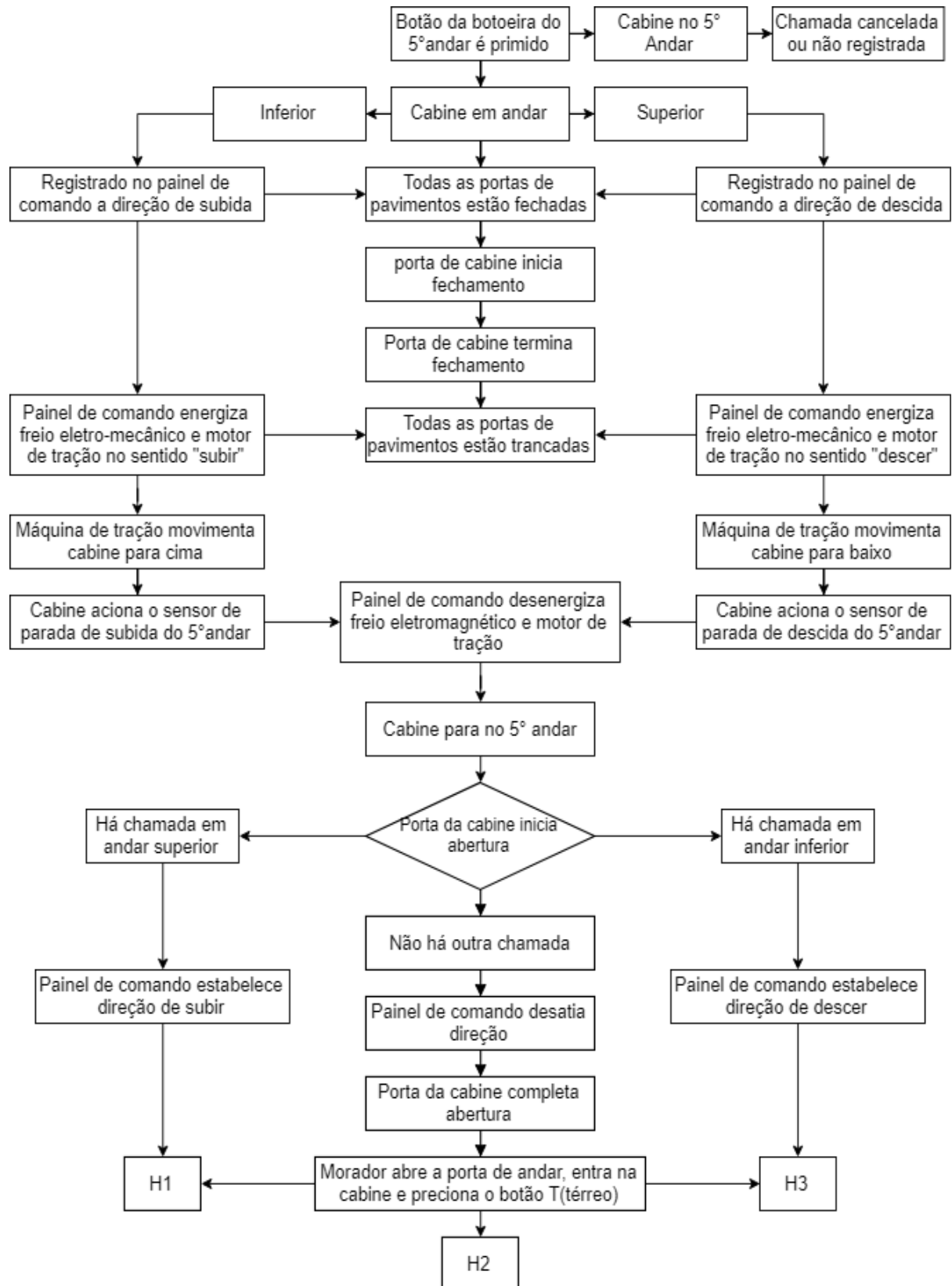


Figura 5 - Fluxograma

3. MÉTODO

O presente trabalho pode ser classificado como conhecimento técnico. O método proposto combina o uso de ferramentas que auxiliam o entendimento da estratégia de negócio e operacional de uma organização com a sequência de ferramentas lógicas para identificação, análise e solução de problemas elaborado por Dettmer (1997).

Utilizando-se como materiais, instrumento e ferramentas para construção do projeto

Material – MDF;

Sensores- reed switch, botão de acionamento;

Programadores – Arduino Mega 2560, Ponte H L293D;

Tração – Motores DC;

3.1 - MONTAGEM

A estrutura para o nosso projeto foi praticamente feita toda em MDF uma madeira de fácil manuseio e leve, para o trilho da cabine do elevador foi usado 4 vergalhões de 4mm, a cabine é uma caixinha de plástico leve ideal para que o motor não faça muita força e possa simular a subida e descida de um elevador.

3.2 - SENSORES

Sensores são dispositivos que responde a estímulos químicos e/ou físicos que pode ser transformada em grandezas físicas para fins de medição e/ou monitoramento.

Atualmente no mercado existem vario tipos de sensores por exemplo: sensores ópticos, de som, de temperatura, de calor, de radiação, de resistência elétrica, de tensão elétrica, magnéticos etc.

Para o nosso projeto utilizamos os sensores magnéticos (reed switch), muito utilizado como chave liga/desliga, por isso tendo muitas aplicações, como detectar abertura de portas, portões e janelas, e até mesmo sensor de presença.

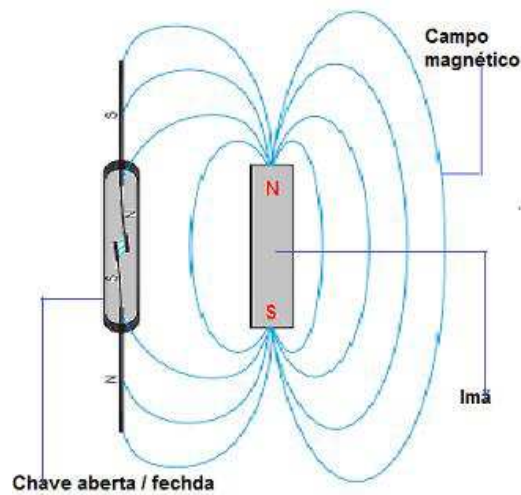


Figura 6 - Reed Switch

3.3 - CONTROLE

3.3.1- Hardware

Para controlar nosso elevador foi escolhido o Arduino Mega 2560 ele é projetado para projeto mais complexos devido a sua quantidade de pinos, possibilitando a ligação de vários componentes a ele, conta com um total de 54 pinos.



Figura 7 - Arduino Mega 2560

Utilizamos um total de 15 pinos digitais, 5 para os botões configurados com INPUT_PULLUP e outros 10 para os reed switch responsável por dizer qual andar está localizado a cabine do elevador são configurados como INPUT.

Para a ligação e controle dos motores DC utilizamos a Motor Shield L293D que possui 2 chips L293D e um 74HC585, e permite controlar 4 motores DC, 2 servos ou 2 motores de passo e suporta motores com tensões de 4,5 à 16 VDC

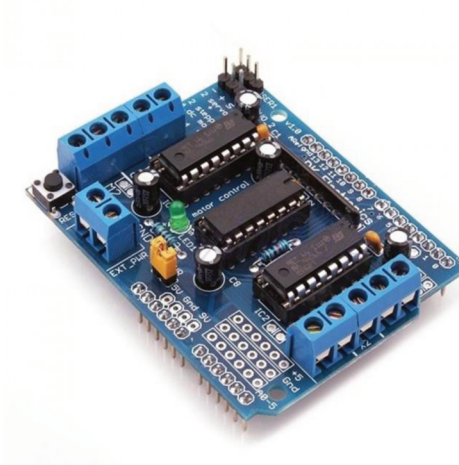


Figura 8 - Ponte H L293D

O chip L293D é também chamado de ponte H, essa ponte possui um circuito simples composto por 4 chaves.

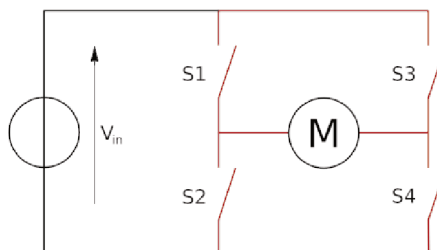


Figura 9 - Circuito básico Ponte H

Quando acionamos a chave S1 e a chave S4, o sentido da corrente será da esquerda para a direita, acionando o motor. Desligando as chaves S1 e S4 e ligando as chaves S2 e S3, o sentido da corrente passa a ser da direita para a esquerda, invertendo o sentido de rotação do motor. Este é o conceito de ponte H.

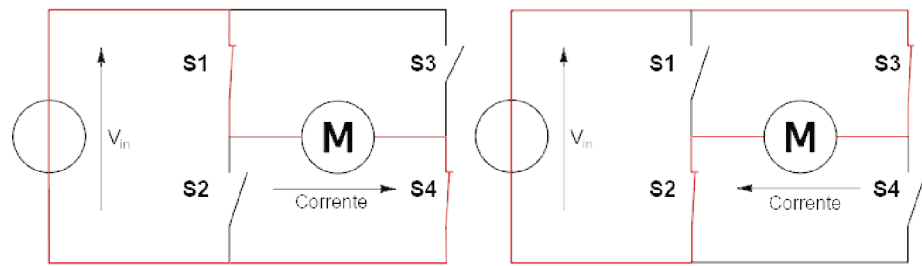


Figura 10 - Funcionamento Ponte H

3.3.2- Software

Para a programação do Arduino utilizamos o software disponível no próprio site da Arduino, de versão 1.8.7, é um software Arduino (IDE) de código aberto facilita a gravação de código e o upload para a placa, compatível com Windows, MAC OSX e Linux, é escrito em Java e baseado em Processing e outro software de código aberto. E ele pode ser usado com qualquer Arduino.

Para iniciar a programação foi necessário a utilização de duas bibliotecas, uma para a ponte H, chamada AFMotor, onde ela configura todos o pinos utilizados pela shield e facilitando a sua programação onde conseguimos controlar a velocidade do motor com o comando `motorT1.setSpeed(velocidade)`, a variação da velocidade pode obter um valor 0 (motor parado e 255 (rotação máxima), e o controle do sentido pode ser controlado pelo comando `motor.run(sentido)` aciona o motor no sentido definido `forward(frente/horário)`, `backward (sentido contrário/anti- horário)`, ou `release` para parar o motor.

Para os botões foi utilizado a biblioteca PushButton, onde ela resolve o efeito bounce e define ele como `input_pullup`, o efeito bounce acontece quando utilizamos chaves mecânicas com circuitos elétricos, ele é um pequeno problema que pode equivocadamente indicar diversos acionamentos em um pequeno intervalo de tempo.

A biblioteca PushButton atua da seguinte maneira, pega-se o estado anterior do botão e confere com o estado atual do botão, se eles forem diferentes efetua-se o que é descrito para executar.

O programa foi basicamente dividido em 2 etapas, a primeira é quando nenhum andar é chamado nesse bloco de comando ele fica examinando se algum botão foi pressionado e faz o calculo da localização de cada torre para posteriormente verificar qual torre ser chamada.

teste_tcc_2_torre	elevador1	elevador_2	motor_1	motor_2
-------------------	-----------	------------	---------	---------

```

if (botao2.pressed()) {
  andarChamado = 2;
  delayTempo = millis();
}

if (botao3.pressed()) {
  andarChamado = 3;
  delayTempo = millis();
}
if (botao4.pressed()) {
  andarChamado = 4;
  delayTempo = millis();
}

if (andarChamado >= andarT1()){
  torre1 = andarChamado - andarT1();
}

if (andarChamado <= andarT1()){
  torre1= andarT1() - andarChamado;
}

if (andarChamado >= andarT2()){
  torre2= andarChamado - andarT2();
}

if (andarChamado <= andarT2()){
  torre2= andarT2() - andarChamado;
}

```

Figura 11 - Programa Arduino

A segunda parte do programa é para quando algum andar foi chamado, ele verifica qual torre está mais perto, assim selecionando qual motor acionar para que a andar chamado seja atendido.

```

else {

    if (torre1 >= torre2){
        (motor2());
    }

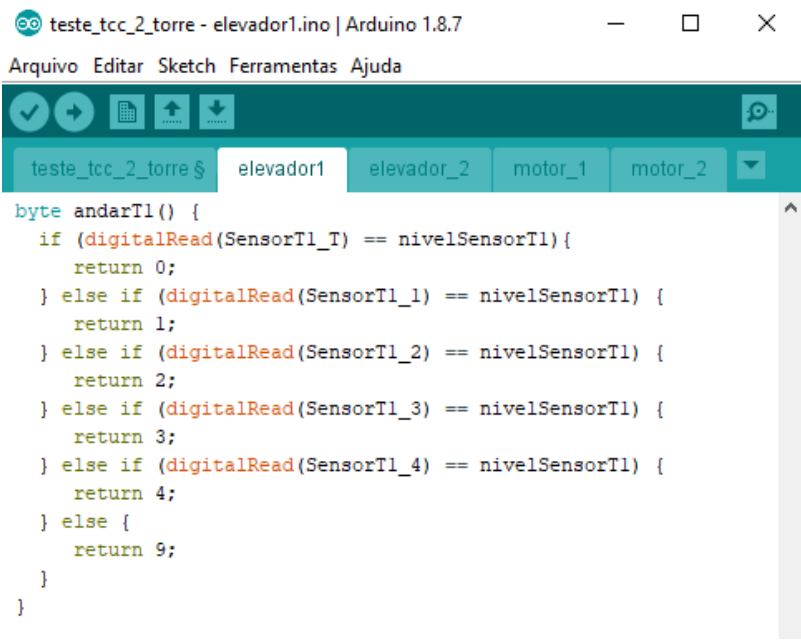
    if (torre1 < torre2){
        (motor1());
    }

}

```

Figura 12- Programa Arduino seleção da torre

Para a leitura das cabines foi feita uma função separa onde cada sensor é lido e retorna o número do andar correspondido.



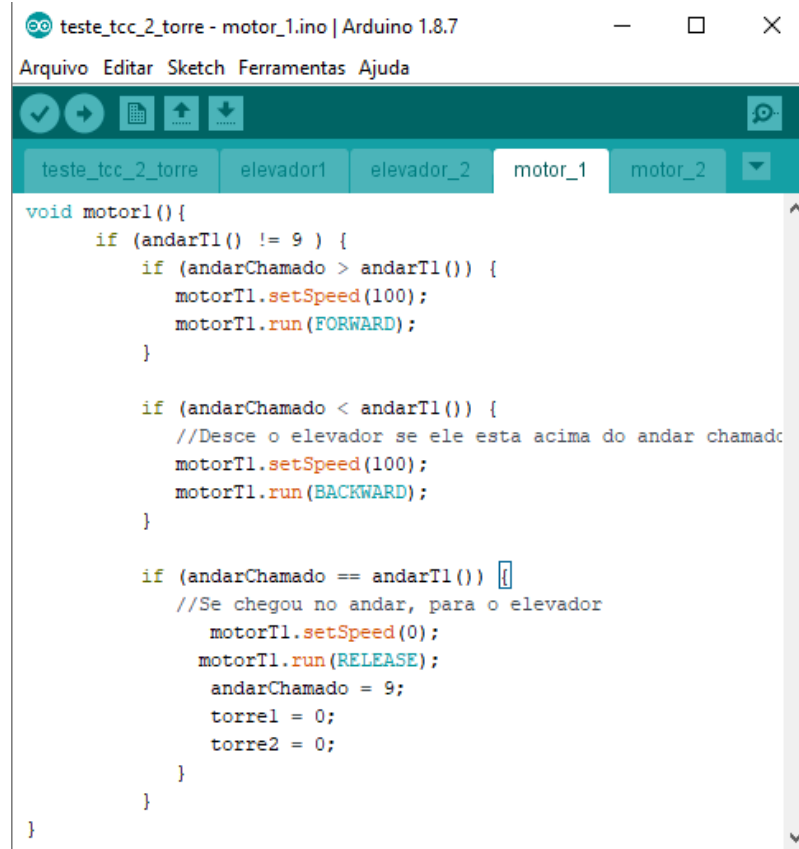
```

teste_tcc_2_torre - elevador1.ino | Arduino 1.8.7
Arquivo Editar Sketch Ferramentas Ajuda
teste_tcc_2_torre $ elevador1 elevador_2 motor_1 motor_2
byte andarT1() {
  if (digitalRead(SensorT1_T) == nivelSensorT1){
    return 0;
  } else if (digitalRead(SensorT1_1) == nivelSensorT1) {
    return 1;
  } else if (digitalRead(SensorT1_2) == nivelSensorT1) {
    return 2;
  } else if (digitalRead(SensorT1_3) == nivelSensorT1) {
    return 3;
  } else if (digitalRead(SensorT1_4) == nivelSensorT1) {
    return 4;
  } else {
    return 9;
  }
}

```

Figura 13 - Programa Arduino identificação de andar

Para cada motor também foi feita uma função utilizando os comandos da biblioteca AFMotor.



```

teste_tcc_2_torre - motor_1.ino | Arduino 1.8.7
Arquivo Editar Sketch Ferramentas Ajuda
teste_tcc_2_torre elevador1 elevador_2 motor_1 motor_2
void motor1() {
  if (andarT1() != 9) {
    if (andarChamado > andarT1()) {
      motorT1.setSpeed(100);
      motorT1.run(FORWARD);
    }

    if (andarChamado < andarT1()) {
      //Desce o elevador se ele esta acima do andar chamado
      motorT1.setSpeed(100);
      motorT1.run(BACKWARD);
    }

    if (andarChamado == andarT1()) {
      //Se chegou no andar, para o elevador
      motorT1.setSpeed(0);
      motorT1.run(RELEASE);
      andarChamado = 9;
      torrel = 0;
      torre2 = 0;
    }
  }
}

```

Figura 14 - Programa Arduino Controle do Motor

3.4 - ACIONAMENTO

O acionamento dos motores foram feitos pela ponte H uma shield descrita na parte de hardware, permitindo que eles girem nos sentido horário e anti-horário, a ligação dos motores é muito simples basta liga-los nos terminais indicados na shield.

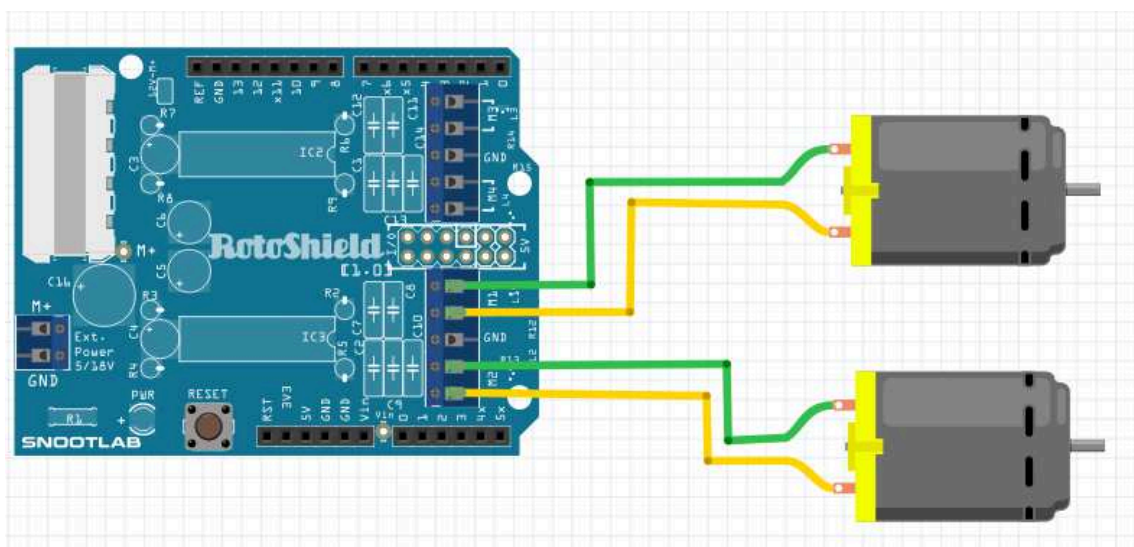


Figura 15 - Ligação motores DC

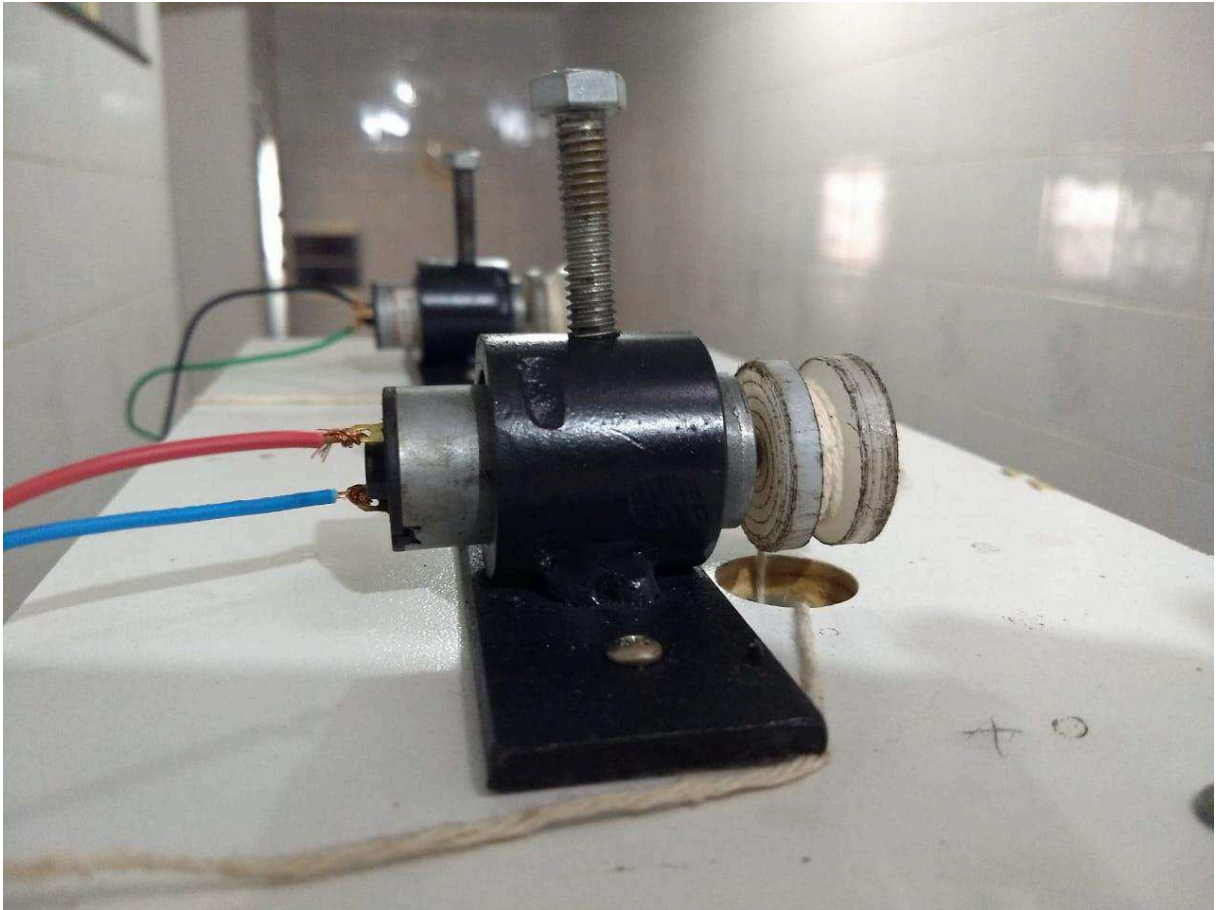


Figura 16 - Motor DC

4. RESULTADOS

A sistemática para a realização do projeto de análise da conformidade no período do 2º semestre de 2018 vem no intuito da realização dessas atividades e foi possível desenvolver boas habilidades com relação ao sistema dos elevadores em si com tudo tendo acompanhando o andamento de todo seu processo, para o intuito de uma melhoria. Além disso, foi possível desenvolver uma boa noção do desenvolvimento da fabricação do equipamento.

O intuito do projeto é formar um Plano de Ação eficaz de conseguir a seleção de chamada dos andares o elevador estiver mais próximo possível da incidência de defeitos e erros. Assim nesta etapa obteve-se o máximo de conhecimento das fases dos processos e a preocupação com seu sistema de controle, sendo elaborada uma pesquisa literária. Foi confeccionada uma maquete que foi utilizada para simulação do projeto na sua execução para todos os envolvidos no desenvolvimento do mesmo, possa analisar e corrigir os erros e fatos para a sua conformidade.



Figura 17- Projeto

5. CONCLUSÃO

A experiência adquirida na realização do TG acrescentou fatos positivos à minha formação pessoal e profissional, uma oportunidade única, e que o mais importante para a realização é saber agregar somente os bons valores e conhecimentos aprendidos, descartando todos os vícios e maus exemplos profissionais, aperfeiçoando cada vez mais tecnicamente e pessoalmente, e amadurecendo com os acontecimentos e com a vivência profissional.

O trabalho é de fundamental importância para a construção do jovem profissional, sendo uma das primeiras experiências de vida, que deve ser plenamente aproveitada, sendo assim, recomendo para as próximas turmas que aproveitem o máximo do período de aprendizado, que sem dúvidas será de grande valia para a construção de um novo profissional.

O projeto conseguiu chegar no resultado esperado, mas ainda pode ser feito algumas melhorias futuras como por exemplo a instalação de led indicador de andar para cada torre, uma interface homem máquina para que possa ser apresentado horas trabalhadas de cada motor, qual andar é mais chamado, alerta de manutenção preventiva de alguns componentes.

REFERÊNCIAS

DETTMER, H.W., Goldratt's Theory of Constraints: A Systems Approach to Continuous Improvement. Milwaukee: ASQ Quality Press. 1997,

MCROBERTS, Michael. Arduino Básico. Novatec Editora Ltda, Set. 2011
https://edisciplinas.usp.br/pluginfile.php/4287597/mod_resource/content/2/Ardu%C3%ADno%20B%C3%A1sico%20-%20Michael%20McRoberts.pdf > Acessado 15 de outubro 2018

PATRICIA, Maria, ACIONAMENTO DE MOTOR ELÉTRICO, Jun.2010
<http://repositorio.uniceub.br/bitstream/123456789/3209/2/20465952.pdf> > Acessado 17 de outubro 2018

<https://super.abril.com.br/mundo-estranho/como-funciona-o-elevador/>
 > acessado 07 setembro 2018

<http://comoissofunciona.blogspot.com/2007/10/como-funciona-o-elevador.html> > acessado 07 setembro 2018

http://www.roboliv.re/conteudo/sensor-magnetico_ > acessado 07 setembro 2018

<http://eletronworld.com.br/eletronica/efeito-bounce/> > acessado 03 novembro 2018

<https://store.arduino.cc/usa/arduino-mega-2560-rev3> > acessado 16 outubro 2018

<https://www.filipeflop.com/blog/controle-motor-dc-arduino-motor-shield/> > acessado 16 outubro 2018

<https://www.portalsaofrancisco.com.br/curiosidades/elevador> > acessado 02 dezembro 2018

ABNT – Associação Brasileira de Normas Técnicas. NBR 16042, ABRI 2012.

ABNT – Associação Brasileira de Normas Técnicas. NR35, MARÇO 2012.

ABNT – Associação Brasileira de Normas Técnicas. NR18, ABRI 2012.

ABNT – Associação Brasileira de Normas Técnicas, NBR 13994, MAIO 2000

**Ficha catalográfica elaborada pelo
SIBi – Sistema Integrado de Bibliotecas / UNITAU**

L772e Lissone, Murillo Imakawa
Elevador inteligente usando arduino / Murillo Imakawa
Lissone, Luiz Fernando Siqueira Bonifácio. – 2018.
27f. : il.

Monografia (Graduação) – Universidade de Taubaté,
Departamento de Engenharia Mecânica e Elétrica, 2018.
Orientador: Prof. Me. Marcelo Pinheiro Werneck,
Departamento de Engenharia Mecânica.

1. Elevador. 2. Arduino. 3. Duas torres. I. Bonifácio, Luiz
Fernando Siqueira. II. Título.

CDD- 670