

# **AUTOMAÇÃO DE UM PROTÓTIPO DE ELEVADOR**

ARTHUR PUK G L DE SOUSA – arthurpukles@gmail.com

BRUNO HENRIQUE DA COSTA E SILVA – bruno.h@uol.com.br

RODRIGO MARIANO GONÇALVES VICENTE – rodrigomgvicente@gmail.com

PROF. DR. JOSÉ IGNACIO HERNÁNDEZ LÓPEZ (Orientador) – jihlpez@gmail.com

## **RESUMO**

Este artigo cujo tema fala sobre uma Automação de um Protótipo de Elevador controlado eletronicamente por um Arduino, apresenta a montagem e a programação desenvolvida em C++ para controlar botões de emergência em cada andar do elevador, para pessoas que precisam se locomover o mais rápido possível para determinado andar, tanto em prédios comerciais como residenciais. Se um botão de emergência for pressionado, todos os chamados anteriores do elevador são ignorados e o elevador mais próximo se direciona diretamente para o andar emergencial, com velocidade levemente elevada.

Palavras-chave: Elevador. Automação. Arduino.

## **AUTOMATION OF AN ELEVATOR PROTOTYPE**

### **ABSTRACT**

This article discusses about an Arduino Electronically Controlled Elevator Prototype, in which presents the assembly and code developed using C++ programming language to control emergency buttons on each elevator floor for people who need to move around as quickly as possible for certain floor, in both commercial and residential buildings. In case an emergency button is pressed, all previous calls from the elevator are ignored and the nearest elevator goes directly to the emergency floor with slightly elevated speed.

Keywords: Elevator. Automation. Arduino.

# 1 INTRODUÇÃO

A área da Engenharia Mecatrônica apresenta pra sociedade uma forma de facilitar a vida no dia a dia devido a utilização de tecnologias modernas que podem ser aplicadas em diversos equipamentos para que eles possam ser automatizados. Apesar do primeiro elevador elétrico ter sido construído em 1880 muitos avanços ainda podem ser aplicados aos elevadores, sejam eles para locomoção de bens ou pessoas.

## 1.1 Breve história

A história do elevador começa como uma peça de teatro. Desde maio até outubro de 1854, o mecânico Elisha Graves Otis fez diversas apresentações na Exposição da Indústria de todos as nações em Nova York, a fim de demonstrar a efetividade e a segurança do dispositivo que ele havia inventado. No ano seguinte em setembro Otis fundou a E. G. Otis Elevator Company, em Yonkers, Nova York. No Crystal Palace localizado na quadragésima segunda rua, Otis instalou uma plataforma que era guiada por trilhos onde foi içado até a altura máxima, para o terror da plateia ali assistindo, ele cortou o cabo de suspensão. Mas ao invés de cair de 15 metros para o chão, o elevador parou somente alguns centímetros abaixo após o corte. “Tudo seguro, cavalheiros, tudo seguro”, Otis tranquilizando a plateia. A E. G. Otis está até hoje no mercado, atuando em diversos segmentos de elevadores e escadas rolante, sendo uma das maiores empresas desse segmento desde sua criação.

Ao longo do desenvolvimento tecnológico diversas mudanças foram feitas nos projetos dos elevadores, como por exemplo, para controlar a localização dos antigos elevadores se usavam sensores eletromagnéticos em cada andar, que mandavam sinais para uma caixa de relés, que era de grandes dimensões e peso. Hoje em dia tudo é controlado por computadores e sensores com dimensões e pesos muito inferiores comparados aos antigos.

Os elevadores da época de Otis demoravam em média 2 minutos para alcançar o oitavo andar, já os atuais podem chegar a 550 m/min o que significa que são mais de 45 vezes mais rápidos que seus antecessores.

## 1.2 Problemas de pesquisa

Os elevadores modernos possuem diversas tecnologias para proporcionar conforto e eficiência para seus passageiros e cargas. Cada grande fabricante possui suas próprias tecnologias e são protegidas e bem restritas para pesquisa.

Será apresentado o código desenvolvido para controlar o elevador e suas funções emergenciais, beneficiando quem necessita se locomover rapidamente de um determinado andar, funções estas que atualmente não são comuns em elevadores prediais, sejam estes residenciais ou comerciais.

### 1.3 Justificativa

Atualmente o mercado de elevadores está centralizado em algumas grandes empresas que o dominam, pela qualidade, tecnologia, segurança e serviço prestado para com seus clientes. Seus segredos de tecnologia são bem guardados garantindo que seu produto seja diferenciado, assim esses diferenciais são bem restritos para pesquisa.

Para que um ambiente possa ser considerado de boa acessibilidade espacial, ele deve proporcionar às pessoas condições de ir e vir com total independência. O indivíduo necessita de segurança e conforto. É preciso que haja uma boa orientação, onde as informações estejam em lugares estratégicos para que o usuário não tenha qualquer dificuldade em relação a utilização dos equipamentos (DISCHINGER, ELY & PIARDI, 2012). Com isso, elevadores necessitam estar adaptados para facilitar a locomoção das pessoas com limitações (ALMEIDA et al., 2008).

Cogita-se neste artigo a viabilização de uma opção otimizada de elevadores para os portadores de necessidades físicas ou deficiências locomotoras e casos emergenciais. Nela, deseja-se priorizar a utilização do mesmo, por pessoas com necessidades especiais e em casos de emergência o elevador irá entrar em um modo prioritário para atender ao chamado.

Uma maior adequação de ambientes, construções e espaços públicos faz-se necessária para a promoção da qualidade de vida da população (MACHADO JUNIOR, OLIVEIRA & SILVA, 2005).

A solução deste problema está baseada no uso de microcontroladores, circuitos elétricos e no resultado da tecnologia disponível para a época.

O resultado disto é desenvolver uma programação juntamente com um circuito para realizar a automação do protótipo que abranja todas as características anteriormente citadas.

### 1.4 Objetivo geral

Desenvolver e automatizar o funcionamento de um elevador através de um microcontrolador Arduino, utilizando-se de linguagem de programação C++ para controlar as funções básicas do elevador, bem como seus botões de emergência.

### 1.5 Objetivos específicos

O protótipo foi desenvolvido com o propósito de utilizar o microcontrolador Arduino para controlar um motor de corrente contínua, controlando sua velocidade e também gerir o sistema de inteligência do elevador, que é focado em sua codificação para automatizar as funcionalidades dos botões de emergência.

## 2 METODOLOGIA

### 2.1 Arduino MEGA 2560 R3

Microcontroladores são uma mistura de hardware e software. Através de uma programação pode-se controlar um hardware para que ele faça funções específicas. Trata-se de um sistema integrado, assim como um microprocessador. Eles incorporam vários recursos em um único lugar, tal como: memória, processador, portas de entrada e saída, etc.

Foi utilizado o Arduino Mega 2560, o qual possui um microcontrolador Atmega2560, 54 pinos de entradas e saídas digitais onde 15 dessas são saídas PWM além disso possui 16 entradas analógicas e 4 portas de comunicação serial.

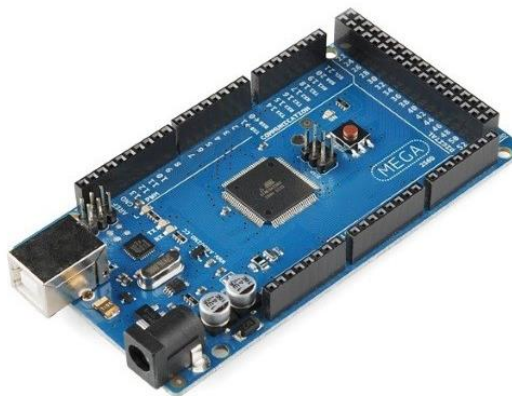


Figura 3 - Arduino MEGA 2560 R3

Fonte: <https://www.filipeflop.com/produto/placa-mega-2560-r3-cabo-usb-para-arduino/>

Foi utilizado a plataforma Arduino Software (IDE), que é um conjunto de ferramentas open source, na qual possui um ambiente de desenvolvimento. A placa Arduino possui entradas e saídas que permitem a integração com outros dispositivos ou aparelhos.

### 2.2 – Micromotor DC Akiyama com Caixa de Redução 12V 44RPM

O micromotor DC é uma forma simples para que se possa obter movimentos mecânicos em dispositivos eletromecânicos. São potentes e compactos, possuem caixa de redução acoplada, limitando a rotação até 44 RPM e torque máximo de 2.2 Kgf.cm.

Especificações técnicas do produto:

- Corrente: 410mA;
- Potência: 2W;
- Tensão (nominal): 12V;
- Tensão (faixa de operação): 12V ~ 24V\*;
- Torque: 2.2 kgf.cm;

- Velocidade (sem carga): 44RPM;
- Velocidade (carga máxima): 33RPM;
- Peso unitário: 125g;



Figura 7 - Micromotor DC Akiyama com Caixa de Redução 12V 44RPM  
 Fonte: <https://www.techmakers.com.br/micro-motor-dc-com-caixa-reducao-12v-44rpm-2kgfcm>

### 2.3 – Driver Motor ponte-H – L298N

Este driver é baseado no circuito tipo Ponte H e é dedicado para controle de motor DC. O mesmo possui dois canais e permite controlar velocidade e sentido de rotação de até dois motores ao mesmo tempo.

### 2.4 – Montagem da estrutura do protótipo do elevador

Através de uma estrutura pré-montada de um protótipo de elevador fabricado pela Arducore, aproveitamos somente de sua estrutura em MDF. Essa estrutura é composta de diversas peças de encaixe. Segue abaixo as medidas:

MDF 3 mm.

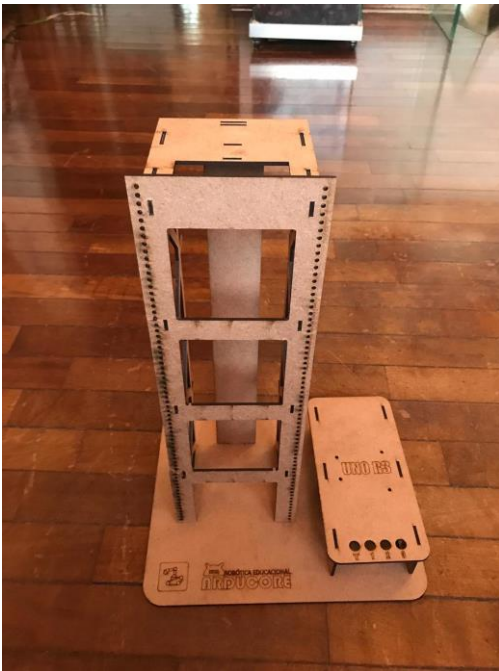
Dimensões da torre:

- 420 mm Comprimento (altura)
- 137 mm diâmetro (largura)
- 122 mm profundidade.

Dimensões da base:

- 300 mm diâmetro (largura)
- 250 mm profundidade

Peso da estrutura: 450g



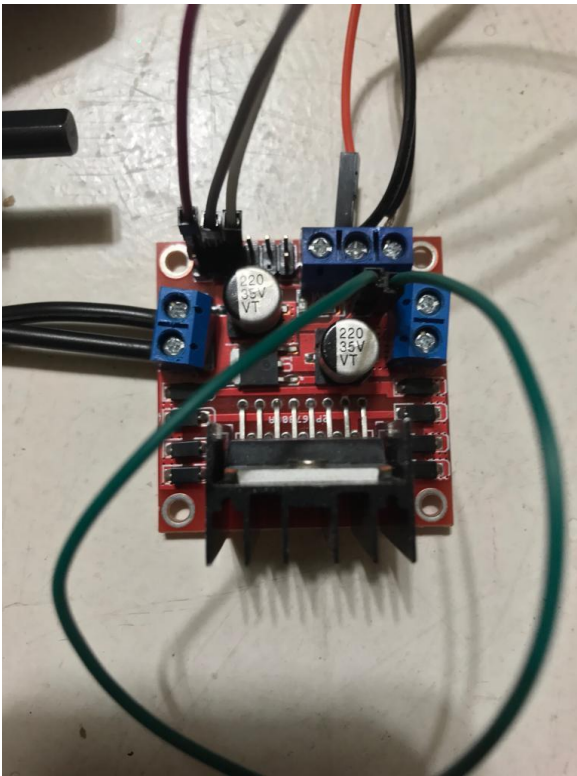
Figuras 1 e 2 – Estrutura do elevador

## 2.5 Ligação do motor no Arduino

Foi utilizado o Micromotor DC Akiyama com caixa de redução 12V 44 RPM. Este motor foi conectado a uma das saídas da Ponte H (OUT3 e OUT4), através de um fio de 2,5 mm.

Após a ligação do motor na ponte H, foi conectado a ponte H ao Arduino, através das entradas IN3, IN4 e a Ativa ENB. A entrada IN3 foi conectada na porta 11 do Arduino, a entrada IN4 foi conectada 12 e a ENB conectada a porta 10 (PWM). A Ativa ENB precisa ser conectada a uma porta PWM pois esse pino controla a velocidade do motor, regulando a tensão que será transferida para o motor. As portas IN3 e IN4 são responsáveis pelo sentido de rotação do motor, alterando um valor HIGH para uma e LOW para outra, para alterar a rotação basta inverter o valor de cada pino.

A alimentação dos motores necessita de uma fonte externa, pois o Arduino somente oferece 3.3V, 5V, e o Vin (que oferece a voltagem da alimentação do Arduino). Foi utilizada uma fonte de 12V DC 2.5V conectada a entrada de 12V da ponte H. É necessário também conectar um fio do GND de alimentação da ponte H ao GND do Arduino.

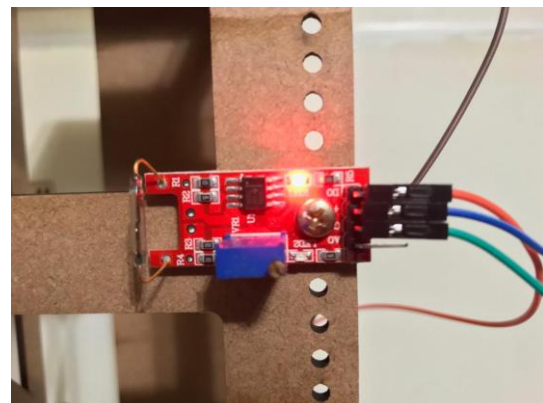
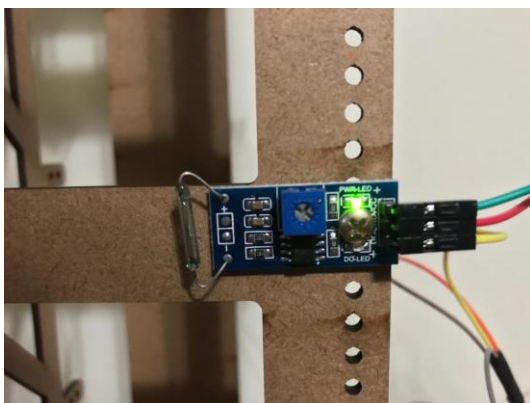


Figuras 3 e 4 – Ligação motor Ponte H

## 2.6 Sensores reed-switch

Foram utilizados dois módulos de sensores, um que trabalha em LOW e outro que trabalha em HIGH. Para cada elevador foram utilizados quatro sensores, para o andar térreo, primeiro andar, segundo andar e terceiro andar.

Para conectar os sensores ao Arduino foram utilizados três pinos dos sensores, são eles: Pino digital, VCC e o GND. Os pinos digitais foram conectados as portas 2, 3, 4 e 5 do Arduino. A porta 2 para o térreo, a porta 3 para o primeiro andar, e assim sucessivamente. Os pinos GND foram conectados a protoboard onde um jumper conecta ao GND do Arduino. Os pinos VCC foram também conectados a protoboard onde um jumper conecta ao 5V do Arduino.



Figuras 5 e 6 – Sensores reed switch

## 2.7 Botões de chamada, botões emergenciais e botões preferenciais

Foram utilizados quatro botões (Pushbutton N.A.) para cada elevador. Cada botão foi conectado a uma porta digital do Arduino, BotãoT na porta 24, Botão1 na 26, Botão2 na 28 e Botão3 na 30. O outro pino do botão foi conectado a protoboard e um jumper conecta ao GND do Arduino.

Tanto os botões emergenciais quanto os preferenciais possuem o mesmo procedimento de conexão com o Arduino, a única diferença é que foram conectados em portas digitais diferentes, expressas na programação.



Figura 7 - Botões

## 2.8 Programação

A programação do elevador foi feita na linguagem C++ pelo software Arduino IDE, utilizando-se apenas das bibliotecas padrões do software, não foi incluída nenhuma biblioteca externa.

A seguir será mostrado alguns trechos mostrando as principais funcionalidades do código desenvolvido.



## 2.8.1 - Sensores

Primeiro foram definidos os pinos para cada sensor e fixamos LOW e HIGH para cada modelo de módulo reed switch. No programa essas pinagens foram declaradas como INPUT.

```
#define pinSensorT 2
#define pinSensor1 3
#define pinSensor2 4
#define pinSensor3 5
#define nivelSensor LOW
#define nivelSensor2 HIGH
```

A seqüência a seguir é uma regra criada para que em diferentes trechos do programa a variável “andar” receba o valor dos sensores que representam o andar em que o elevador se encontra.

```
if (digitalRead(pinSensorT) == nivelSensor) {
  andar = 1;
} else if (digitalRead(pinSensor1) == nivelSensor) {
  andar = 2;
} else if (digitalRead(pinSensor2) == nivelSensor2) {
  andar = 3;
} else if (digitalRead(pinSensor3) == nivelSensor2) {
  andar = 4;
}
```

## 2.8.2 – Botões

### 2.8.2.1 – Botões comuns

Regra aplicada para armazenar na variável “andarChamado” o valor do botão que foi pressionado.

```
if (andarChamado == 9) {
  //Se nenhum andar foi chamado
  Serial.println("Nenhum andar chamado \n");
  Serial.println("Andar:");
  Serial.println(andar);
  analogWrite(velocidadeB, velocidade);
  velocidade = 0;

  if (botaoT.pressed()) {
    andarChamado = 1;
    Serial.print ("Botão T apertado \n");
    delayTempo = millis();
  }

  if (botao1.pressed()) {
    andarChamado = 2;
    Serial.print ("Botão 1 apertado \n");
    delayTempo = millis();
  }

  if (botao2.pressed()) {
    andarChamado = 3;
    Serial.print ("Botão 2 apertado \n");
    delayTempo = millis();
  }

  if (botao3.pressed()) {
    andarChamado = 4;
    Serial.print ("Botão 3 apertado \n");
    delayTempo = millis();
  }
}
```

Regra utilizada para quando a variável obtida pelo sensor (“andar”) tiver o mesmo valor da variável do botão (“andarChamado”), o elevador irá parar e ter um delay de 6 segundos para reiniciar as tarefas de rotina.

Atribuindo o valor 9 para a variável “andarChamado”, a rotina irá cair na condição apresentada anteriormente, onde, se a variável “andarChamado” for igual a 9 o elevador irá permanecer parado pois nenhum outro botão foi pressionado.

```
if (andarChamado == andar) { //Se chegou no andar, para o elevador
    Serial.print ("Elevador no andar! \n");
    Serial.println(andar);
    velocidade = 0;
    analogWrite(velocidadeB, velocidade);
    andarChamado = 9;
    delay (6000);
}
```

#### 2.8.2.2 – Botões emergenciais e preferenciais

Foi criada uma variável “andarEmergencia” para armazenar o andar onde está ocorrendo uma emergência, da mesma forma foi criada outra regra igual para os botões preferenciais. No caso do botão emergencial ele também acendera uma luz de emergência na parte da cabina do elevador para alertar as pessoas que estão dentro do elevador, para que evacuem a cabina assim que o elevador parar em um andar. No interior da cabina terá uma explicação do procedimento padrão de evacuação.

```
if (botaoE1.pressed()) {
    andarEmergencia = 2;
    digitalWrite(38, HIGH);
    Serial.print ("Botão de Emergencia 1 andar apertado \n");
    delayTempo = millis();
}
.....
if (botaoE2.pressed()) {
    andarEmergencia = 3;
    digitalWrite(38, HIGH);
    Serial.print ("Botão de Emergencia 2 andar apertado \n");
    delayTempo = millis();
}
.....
if (botaoE3.pressed()) {
    andarEmergencia = 4;
    digitalWrite(38, HIGH);
    Serial.print ("Botão de Emergencia 3 andar apertado \n");
    delayTempo = millis();
}
```

Se algum dos botões emergenciais forem pressionados o programa irá priorizar essa regra a seguir:

```

if (botaoE1.pressed() || botaoE2.pressed() || botaoE3.pressed()){
Serial.println("Botao de emergencia pressionado:");
Serial.println(andarEmergencia);
Serial.println(andar);
}

```

Com duas regras similares (demonstradas abaixo) foi criada uma rotina para o programa detectar se o elevador se encontra acima ou abaixo do andar emergencial chamado, e inicia a rotação do motor para a sentido correto com velocidade elevada e para o elevador no andar emergencial ignorando todos os chamados anteriores.

```

while (andarEmergencia > andar) { //Sobe o elevador para o andar de emergencia
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
velocidade = 150;
analogWrite(velocidadeB, velocidade);
Serial.print ("Elevador Subindo para andar EMERGENCIAL \n");
Serial.println(andarEmergencia);
Serial.println(andar);
    if (andarEmergencia == andar){
        velocidade = 0;
        analogWrite(velocidadeB, velocidade);
        digitalWrite(38, LOW);
        delay (3000);
    }
}

```

## 2.9 – Cálculos e Normas

A escolha de um elevador depende do tipo de edifício em que ele será instalado – residencial ou comercial. É necessário ter como base a Norma Brasileira NBR 5665.

O número de paradas prováveis que um elevador pode realizar em uma viagem é feito em função da capacidade da cabina (lotação máxima da cabina) e da quantidade de pavimentos a serem atendidos. Esse número é obtido com base no Cálculo de Probabilidades, através da seguinte fórmula da NBR-5665:

$$N = P - (P - 1) \cdot \frac{(P - 2)c}{P - 1} \quad (1)$$

Onde:

N = número de paradas prováveis

P = número de paradas do elevador

c = lotação da cabina

A velocidade do elevador é estabelecida de acordo com a altura do prédio, e as velocidades recomendadas são:

Para edifícios residenciais:

Quadro 1 – Relação altura x velocidade edifício residencial

Percurso (m)	Velocidade (m/s)
até 29	de 0,75 a 1,00
de 30 a 44	de 1,00 a 1,50
de 45 a 59	de 1,25 a 2,00
de 60 a 74	de 1,75 a 2,50
de 75 a 90	de 2,50 a 3,50

Para edifícios não residenciais:

Quadro 2 – Relação altura x velocidade edifício não residencial

Percurso (m)	Velocidade (m/s)
até 17	de 0,50 a 1,00
de 18 a 29	de 1,00 a 1,75
de 30 a 44	de 1,75 a 2,50
de 45 a 59	de 2,50 a 3,50
de 60 a 74	de 3,50 a 4,00
de 75 a 89	de 4,00 a 5,00
de 90 a 150	de 5,00 a 6,00
acima de 150	de 6,00 a 8,00

Estas velocidades são recomendadas visando um melhor atendimento aos usuários do elevador, e, portanto, não são obrigatórias.

Pode-se obter também através da NBR - 5665 o tempo gasto para acelerar e desacelerar o elevador em função de sua velocidade.

Quadro 3 – Tempo de desaceleração em função da velocidade

Velocidade (m/s)	Tempo por parada
0,75	2,5
1,00	3,0
1,25	3,0
1,50	3,5
1,75	4,0
2,00	4,5
2,50	5,5
acima de 2,50	6,0

O tempo gasto para abertura e fechamento das portas do elevador em cada parada é relacionado em função do tipo de porta e pode ser obtido através da seguinte tabela da NBR - 5665:

Quadro 4 – Relação do tipo de porta x tempo de parada

Tipo de Porta	Tempo por parada
Abertura Central (AC)	3,9
Abertura Lateral (AL)	5,5
Eixo Vertical (EV)	6,0

O tempo para entrada e saída de passageiros da cabina por parada é calculado em função da abertura livre da porta, através da seguinte tabela da NBR - 5665:

Quadro 5 – Relação abertura da porta x tempo por passageiro

Abertura da Porta	Tempo por passageiro
menor que 1,10m	2,4
maior ou igual a 1,10m	2,0

O tempo total gasto pela cabina para percorrer determinado percurso, de ida e volta, sem parar em nenhum pavimento é calculado através da seguinte fórmula:

$$T = \frac{2 \cdot S}{V} \quad (2)$$

Onde:

S = percurso em metros

V = velocidade do elevador

A capacidade de transporte é calculada para determinar a quantidade de pessoas que serão transportadas em 5 minutos (300 segundos) por um elevador.

$$X = \frac{300 \cdot C}{T} \quad (3)$$

Onde:

X = capacidade de transporte

C = capacidade da cabina

T = tempo total da viagem (em segundos)

O intervalo de tráfego é o tempo total de viagem dividido pelo número de elevadores:

$$I = \frac{T}{N} \quad (4)$$

Onde:

I = intervalo de tráfego

T = tempo total de viagem (em segundos)

n = número de elevadores do grupo

Conceitualmente, I é o tempo máximo que um usuário pode esperar pelo elevador, ou seja, o máximo tempo de espera entre a partida de um elevador e a chegada de um próximo.

O grau de serviço determina o grau de satisfação dos usuários, calculado de acordo com a seguinte fórmula:

$$K = \frac{T}{4} + \frac{I}{2} \quad (5)$$

Onde:

K = grau de serviço

T = tempo total de viagem (em segundos)

I = intervalo de tráfego

Quadro 6 – Relação do grau de serviço x satisfação do usuário

Grau de Serviço	K
Excelente	até 45
Bom	até 55
Regular	até 65
Inexistente	~

Com as exigências de transporte em 5 minutos e de intervalos máximos, a instalação dos elevadores, além de atender a uma determinada necessidade de transporte, evita que os passageiros tenham esperas longas ou viagens exageradamente demoradas, resultando em satisfação dos usuários pelo melhor serviço prestado.

### 3 RESULTADOS E DISCUSSÃO

Foram visitados 8 prédios comerciais com grande circulação de pessoas e grande número de elevadores, nos quais foi constatado que havia uma necessidade de priorizar certos casos, como por exemplo a dificuldade de pessoas com necessidades especiais, deficiência física e casos emergenciais.

Após se ter notado essa deficiência nos elevadores foi decidido conversar com pessoas que sofrem de alguma deficiência ou já tiveram alguma experiência relacionado a essas lacunas nos elevadores, concluindo que é necessário alterar algumas das funcionalidades dos elevadores padrões, como um botão emergencial e um botão preferencial.

Através de um protótipo e o desenvolvimento de um programa foi criada uma solução que atenda essas lacunas existentes na maioria dos elevadores, tanto comerciais como residenciais.

O desenvolvimento do projeto passou por diversas fases, onde fomos implementando melhorias e adaptando soluções para os problemas que impediam a conclusão do projeto, trocando equipamentos e programação.

Através da programação se obteve os seguintes resultados:

- Botões emergenciais: se um botão de emergência for pressionado, todos os chamados anteriores do elevador são ignorados e o elevador mais próximo se direciona diretamente para o andar emergencial, com velocidade levemente elevada. Uma luz no interior da cabina indica o estado emergencial do elevador e instruções no interior irão auxiliar os passageiros que estão em locomoção para efetuar uma evacuação no andar de parada. Após parada no andar emergencial, o elevador automaticamente desce para o andar térreo ainda com velocidade levemente acelerada, ignorando todos os chamados. Assim que o elevador terminou a rotina de emergência o mesmo volta automaticamente para o andar emergencial para buscar as pessoas que antes estavam dentro do elevador e efetuaram a evacuação de emergência e o elevador retorna a sua rotina comum.
- Botões preferenciais: se um botão preferencial for pressionado, o elevador mais próximo irá ignorar as chamadas comuns do elevador, não passando na frente somente das emergências, e irá se direcionar diretamente para o andar solicitado. Após chegada no andar solicitado o elevador retorna a sua rotina comum.

Baseado nas Normas NBR – 5665, foram realizados os seguintes cálculos utilizando as fórmulas apresentadas na seção 2.9 – Cálculos e Normas.

- Velocidade padrão do elevador:

$$\omega = \frac{2.\pi.rpm}{60} = \frac{2.\pi.24}{60} = 2,51 \text{ rad/s}$$



Velocidade =  $\omega \cdot r$

Onde  $r = 10 \text{ mm}$

$$\text{Velocidade} = 2,51 \cdot 0,01 = 0,0251 \text{ m/s}$$

De acordo com o Quadro 2 – Relação altura x velocidade edifício não residencial, a velocidade normal utilizada representa um prédio de 45 a 59 metros.

- Velocidade de emergência:

$$\omega = \frac{2 \cdot \pi \cdot rpm}{60} = \frac{2 \cdot \pi \cdot 46}{60} = 4,82 \text{ rad/s}$$

Velocidade =  $\omega \cdot r$

Onde  $r = 10 \text{ mm}$

$$\text{Velocidade} = 4,82 \cdot 0,01 = 0,0482 \text{ m/s}$$

De acordo com o Quadro 2 – Relação altura x velocidade edifício não residencial, a velocidade de emergência utilizada representa um prédio de 90 a 150 metros.

De acordo com o Quadro 4 – Relação do tipo de porta x tempo de parada, podemos ter um tempo de parada de 6,0 segundos.

Através de medição obtivemos o percurso total do elevador e este possui 380 mm.

Tempo total gasto pela cabina para percorrer determinado percurso, de ida e volta, sem parar em nenhum pavimento, de acordo com a fórmula número 2:

$$T = \frac{2 \cdot S}{v} = \frac{2 \cdot 0,38}{0,0251} = 30,3 \text{ segundos}$$

Subida medida: 15 segundos.

Cálculo do intervalo de tráfego de acordo com a fórmula número 4:

$$I = \frac{T}{N} = \frac{30,3}{2} = 15,1 \text{ segundos (tempo máximo de espera)}$$

Cálculo do grau de serviço de acordo com o Quadro 6 – Relação do grau de serviço x satisfação do usuário, de acordo com a fórmula número 5:

$$K = \frac{T}{N} + \frac{I}{N} = \frac{30,3}{4} + \frac{15}{2} = 15 \text{ (grau de serviço excelente)}$$

Através da IDE do Arduino foi calculado a aceleração e desaceleração referentes a velocidade normal e velocidade de emergência:

- Velocidade normal:

Aceleração:  $0,273 \text{ m/s}^2$

Desaceleração:  $0,409 \text{ m/s}^2$

- Velocidade de emergência:  
Aceleração:  $1,927 \text{ m/s}^2$   
Desaceleração:  $0,306 \text{ m/s}^2$

Foi obtida esta diferença de valores devido a voltagem de arranque do motor.

## **6 CONSIDERAÇÕES FINAIS**

Com os resultados obtidos foi apresentado o funcionamento do protótipo de elevador de acordo com a programação desenvolvida, a qual prioriza a movimentação do elevador de acordo com o botão que foi pressionado, seja ele para locomover para um andar em sua rotina normal ou para ativar o modo emergencial, apresentando também os cálculos de velocidade e aceleração.

Ao longo do processo de desenvolvimento foi percebida uma falta de materiais de estudos disponíveis para pesquisa, como normas, ensaios, teses, etc. Assim a programação foi desenvolvida sem nenhum tipo de referência anterior, para casos reais de um elevador. Foram obtidas através de pesquisas apenas informações e códigos de casos extremamente simples, que não se aplicam em uma situação real.

O controle de velocidade e aceleração de micromotores é muito limitado por sua potência de arranque elevado, dificultando uma suave aceleração do elevador. Por isso o foco principal foi na parte lógica da programação para suprir a lacuna principal do trabalho e não focar em relação a construção de um protótipo que segue normas efetivas de um elevador real.

Para pesquisas futuras pode-se aperfeiçoar a parte mecânica para que o protótipo se aproxime mais de um elevador real, sendo ele tanto para carga como para transporte de pessoas, utilizando motores melhores e implementando novos sensores ao projeto.

## REFERÊNCIAS

ALMEIDA, P. C.; CARVALHO, Z. M. F.; MORAES, P. O. F.; ROLIM, A. A Acessibilidade em Cadeira de Rodas nas Clínicas e Consultórios de Neurologia e Neurocirurgia de Fortaleza – Brasil. Revista Enferm. Global, num. 14, Ceará, out. 2008.

ARDUINO Uno. Arduino. [S.l.]. Disponível em: <<http://arduino.cc/en/Main/ArduinoBoardUno>>. Acesso em: 28 abr. 2019.

BERNARD, Andreas. Lifted: A Cultural History of the Elevator. 2. ed. Nova York: Nyu Press, 2014. 309 p.

CARVALHO, Maurício Feo Pereira Rivello de. Automação e controle residencial via internet utilizando arduino. In: SEMANA DE EXTENSÃO, 1., Rio de Janeiro. Anais... Rio de Janeiro: [online], 2011. Disponível em: <[http://portal.cefetrj.br/files/extensao/outros/livro\\_sem\\_ext\\_2011.pdf#page=34](http://portal.cefetrj.br/files/extensao/outros/livro_sem_ext_2011.pdf#page=34)> Acesso em: 20 jun. 2019.

DISCHINGER, M.; ELY, V. H. M. B.; PIARDI, S. M. D. G. Promovendo acessibilidade espacial nos edifícios públicos: Programa de acessibilidade às pessoas com deficiência ou mobilidade reduzida nas edificações de uso público. Ministério Público do Estado de Santa Catarina, 2012.

ELEVADOR ROBÓTICO. Disponível em: <<https://www.arduocore.com.br/elevador-robotico-eletronica>>. Acesso em: 15 fev. 2019.

MACHADO JUNIOR, N.; OLIVEIRA, R. F. U.; SILVA, R. N. As condições de acessibilidade e adequação, para pacientes em cadeiras de rodas, em clínicas de fisioterapia na cidade de Goiânia. Universidade Católica de Goiás, Goiânia, 2005. Disponível em <[http://www.wgate.com.br/conteudo/medicinaesaude/fisioterapia/variedades/acessibilidade\\_rafael.htm](http://www.wgate.com.br/conteudo/medicinaesaude/fisioterapia/variedades/acessibilidade_rafael.htm)> Acesso em 13 mar. 2019.

NORMAS BRASILEIRAS E CÁLCULO DE TRÁFEGO. Disponível em: <<https://www.elevadoresmais.com.br/2016-04-11-19-14-56/downloads/9-normas-brasileiras-e-calculo-de-trafego/file>>. Acesso em 10 fev. 2019.

SILVA, Davidson Felipe da. Sistema de comunicação Bluetooth utilizando microcontrolador. 2009. 17f. Trabalho de Conclusão de Curso (Bacharel em Engenharia da Computação) – Curso de Engenharia da Computação, Escola Politécnica de Pernambuco – Universidade de Pernambuco.

TORRES, André Euler. Introdução ao funcionamento e ao acionamento de motores DC. 1995. Disponível em: <<http://www.coep.ufrj.br/~jpaulo/MOTOR-DC-Euler.pdf>>. Acesso em: 20 maio 2018.