

CENTRO DE ENSINO SUPERIOR DE CONSELHEIRO  
LAFAIETE – CES-CL  
ENGENHARIA ELÉTRICA

## **RELÉ DIGITAL DE PROTEÇÃO UTILIZANDO ARDUINO**

MARCELO AUGUSTO VIEIRA

CONSELHEIRO LAFAIETE – MG

2017

MARCELO AUGUSTO VIEIRA

## **RELÉ DIGITAL DE PROTEÇÃO UTILIZANDO ARDUINO**

Trabalho de Conclusão de Curso apresentado ao Centro de Ensino Superior de Conselheiro Lafaiete – CES-CL, como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Professor Especialista Paulo Nazareth

CONSELHEIRO LAFAIETE – MG

2017

Marcelo Augusto Vieira

## **RELÉ DIGITAL DE PROTEÇÃO UTILIZANDO ARDUINO**

Trabalho de Conclusão de Curso apresentada ao Centro de Ensino Superior de Conselheiro Lafaiete – CES-CL, como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica.

Conselheiro Lafaiete – MG, 2017.

DATA DE APROVAÇÃO: \_\_\_\_/\_\_\_\_/\_\_\_\_

---

Professor Especialista Paulo Rogério Pinheiro Nazareth - Orientador

---

Professor MSc Alexsandro de Souza Oliveira

---

Professor MSc Marcos Antônio Abdalla Júnior

Dedico este trabalho à minha família e aos meus amigos que sempre estiveram ao meu lado durante esta caminhada, me apoiando e dando força para que fosse possível chegar ao final com o objetivo cumprido.

## **AGRADECIMENTOS**

Obrigado a Deus.

À toda turma de Engenharia Elétrica 2013-2017.

Ao professor Paulo Nazareth.

Aos demais funcionários desta instituição.

Companheiros de trabalho.

Aos amigos.

Aos meus Familiares.

## RESUMO

O sistema elétrico e seus equipamentos e componentes estarão sempre sujeitos a ocorrência de falhas que poderão trazer sérios danos para o próprio sistema, além de trazer riscos à segurança das pessoas, por isso é de extrema importância que se tenham dispositivos de proteção, que na ocorrência de faltas, deverão eliminá-las de forma rápida e seletiva para que o trecho ou equipamento defeituoso seja desconectado do restante do sistema, mantendo o fornecimento de energia constante e garantindo a integridade de seus componentes. Observando a importância de um sistema de proteção, este trabalho visa o desenvolvimento do projeto de um relé de proteção digital de sobrecorrente, funções ANSI (*American National Standards Institute*) 50/51 sobrecorrente de fase e 50/51 GS (*Ground Sensor*) sobrecorrente à terra, utilizando da plataforma ARDUINO. As funções de nomenclatura 50 são denominadas instantâneas, ou seja, quando a corrente ultrapassar um valor ajustado ela atua instantaneamente. Já as funções de nomenclatura 51 são denominadas temporizadas, após a corrente ultrapassar o valor ajustado, quanto maior for essa corrente menor deverá ser o tempo de atuação. Podemos concluir que as funções de sobrecorrente são as mais utilizadas em qualquer sistema elétrico de potência, estando presente na maioria das cargas consumidoras, portanto criar dispositivos que exerçam essas funções com eficiência e custo reduzido tem se tornado um desafio.

**Palavras-chave:** Relé. Proteção. Arduino. Eficiência. Sobrecorrente.

## **ABSTRACT**

The electrical system and its equipment and components will always be subject to occurrence of failures that can cause serious damage to the system itself, in addition to bringing risks to the safety of people, so it is extremely important to have protection devices, which in the occurrence should eliminate them quickly and selectively so that the piece of equipment or faulty equipment is disconnected from the rest of the system, keeping the power supply constant and ensuring the integrity of its components. Observing the importance of a protection system, this work is aimed at developing the design of a digital overcurrent protection relay, ANSI (American National Standards Institute) 50/51 phase overcurrent and 50/51 GS (Ground Sensor) overcurrent using the ARDUINO platform. The nomenclature functions 50 are called instantaneous, that is, when the current exceeds a set value it acts instantaneously. Since the functions of nomenclature 51 are called the inverse time curve, after the current exceeds the set value, the higher this current is, the less time should be the actuation time. We can conclude that the overcurrent functions are the most used in any electric power system, being present in most consumer loads, so creating devices that perform these functions with efficiency and low cost has become a challenge.

Keywords: Relay. Protection. Arduino. Efficiency. Overcurrent.

## LISTA DE ABREVIATURAS E SIGLAS

A – Ampere

ABNT - Associação Brasileira de Normas Técnicas

ADC - Conversor Analógico Digital

ANSI - *American National Standards Institute*, Instituto Nacional Americano de Padrões

CI – Circuito Integrado

DDP - Diferença De Potencial

DT – Dial de Tempo

GPS - *Global Positioning System*, Sistema de Posicionamento Global

GS - *Ground Sensor*, Sensor de Terra

Hz – Hertz

IDE - *Integrated Development Environment*, Ambiente de Desenvolvimento Integrado

IEC - *International Electrotechnical Commission*, Comissão Eletrotécnica Internacional

KB - Kilobyte

LCD - *Liquid Crystal Display*, Tela de Cristal Líquido

LED - *Light Emitting Diode*, Diodo Emissor de Luz

ma – miliamperes

MHz - Mega-Hertz

ms – milissegundos

P – Potência Ativa

Q – Potência Reativa

S – Potência Aparente

s - segundos

SEP- Sistema Elétrico de Potencia

TC - Transformadores de Corrente

TP - Transformador de Potencial

μs - microssegundos

V - Volts

VA - Volt Ampere

VAr - Volt Ampere Reativo

VSM - *Virtual System Modelling*, Sistema Virtual de Modelagem

W - Watts



## LISTA DE FIGURAS

FIGURA 01: Triângulo de potência .....	16
FIGURA 02: Sistema de Proteção .....	20
FIGURA 03: Curva Característica de um Fusível .....	21
FIGURA 04: Relé Eletromecânico de Indução .....	25
FIGURA 05: Relé Eletrônico .....	26
FIGURA 06: Relé de Proteção Digital .....	27
FIGURA 07: Curva de Tempo Definido .....	30
FIGURA 08: Curva de Tempo Normal Inverso .....	31
FIGURA 09: Curva de Tempo Muito Inverso .....	32
FIGURA 10: Curva de Tempo Extremamente Inverso .....	33
FIGURA 11: Arduino Mega .....	36
FIGURA 12: Motor <i>Shield</i> .....	37
FIGURA 13: Esquema de Ligação do Sensor SCT .....	40
FIGURA 14: Esquemático do Projeto Montado no <i>Software</i> Proteus .....	42
FIGURA 15: Modulo relé 5V, 8 canais .....	43
FIGURA 16: Montagem do Relé – Parte de Trás.....	44
FIGURA 17: Frente do Relé Usando Arduino .....	45
FIGURA 18: Unidade de Ensaio de Relés .....	46
FIGURA 19: Circuito Prático Montado para Simulação de Sobrecarga e Verificação de Atuação do Relé.....	49

## **LISTA DE TABELAS**

TABELA01. Precisão de Medição dos Sensores de Corrente – Erro Percentual .....	41
TABELA 02. Tempos de Atuação – Temporização por Curva de Tempo Inverso.....	46
TABELA 03. Tempo de Atuação - Temporização por Tempo Definido .....	47

## **LISTA DE GRÁFICOS**

GRÁFICO 01. Curva de Atuação de Tempo Muito Inverso – Comparação entre ideal e obtido.....	47
GRÁFICO 02. Curva de Atuação por Tempo Definido – Comparação entre valores ajustados e obtidos.....	48

# SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>13</b>
<b>2 REFERENCIAL TEÓRICO .....</b>	<b>15</b>
<b>2.1 Corrente Elétrica.....</b>	<b>15</b>
<b>2.2 Tensão Elétrica .....</b>	<b>15</b>
<b>2.3 Potência.....</b>	<b>16</b>
2.3.1 Potência Ativa.....	16
2.3.2 Potência Reativa.....	17
2.3.3 Potência Aparente.....	17
<b>2.4 Transformadores de Instrumentos.....</b>	<b>18</b>
2.4.1 Transformadores de Corrente (TC) .....	18
2.4.2 Transformador de Potencial (TP) .....	19
2.4.3 Transdutores .....	19
<b>2.5 Sistemas de Proteção Elétrica .....</b>	<b>20</b>
<b>2.6 Dispositivos de Proteção.....</b>	<b>21</b>
2.6.1 Fusíveis de Proteção .....	21
2.6.2 Relés de proteção .....	22
<b>2.7 Requisitos Básicos de um Relé de Proteção.....</b>	<b>22</b>
2.7.1 Sensibilidade .....	23
2.7.2 Seletividade .....	23
2.7.3 Velocidade .....	23
2.7.4 Confiabilidade .....	24
<b>2.8 Tipos de Relés de Proteção.....</b>	<b>24</b>
2.8.1 Relés Eletromecânicos de Indução.....	25
2.8.2 Relés Eletrônicos .....	26
2.8.3 Relés Digitais .....	27
<b>2.9 Relés de Sobrecorrente.....</b>	<b>28</b>
2.9.1 Ajustes de um relé de sobrecorrente .....	28
2.9.2 Relé de Sobrecorrente de Fase .....	29
2.9.3 Relé de Sobrecorrente de Neutro.....	29
<b>2.10 Tipos de Curvas de Atuação.....</b>	<b>29</b>

2.10.1 Tempo Definido .....	30
2.10.2 Tempo Normal Inverso .....	30
2.10.3 Tempo Muito Inverso .....	32
2.10.4 Tempo Extremamente Inverso .....	33
<b>2.11 Arduino.....</b>	<b>34</b>
2.11.1 Hardware .....	34
2.11.2 Software .....	35
2.11.3 Arduino Mega .....	35
2.11.4 Pinos de entrada e saída Digital e Analógica.....	36
2.11.5 <i>Shields</i> .....	37
<b>3 METODOLOGIA .....</b>	<b>39</b>
<b>4 RESULTADOS E DISCUSSÃO .....</b>	<b>40</b>
<b>5 CONSIDERAÇÕES FINAIS .....</b>	<b>50</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>51</b>
ANEXO A – TABELA ANSI .....	53
APÊNDICE A - CÓDIGO UTILIZADO NO ARDUINO PARA TEMPORIZAÇÃO COM CURVA DE TEMPO INVERSO .....	59
APÊNDICE B - CÓDIGO UTILIZADO NO ARDUINO PARA TEMPORIZAÇÃO COM TEMPO DEFINIDO .....	69

## 1 INTRODUÇÃO

Para entender a função do sistema de relés de proteção, deve-se estar familiarizado com a natureza e modos de operação de um sistema de potência. A energia elétrica é um dos recursos fundamentais da sociedade moderna que está disponível a qualquer momento, na tensão e frequência corretas e na quantidade exata que o consumidor necessita. Este desempenho notável é alcançado através de planejamento, projeto, construção e operação cuidadosos de uma complexa rede elétrica composta por geradores, transformadores, linhas de transmissão e distribuição e outros equipamentos auxiliares. Para um consumidor o sistema elétrico parece comportar-se sempre em estado permanente: imperturbável, constante e capacidade inesgotável. Entretanto, o sistema de potência está sujeito a constantes distúrbios criados pelas variações aleatórias das cargas, pelas faltas oriundas de fenômenos naturais, e em alguns casos como resultados de falhas de equipamentos ou humanas. Apesar destas constantes perturbações o sistema elétrico se mantém num estado quase permanente devido a dois fatores básicos: o tamanho das cargas ou geradores individuais é muito pequena em relação ao tamanho do sistema, o segundo se dá devido à ação rápida e correta dos equipamentos de proteção quando da ocorrência de perturbações.

Um sistema de proteção detecta uma condição anormal de um sistema de potência e inicia uma ação corretiva tão rapidamente quanto possível para que o sistema de potência não seja levado para fora do seu estado normal. A rapidez de resposta é essencial em um sistema de proteção, podendo ter tempos da ordem de uns poucos milissegundos. A atuação de um sistema de proteção deve ser automática, rápida e restringir ao mínimo a região afetada. Em geral, relé de proteção não evita danos nos equipamentos, ele opera após a ocorrência de algum tipo de distúrbio que já pode ter provocado algum dano. As funções, portanto, são: limitar os danos, minimizar o perigo às pessoas, reduzir o “stress” em outros equipamentos e, acima de tudo, manter a integridade e estabilidade do restante do sistema elétrico, facilitando o restabelecimento.

Os relés de proteção foram os primeiros automatismos utilizados em sistemas elétricos de potência. Até a década de 70 os relés com o princípio eletromecânico dominaram completamente o mercado de dispositivos de proteção.

Os primeiros relés de proteção de concepção eletrônica foram introduzidos no final da década de 50. O desenvolvimento e aperfeiçoamento desses dispositivos utilizando componentes discretos cresceu durante a década de 60, tendo como principal foco melhorar a exatidão, a velocidade e o desempenho em geral. Entretanto, a excessiva quantidade de componentes, além da sua susceptibilidade à variações das condições ambientais, seu desempenho não era superior aos equivalentes eletromecânicos devido à robustez do mesmo. A consolidação deste tipo de relés só veio a ocorrer na década seguinte quando começaram a utilizar os circuitos integrados (CI), devido a diminuição de componentes e consequentemente das conexões associadas. O surgimento de componentes altamente integrados e a sua utilização na construção de relés de proteção permitiu aumentar a gama de funções. O desenvolvimento de microprocessadores com memórias de alta velocidade levaram a um rápido crescimento de computadores pessoais durante a década de 80. Essas novas tecnologias foram também utilizadas para o desenvolvimento de relés de proteção, que são os denominados relés digitais.

Diante de tais fatos esse trabalho de conclusão de curso teve como meta realizar estudos e análises a respeito de sobrecorrente no sistema elétrico de potência e com base nos relés digitais microprocessados já existentes, utilizar destes princípios para desenvolver um relé de sobrecorrente utilizando o ARDUINO, uma plataforma com uma infinidade de recursos que pode ser programada para a realização dos mais diversos tipos de projetos de automação, desde os mais simples aos que requerem um conhecimento mais avançado do software.

O projeto consiste basicamente em enviar os sinais de correntes, amostrados através de TC (Transformadores de Corrente), e implementar um programa lógico no ARDUINO que monitore estes sinais de corrente, e quando os mesmos saírem fora da faixa de valor ajustada como referência, o dispositivo entenda que está ocorrendo alguma falha e então desligue o circuito, eliminando o defeito e mantendo a integridade do restante do sistema elétrico, permitindo assim um fornecimento de energia ininterrupto e confiável.

## 2REFERENCIAL TEÓRICO

### 2.1 Corrente Elétrica

Corrente elétrica é o fluxo ordenado dos elétrons percorrendo um circuito fechado (CARVALHO, 2015).

Para que haja circulação de corrente elétrica é necessário que haja uma DDP (diferença de potencial), e a intensidade dessa corrente irá depender diretamente do número de elétrons que passa em uma determinada região do condutor por uma unidade de tempo, como pode ser visto na equação a seguir;

$$i = \frac{dq}{dt} \quad [^1]$$

Onde:

q é a carga elétrica em coulombs (C);

t é o tempo em segundos (s);

i é a corrente em ampères (A).

### 2.2 Tensão Elétrica

De acordo com Santos (2011, p.25) “a tensão elétrica é a diferença de potencial gerada entre dois pontos quaisquer. Essa diferença é responsável por colocar em movimento ordenado as cargas elétricas livres do meio do condutor”.

---

<sup>1</sup> Corrente elétrica - CARVALHO, 2015.



A d.d.p vai depender diretamente do quanto um ponto se encontra mais carregado eletricamente que o outro, ou seja, quanto maior essa diferença entre dois pontos maior será a tensão elétrica.

## 2.3 Potência

Potência é todo o trabalho realizado por uma determinada corrente em um intervalo específico de tempo. Dentro deste conceito, esta pode ser classificada como potência ativa, reativa e aparente. Segue abaixo o triângulo de potência.

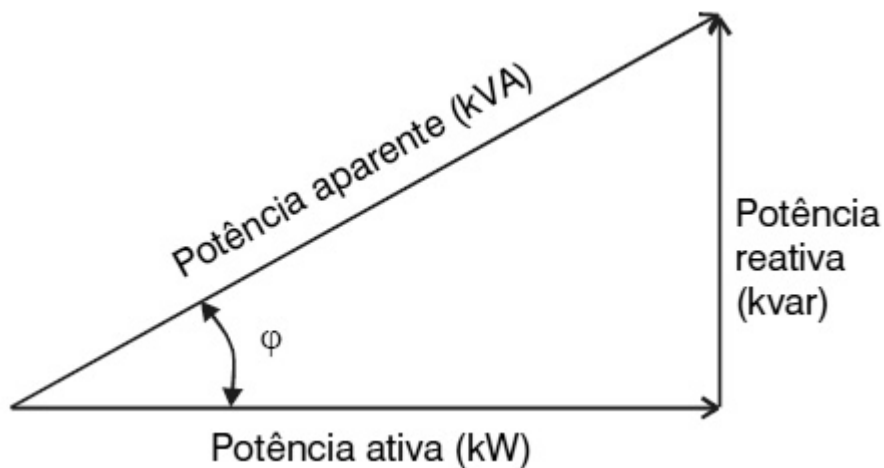


FIGURA 01. Triângulo de potência  
Fonte: Engeletrica, 2017.

### 2.3.1 Potência Ativa

De acordo com Engeletrica (2011), potência ativa é aquela que efetivamente realiza trabalho gerando luz, calor ou movimento e sua unidade de medida é o Watt(W).

$$P = V I \cos \varphi$$

[<sup>2</sup>]

<sup>2</sup> Potência ativa - ENGELETRICA, 2011 <<http://www.engeletrica.com.br>>

Onde:

V= tensão da rede em Volts (V);

I = corrente elétrica em A;

$\varphi$  = ângulo de fase entre a tensão e a corrente;

P = potência ativa em Watts (W);

### 2.3.2 Potência Reativa

É utilizada para criar e manter campos eletromagnéticos das cargas indutivas e é medida em VAR conforme (ENGELETRICA, 2011).

$$Q = V I \text{ sen.}\varphi \quad [^3]$$

Onde:

V= tensão da rede em V;

I = corrente elétrica em A;

$\varphi$  = ângulo de fase entre a tensão e a corrente;

Q = potência reativa em volt ampères reativo (Var);

### 2.3.3 Potência Aparente

De acordo com Carvalho (2015, p.24) “Produto dos valores eficazes de tensão e corrente denominamos a potência aparente que nada mais é que a raiz da soma dos quadrados da potência ativa e reativa”.

$$S = \sqrt{P^2 + Q^2} \quad [^4]$$

---

<sup>3</sup> Potência Reativa- ENGELETRICA, 2011 <<http://www.engeletrica.com.br>>

<sup>4</sup> Potência Aparente - Carvalho (2015, p.24)

Onde:

S= potência aparente em Volt ampère (VA);

Q=potência reativa em Var;

P= potência ativa em W;

## **2.4 Transformadores de Instrumentos**

Os transformadores de instrumento, são equipamentos que permitem aos instrumentos de medida e proteção funcionarem de uma forma adequada sem que seja necessário estar diretamente conectados ao circuito principal, onde as correntes e tensões são de valores elevados.

### **2.4.1 Transformadores de Corrente (TC)**

São equipamentos destinados a transformar, por meio de conversão eletromagnética, correntes de elevados valores, que circulam em seu primário, em correntes secundárias de poucos ampères, mantendo uma relação de transformação.

Destinam-se a evitar a conexão direta de medidores e relés, nos circuitos de corrente alternada de alta tensão, bem como adaptar a grandeza a ser mediada às faixas usuais de aparelhagem. O enrolamento primário tem geralmente poucas espiras, às vezes mesmo uma única; ao contrário, o enrolamento secundário tem maior número de espiras e a ele são ligadas as bobinas dos diversos medidores e/ou relés (CAMINHA, 1977, p.69).

Os transformadores de corrente, também chamados de transformadores de instrumentos, são utilizados em aplicações de alta tensão, onde circulam, frequentemente, baixas correntes, fornecem correntes suficientemente reduzidas e isoladas do circuito primário de forma a possibilitar o seu uso por equipamentos de medição, controle e proteção. Normalmente as correntes nos secundários dos TC's são padronizadas em 1 ou 5 A, sendo a grande maioria 5A.

### 2.4.2 Transformador de Potencial (TP)

Os transformadores de potencial, são equipamentos destinados a converter elevados potenciais de tensão em valores reduzidos seguindo uma relação de transformação.

Os transformadores de potencial, na sua forma mais simples, possuem um enrolamento primário de muitas espiras e um enrolamento secundário através do qual se obtém a tensão desejada, normalmente padronizada em 115V ou  $115/\sqrt{3}$  V (MAMEDE, 2011, p.54).

Com a utilização de tensões reduzidas nos secundários dos TP's se torna mais fácil utilizar esta amostra de tensão, permitindo assim que os equipamentos de medição e proteção tenham uma classe de isolamento inferior ao da rede em que estão ligados.

### 2.4.3 Transdutores

Os transdutores são dispositivos capaz de transformar os sinais obtidos por sensores que monitoram uma determinada variável, como por exemplo temperatura ou pressão, em outra forma de sinal, como elétrico (4 a 20 ma), que são utilizados em sistemas de controle e automação.

Segundo Almeida (2004, p.01), “O transdutor é qualquer dispositivo capaz de transformar um tipo de sinal em outro para permitir o controle de processos físicos, ou realizar uma medição, etc.”

Existem atualmente inúmeros tipos de transdutores que convertem as mais variadas formas de grandezas em outra forma de sinal de saída para serem utilizados nos processos de medição, controle e automação de sistemas.

## 2.5 Sistemas de Proteção Elétrica

Os componentes elétricos de um sistema devem ser protegidos contra possíveis sobrecorrente ou curto-circuito que venham a ocorrer em algum ponto do sistema.

Como afirma Sato (2005) na ocorrência de algum destes distúrbios é necessário que a parte afetada seja rapidamente isolada do restante da rede elétrica, afim de evitar danos materiais e facilitar a recuperação do sistema.

Para isso são desenvolvidos os sistemas de proteção, cujo princípio básico de funcionamento pode ser visto a seguir (Figura 02):

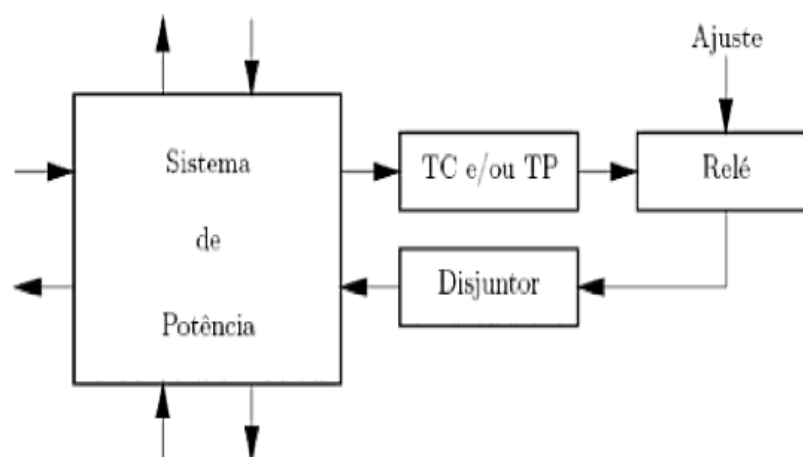


FIGURA 02. Sistema de Proteção

Fonte: Sato, 2005.

Os parâmetros do sistema são monitorados constantemente pelos transformadores de instrumentos (TC's e TP's) que enviam sinais analógicos transformados em grandezas secundárias para um sistema de decisões lógicas (relé de proteção), que irá comparar o valor medido com um valor previamente ajustado no relé. A atuação do relé ocorrerá sempre que o valor medido ultrapassar o valor ajustado, comandando assim a abertura do disjuntor, fazendo com que a corrente de falta seja eliminada.

## 2.6 Dispositivos de Proteção

Basicamente existem dois tipos de dispositivos usados para proteção de sistemas elétricos, que são os fusíveis e os relés. Os fusíveis são dispositivos que operam através da fusão de um elemento metálico construído de maneira a suportar um determinado valor de corrente, ou seja, possui uma característica típica de tempo x corrente. Já os relés constituem uma ampla gama de dispositivos oferecendo proteção contra diversas formas de distúrbios como: sobrecarga, curto-circuito, sobretensão e outras.

### 2.6.1 Fusíveis de Proteção

Segundo Niskier (2008, p.146) “o fusível é um dispositivo adequadamente dimensionado para interromper a corrente de sobrecarga ou curto-circuito”.

São dispositivos amplamente utilizados em ramais de alimentadores e em equipamentos (transformadores de distribuição e capacitores) ligados à rede de distribuição. As curvas (Figura 03) característica de tempo x corrente de um fusível são apresentadas na forma de tempo mínimo de fusão e tempo total de interrupção.

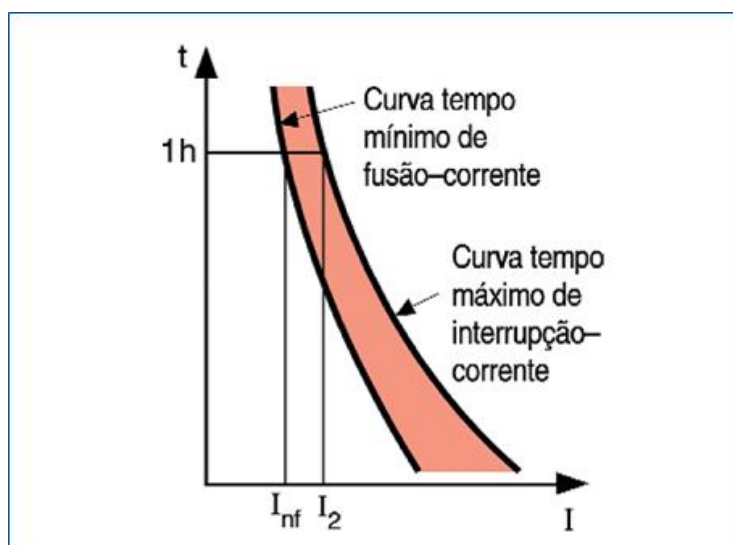


FIGURA 03. Curva Característica de um Fusível  
Fonte: Sato, 2005.

A figura acima mostra a curva característica de um fusível onde o tempo total de interrupção será dado pela soma do tempo mínimo de fusão com o tempo de arco.

### 2.6.2 Relés de proteção

Relés de proteção são dispositivos destinados a realizar a proteção de equipamentos elétricos, quando estes sofrem alterações nos seus parâmetros normais, podendo vir a causar danos ao equipamento ou ao sistema em que o mesmo se encontra ou até mesmo à segurança das pessoas.

Segundo a Associação Brasileira de Normas Técnicas (ABNT), “o relé é um dispositivo por meio do qual um equipamento elétrico é operado quando se produzem variações nas condições deste equipamento ou do circuito em que ele está ligado, ou em outro equipamento ou circuito associado”.

Estes dispositivos deverão atuar no menor tempo possível, comandando a abertura do disjuntor ou disjuntores que estejam alimentando o ponto em falta.

Existem diversos tipos de relés, para as mais variadas funções de proteção dependendo da grandeza a ser monitorada, como por exemplo os relés de sobrecorrente, de sobretensão e subtensão, de frequência, de potência, diferencial de corrente, de distância, entre outros diversos tipos. Além dos relés específicos para cada função, tem-se alguns modelos mais modernos que podem realizar diversas funções de proteção, são os chamados relés multifunção.

## 2.7 Requisitos Básicos de um Relé de Proteção

Para que um relé de proteção seja considerado eficiente, é necessário que ele atenda à alguns requisitos básicos que lhes garante uma boa performance. A sensibilidade, seletividade, velocidade e confiabilidade são indispensáveis em relés de alto desempenho.

### 2.7.1 Sensibilidade

Para Nazareth (2016, p.08) “Consiste na capacidade do elemento de proteção reconhecer com precisão a faixa e os valores indicados para a sua operação e não operação”.

A sensibilidade de um relé é de extrema importância, pois ele deve ser capaz de ler o valor de uma grandeza e comparar com o valor ajustado, e caso esteja fora da faixa de atuação ele efetue o disparo.

### 2.7.2 Seletividade

Seletividade é a capacidade que uma proteção tem de identificar um defeito e atuar de modo a isolar apenas a parte defeituosa.

Para Niskier (2008), ao ocorrer um defeito em um ponto da instalação, é necessário que o desligamento afete uma parte mínima da mesma. Para que isto aconteça, é necessário que a proteção mais próxima do defeito venha a ser a primeira a atuar.

Para que um sistema de proteção seja seletivo é preciso que haja uma coordenação entre os tempos de atuação, de modo que os tempos de atuação cresçam a medida que as proteções se tornem mais afastadas das cargas e mais próximas da fonte geradora.

### 2.7.3 Velocidade

Quando ocorre uma falha em um determinado ponto do circuito, é de extrema importância que a mesma seja eliminada o mais rápido possível evitando maiores danos aos componentes do mesmo e ao restante do sistema.



É, geralmente, desejável remover a parte atingida pela falta do restante do sistema de potência tão rapidamente quanto possível para limitar os danos causados pela corrente de curto-circuito; entretanto, existem situações em que uma temporização intencional é necessária (SATO; FREITAS, 2014).

A velocidade dos relés podem ser classificadas da seguinte maneira;

- Instantâneos: Quando não se adiciona nenhuma temporização no relé. O tempo inerente fica na faixa de 17 a 100 milissegundos (ms).
- Temporizados: Uma temporização intencional é colocada no relé, entre o tempo de decisão do relé e o início da ação de desligamento.
- Alta Velocidade: Um relé que opera em menos de 50 ms (3 ciclos na base de 60Hz).
- Ultra Alta Velocidade: Possui temporização inferior 4 ms.

É necessário adicionar temporização aos relés de proteção quando se deseja fazer a coordenação entre funções de proteção, em outros casos, quando não há necessidade de coordenação com outro relé pode ser usada a unidade instantânea dependendo da proteção a ser feita.

#### 2.7.4 Confiabilidade

Segundo Nazareth (2016, p.08) confiabilidade “é a probabilidade do elemento de proteção cumprir com segurança e exatidão as funções que lhes foram confiadas”.

Um dispositivo de proteção deverá atuar somente quando for solicitado, e sempre que for solicitado, ou seja, não deverá atuar indevidamente e não pode deixar de atuar quando for preciso.

### 2.8 Tipos de Relés de Proteção

Durante várias décadas os dispositivos de proteção veem sendo melhorados de acordo com a evolução da tecnologia, para que se tenham equipamentos altamente confiáveis, como é o caso da geração dos relés digitais, os mais modernos existentes no mercado.

### 2.8.1 Relés Eletromecânicos de Indução

São dotados de bobinas, disco de indução, molas, contatos fixos e móveis que lhes proporcionam uma grande robustez. “Dado seu mecanismo de operação, são tidos como verdadeiras peças de relojoaria” (MAMEDE, 2011, p.8).

Sua construção mecânica baseia-se no princípio da indução, onde uma bobina ao ser percorrida por uma corrente elétrica faz com que o disco gire e o contato móvel feche com o contato fixo, provocando a atuação do relé. A seguir (Figura 04) está um modelo bem comum de relé eletromecânico.



FIGURA 04. Relé Eletromecânico de Indução

Fonte: Própria, 2017.

Atualmente não são mais fabricados, devido seu princípio de funcionamento ter se tornado arcaico, entretanto são facilmente encontrados em subestações de concessionárias de energia elétrica e em fábricas mais antigas.

## 2.8.2 Relés Eletrônicos

Segundo Mamede (2011, p.9) são constituídos de circuitos integrados dedicados a cada função desempenhada. Os ajustes são realizados através de diais fixados na parte frontal do relé, sendo cada dial com uma função de ajuste, corrente, tensão, tempo etc. São também conhecidos como relés estáticos e acabaram tendo o mesmo destino dos eletromecânicos de indução.

Trouxeram pouca inovação aos sistemas de proteção, basicamente as funções desempenhadas pelos relés eletromecânicos, através de peças mecânicas e indução magnética, foram reproduzidas nos relés eletrônicos utilizando-se circuitos impressos. No entanto sua susceptibilidade as interferências ambientais como poeira, umidade e outras, faziam com que não fossem superior aos seus equivalentes eletromecânicos, o que fez com que parassem de ser produzidos logo com a chegada dos relés digitais.

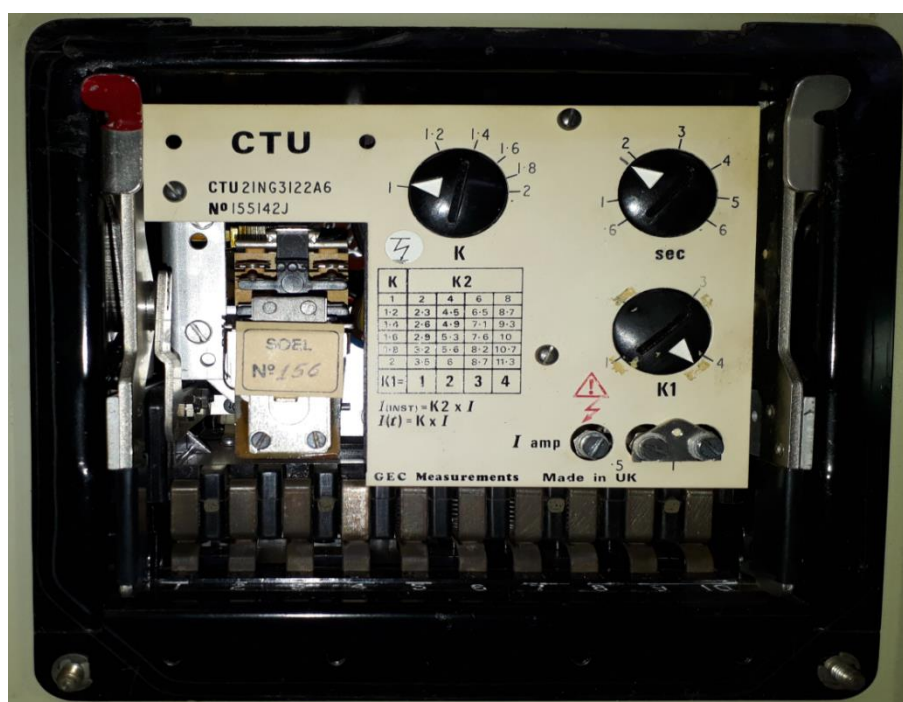


FIGURA 05. Relé Eletrônico

Fonte: Própria, 2017.

A figura (Figura 05) mostra um relé eletrônico de sobrecorrente muito utilizado, onde é possível ver os potenciômetros de ajuste de corrente e de tempo.

### 2.8.3 Relés Digitais

São constituídos de circuitos eletrônicos providos de *chips* de alta velocidade de processamento, funcionam por meio de programas dedicados que processam as informações que chegam por meio dos transformadores de medida TC's e TP's.

Os relés digitais revolucionaram os esquemas de proteção, oferecendo vantagens impossíveis de serem obtidas dos seus antecessores. Além das funções de proteção propriamente ditas, os relés digitais realizam funções de comunicação, medidas elétricas, controle, sinalização remota, acesso remoto etc. (MAMEDE, 2011, p.10).

Atualmente dominam o mercado devido sua alta confiabilidade e velocidade de processamento de dados, além de preços competitivos, e por terem a capacidade de realizar diversas funções de proteção dependendo do modelo utilizado, além de possuírem interface de comunicação com sistemas supervisórios, o que permite acompanhar remotamente o estado do relé com indicações de *trip* ou falhas.



FIGURA 06. Relé de Proteção Digital

Fonte: Própria, 2017.

Acima (Figura 06) está um exemplo de relé digital, no entanto são encontrados diversos modelos de relés digitais para as mais variadas funções, sendo que a grande maioria exerce mais de uma função de proteção. Uma das principais características que diferencia um relé digital são seus sinais de entrada, alguns possuem apenas entradas de corrente outros apenas de tensão, e os modelos mais completos possui ambas entradas.

## 2.9 Relés de Sobrecorrente

Para Mamede (2011, p.97), “entende-se por relé de proteção de sobrecorrente aquele que responde à corrente que flui no elemento do sistema que se quer proteger quando o módulo dessa corrente supera o valor previamente ajustado”.

São considerados como a proteção mínima a ser garantida à um sistema elétrico, normalmente utilizados em alimentadores de média tensão, geradores, motores etc. A proteção com relé de sobrecorrente é a mais econômica de todas as proteções utilizadas em sistemas elétricos, além de ser grande a variedade de relés que desempenham esta função.

### 2.9.1 Ajustes de um relé de sobrecorrente

Os relés de sobrecorrente temporizados possuem basicamente dois pontos a serem ajustados, de acordo com a carga e com a coordenação do sistema, são o *tape* e o “dial” de tempo (DT). O *tape* é ajustado em função da corrente, e o DT é selecionado de acordo com as temporizações requeridas para a coordenação. O valor do *tape* determina a corrente mínima capaz de iniciar a operação do relé, a chamada corrente de *pick-up*.

Segundo Kindermann (1999), “*pick-up* é o termo genérico designado para a menor corrente que é possível atrair o êmbolo ou alavanca, ou seja, fazer o relé operar”.

Para os relés digitais, apesar de não terem partes moveis, a analogia é a mesma, ou seja, o *pick-up* irá determinar sua corrente mínima de atuação, e o DT deverá ser escolhido levando

em conta os tempos ajustados em outros relés que estejam no mesmo circuito, para que seja feita a coordenação.

### 2.9.2 Relé de Sobrecorrente de Fase

Os relés de sobrecorrente de fase são destinados a realizar a proteção contra curto-circuito trifásico e bifásico, são identificados pela nomenclatura ANSI 50/51, sendo a função 50 para a unidade instantânea e 51 para a unidade temporizada de sobrecorrente.

### 2.9.3 Relé de Sobrecorrente de Neutro

Os relés de sobrecorrente de neutro, ou relé de terra, deve atuar para curto-circuito monofásico (ou fase-terra), são identificados pela nomenclatura ANSI 50/51N ou 51GS, podendo este ser ajustado para um valor de partida muito baixo, o que permite que ele atue para curto-circuito monofásico com alta resistência.

## 2.10 Tipos de Curvas de Atuação

As características de resposta dos relés de sobrecorrente são alocadas em um gráfico em função de múltiplo da corrente de *tape* (A) versus tempo (segundos), para cada ajuste dial de tempo. Há uma família de curvas IEC (*International Electrotechnical Commission*), cujas declividades mais usuais são denominadas como se segue, entretanto existe uma infinidade de tipos de curvas.

### 2.10.1 Tempo Definido

Como o próprio nome já diz, este tipo de curva apresenta um único valor de tempo, que pode ser ajustado, porém, é invariável em função da corrente de sobrecarga ou curto-circuito.

Segundo Sato (2005, p.12) “o relé de sobrecorrente de tempo definido tem uma característica tempo versus corrente plana”.

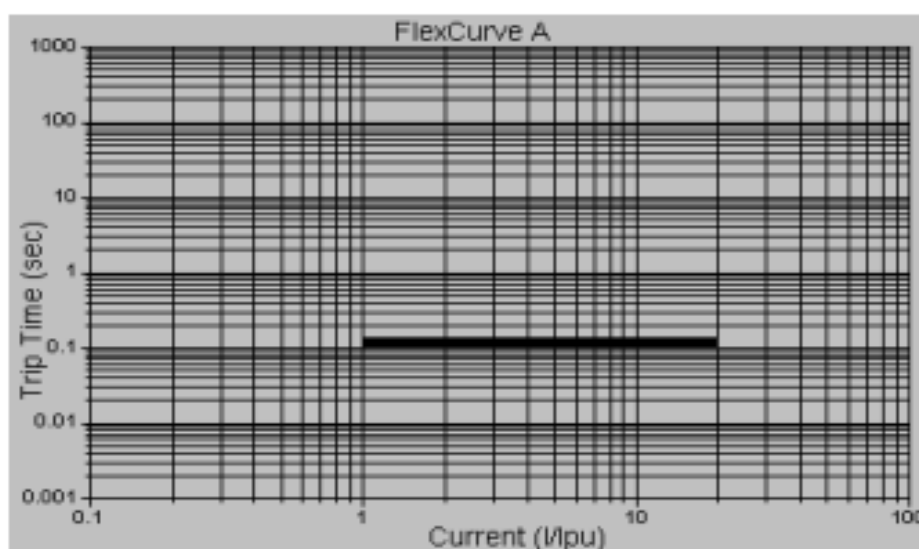


FIGURA 07. Curva de Tempo Definido

Fonte: Casanova, 2008.

Portanto, como pode ser observado (Figura 07) acima, para valores de corrente acima da corrente de *pick-up* o relé sempre atuará com tempos praticamente iguais, independentemente do valor que seja essa corrente de falta.

### 2.10.2 Tempo Normal Inverso

Este tipo de curva geralmente é usada em redes onde a faixa de variação da corrente de curto-circuito é larga, causa essa decorrente da mudança da capacidade de geração.

Conforme Mamede (2011), nas usinas geradoras de energia à medida que a carga vai se reduzindo, como por exemplo, no período de 23 horas e 6 horas, retiram se algumas unidades de geração com o objetivo de economizar recursos como água, combustível etc. Em consequência o nível de curto-circuito do sistema pode diminuir acentuadamente. A equação padrão para o tempo de atuação deste tipo de curva é dada da seguinte maneira;

$$T = \frac{0,14}{\left(\frac{I_{ma}}{I_s}\right)^{0,02} - 1} \times T_{ms} \quad [^5]$$

Onde:

$I_{ma}$ – Sobrecorrente máxima admitida;

$I_s$ – Corrente de ajuste no relé (TAPE);

$T_{ms}$  – Multiplicador de tempo, ou índice de tempo (DT);

$T$ – Tempo de atuação.

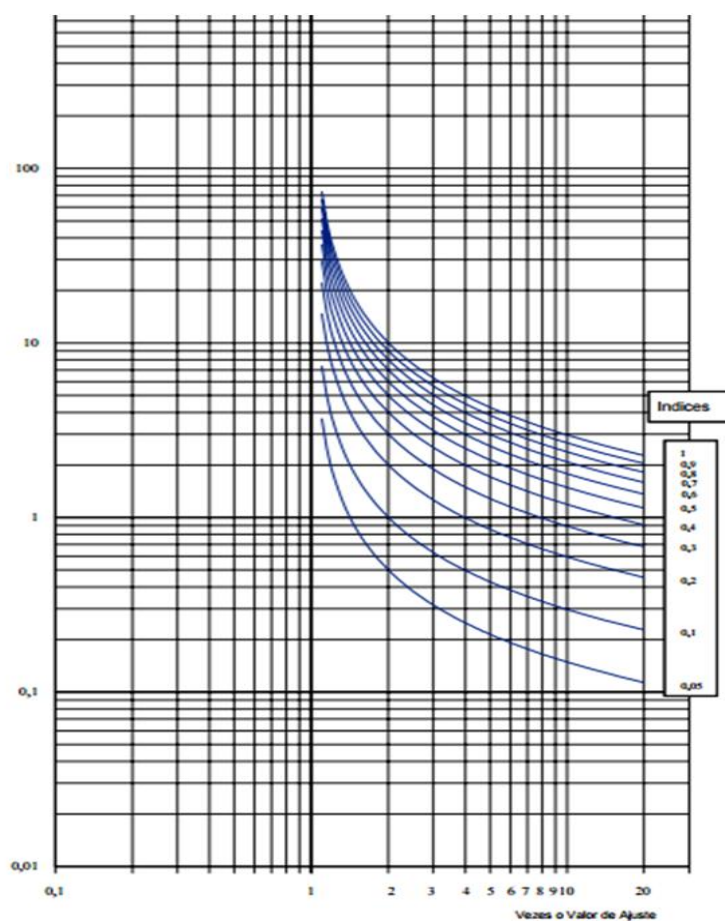


FIGURA 08. Curva de Tempo Normal Inverso

Fonte: Mamede,2011.

<sup>5</sup> Curva de Tempo Normal Inverso – MAMEDE FILHO (2011, pg.128)



A característica (tempo x corrente), relativamente plana permite que o relé opere com razoável rapidez para uma faixa grande de corrente de curto-circuito.

### 2.10.3 Tempo Muito Inverso

Possui uma característica de curva mais íngreme, que faz com que o relé opere lentamente para valores mais baixos de corrente e opere rapidamente para altas correntes de curto-circuito. O tempo de atuação é dado pela seguinte fórmula;

$$T = \frac{13,5}{\left(\frac{I_{ma}}{I_s}\right)^{-1}} \times T_{ms}$$

[<sup>6</sup>]

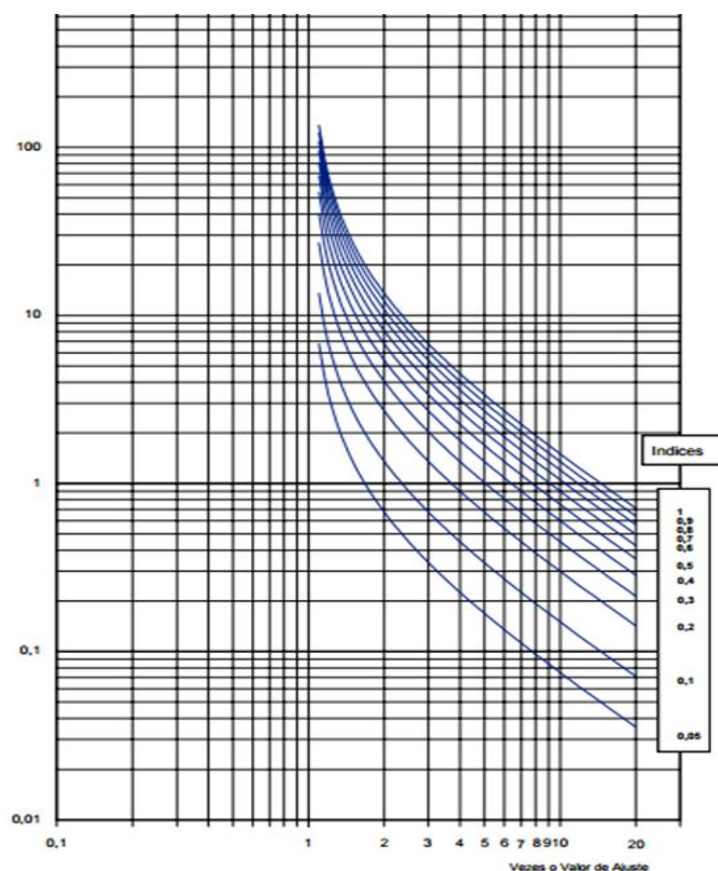


FIGURA 09. Curva de Tempo Muito Inverso  
Fonte: Mamede,2011.

<sup>6</sup> Curva de Tempo Muito Inverso - MAMEDE FILHO (2011, pg.129)

Este tipo de curva não é indicado para sistemas que possuam capacidade de geração variável, são empregados normalmente em sistemas onde a corrente de curto-circuito que circula no relé depende da localização do ponto onde ocorreu o defeito em relação ao relé.

#### 2.10.4 Tempo Extremamente Inverso

Apresenta uma característica bastante íngreme, similar a característica de um fusível. Portanto é bastante usada em sistemas onde possuam fusíveis como proteção, tornando a coordenação mais eficaz. O tempo de atuação é dado pela seguinte equação;

$$T = \frac{80}{\left(\frac{I_{ma}}{I_s}\right)^{-1}} \times T_{ms}$$

[<sup>7</sup>]

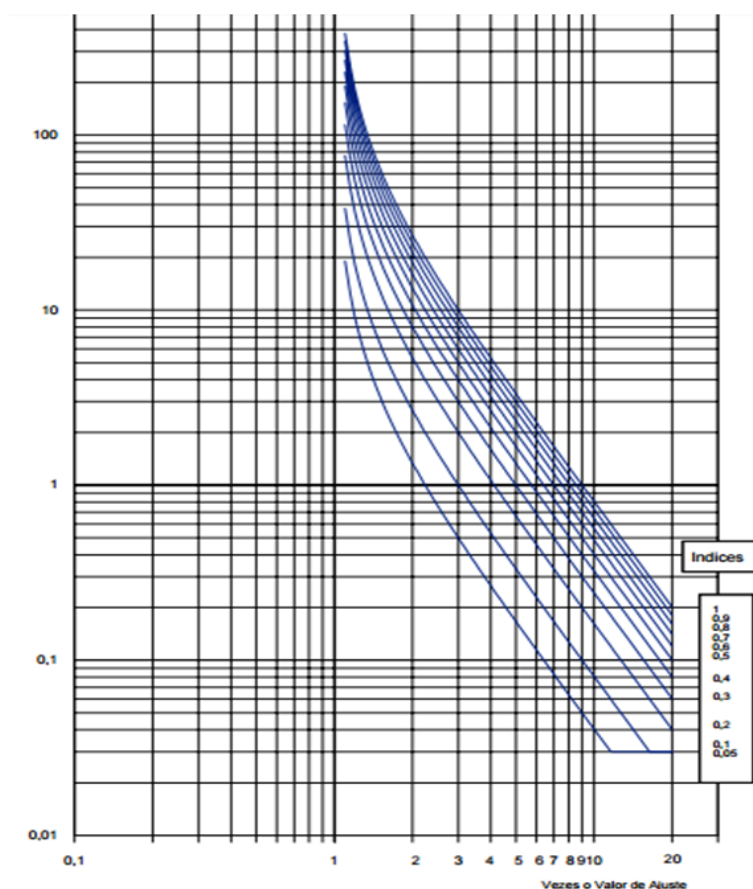


FIGURA 10. Curva de Tempo Extremamente Inverso

Fonte: Mamede, 2011.

<sup>7</sup> Curva de Tempo Extremamente Inverso - MAMEDE FILHO (2011, pg.129)

Devido sua inclinação bastante íngreme, são bastante adequadas a sistemas de distribuição, normalmente constituídas por cargas indutivas de pequeno porte, tais como refrigeradores, bombas, condicionadores de ar e outros, que em geral são ligadas ao mesmo tempo quando do retorno do alimentador após uma interrupção, provocando elevadas correntes transitórias.

## 2.11 Arduino

Um Arduino é um microcontrolador que pode ser programado para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele. Pode ser utilizado para desenvolver objetos interativos independentes, ou pode ser conectado a um computador, à uma rede, ou até mesmo à internet para recuperar e enviar dados do Arduino e atuar sobre eles.

Arduino é um projeto que engloba *software* e *hardware* e tem como objetivo fornecer uma plataforma fácil para prototipação de projetos interativos, utilizando um microcontrolador. Ele faz parte do que chamamos de computação física: área de computação em que o *software* interage diretamente com o *hardware*, tornando possível integração fácil com sensores, motores e outros dispositivos eletrônicos (Justen, 2011, p.08).

### 2.11.1 Hardware

Segundo Justen (pg.08), o *hardware* consiste em uma placa que cabe na palma da mão, basicamente é um computador como qualquer outro: possui microprocessador, memória RAM, memória *flash* (para guardar o *software*), temporizadores, contadores, dentre outras funcionalidades.

A velocidade de *clock* pode variar de acordo com o tipo de placa a ser utilizada, sendo que as versões mais completas tem velocidades de *clock* mais elevadas, o que garante um processamento mais rápido de informações e variáveis.

Basicamente a placa será responsável pelo processamento e controle do seu sistema, para isso ela interage com os componentes a ela conectados fazendo a leitura das variáveis de entrada e

executando o *software* desenvolvido e salvo em sua memória, para tomada de decisões e acionamento dos pinos de saída.

### 2.11.2 Software

No mundo do *software*, uma biblioteca é um pedaço de *software* que fornece algum tipo de funcionalidade, tal como escrever um texto em uma tela LCD (*Liquid Crystal Display*) ou calcular sua posição a partir de um sistema de navegação GPS (*Global Positioning System*). Bibliotecas de *software* trabalham da mesma forma que as bibliotecas de referência convencionais: você solicita uma informação e então a usa em seu projeto. No mundo do Arduino, uma biblioteca é um pedaço de *software* que você inclui em seu *sketch* e fornece algum tipo de funcionalidade.

### 2.11.3 Arduino Mega

Segundo Monk (2013, pg. 12) o Arduino Mega (Figura 11) até pouco tempo era o carro de alta performance das placas de Arduino. Ele ostenta uma grande coleção de portas de entrada e saída. Isso é feito engenhosamente colocando conectores extras em um dos lados da placa, de tal modo que a placa permanece compatível pino a pino com o Arduino Uno e todos os *shields* disponíveis de Arduino.

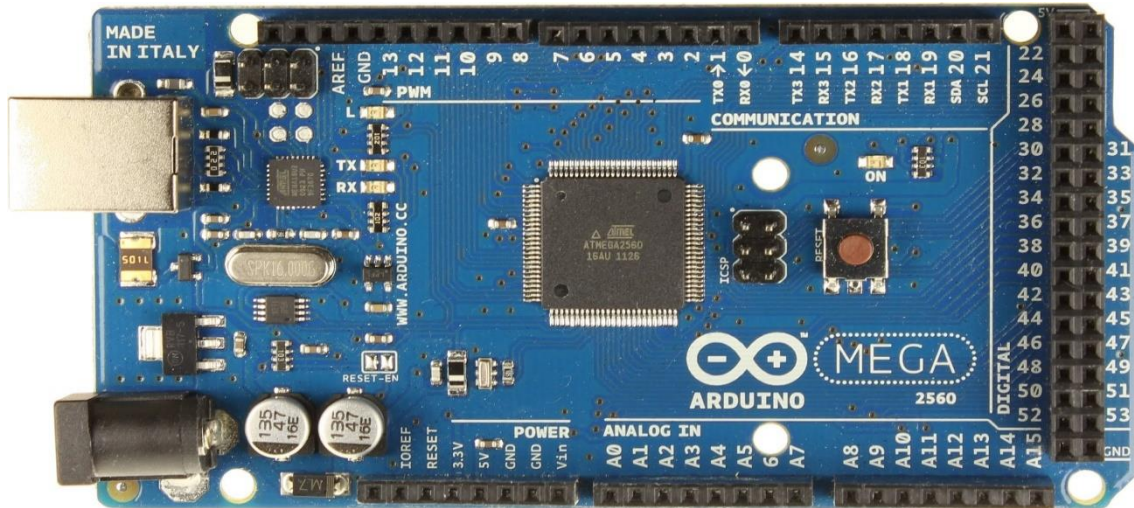


FIGURA 11. Arduino Mega

Fonte: Monk, 2013.

Ele usa o processador Atmega1280 que tem mais pinos de entrada e saída. Como esse *chip* é do tipo montagem superficial, isso significa que ele está fixado de forma permanente à placa e, diferentemente do Uno e de outras placas similares, você não poderá substituí-lo caso venha a danificá-lo acidentalmente.

Entre as características extras oferecidas pelo Mega estão:

- 54 pinos de entrada e saída
- 128 KB (Kilobyte) de memória *flash* para armazenar *sketches* e dados fixos
- 8KB de RAM e 4 KB de EEPROM
- Velocidade de *Clock* de 16 MHz (Mega-Hertz)

#### 2.11.4 Pinos de entrada e saída Digital e Analógica

No mundo digital, tudo possui dois estados; uma chave pode apenas estar ligada ou desligada, um LED (*Light Emitting Diode*) pode estar aceso ou não. Estes estados podem ser descritos de várias maneiras, como uns ou zeros, ligado ou desligado, alto ou baixo. O pino digital do Arduino funciona da mesma maneira, quando ajustado para saída, ele é de 0 ou 5 volts, ou seja, se está em zero significa que não tem tensão no pino, está desligado, quando está em cinco quer dizer que o pino está ativo e possui uma tensão de 5 volts.

Já no mundo analógico, as coisas possuem um intervalo de valores, ou seja, uma onda senoidal varia entre os valores máximos e mínimos, uma temperatura entre a máxima e a mínima. Os pinos analógicos do Arduino são utilizados para estes tipos de sinais variáveis, mas como ele é um dispositivo digital, então é necessário um meio para converter o sinal de entrada a uma representação digital, que é feito através de um conversor analógico para digital (ADC).

#### 2.11.5 Shields

Uma outra razão para o sucesso do Arduino é que ele não se limita a placas com microcontrolador. Há um número enorme de placas acessórias compatíveis com o Arduino denominadas *Shields* (Figura 12).

Segundo Justen (2011, pg.13 ) “ existem *shields* dos mais variados tipos, para as mais diversas funções. Alguns servem como entrada, outros como saída, e ainda outros como entrada e saída”.



FIGURA 12. Motor *Shield*  
Fonte: Arduino, 2017.

Com os *shields* conseguimos, por exemplo, fazer o Arduino se comunicar numa rede *Ethernet*, ou ainda transmitir dados para qualquer dispositivo via *Bluetooth* ou *Wi-Fi*. Essas placas são encaixadas diretamente por cima da placa do Arduino, há *shields* disponíveis

para praticamente qualquer aplicação que se possa imaginar, como segue alguns exemplos abaixo:

- *Ethernet – Shield* que dá recurso para que a placa de Arduino funcione como servidor de *web*
- Motor – para acionar motores elétricos
- *Relays – shield* que comanda relés a partir do seu Arduino
- USB – permite controle de dispositivos USB

Dentre outros com infinitas funcionalidades que o mercado atual oferece.

### 3 METODOLOGIA

Para o desenvolvimento deste trabalho foi utilizado no método do estudo e elaboração do projeto as seguintes técnicas:

- Pesquisou-se literaturas e também sites na internet sobre os assuntos englobados nesse estudo, para o desenvolvimento do projeto;
- Para o desenvolvimento da lógica de controle do sistema foi utilizado o *software* IDE (*Integrated Development Environment*) do Arduino versão 1.6.11, um *software* livre no qual você escreve o código na linguagem que o Arduino compreende (baseada na linguagem C). O IDE permite que você escreva um programa de computador, que é um conjunto de instruções, passo a passo, das quais você faz o *upload* para o Arduino, que então, executará essas instruções, interagindo com o que estiver conectado à ele. No mundo do Arduino, programas são conhecidos como *sketches* (rascunho, ou esboço);
- Para realizar os testes e simulações do circuito afim de verificar o funcionamento do código desenvolvido, foi utilizado o *software* ISIS Proteus VSM (*Virtual System Modelling*) versão 7.8sp2. O *software* permite o desenvolvimento de esquemas eletrônicos que torna possível a validação do circuito criado através de simulações do projeto. É possível criar e simular aplicações de diversos periféricos, incluindo microcontroladores;
- Foram realizados diversos testes e modificações no código que realiza a lógica de controle do circuito até chegar ao funcionamento desejado do mesmo, somente após isso foi realizada a montagem física do circuito;
- Com o circuito já montado foram realizadas simulações reais de defeitos para a comprovação da correta atuação das funções de proteção, assim como verificação do tempo de atuação, sinalização da respectiva proteção atuada e possíveis erros de atuação e medição das grandezas elétricas.



## 4 RESULTADOS E DISCUSSÃO

Para realizar as medições de corrente foi utilizado o sensor SCT 013 30, que tem uma capacidade de medir correntes de até 30 A, a saída do sensor é uma tensão alternada de 1 V que instintivamente induz a pensar sobre a leitura com as entradas analógicas do Arduino. No entanto, estes só podem ler tensões entre 0 e 5V. Lembrando que a tendência é pensar que o sensor responde com uma tensão de 0 a 1V correspondente à escala de 0 a 30A, mas na verdade o sinal do sensor oscila de -1 a + 1V, porque a corrente muda 60 vezes por segundo, o campo eletromagnético gerado inverte a direção com a mesma frequência e consequentemente a inversão da saída de sinal do sensor. Na realidade, portanto, temos um sinal -1 / + 1V correspondente à escala -30 / + 30A. A tensão produzida pelo sensor não pode ser lida diretamente pelo Arduino, pois também contém valores negativos. Então é necessário adicionar uma tensão em série para aumentar o seu valor base e trazer as leituras de volta ao alcance que o Arduino possa ler. Isso foi feito usando um divisor de tensão através de dois resistores iguais (Figura 13), então, a partir da tensão 5V do Arduino, obtemos uma de 2.5V como referência. Desta forma, passamos de -1 / +1V para 1,5 / 3.5V, valores facilmente legíveis pelas entradas analógicas do Arduino.

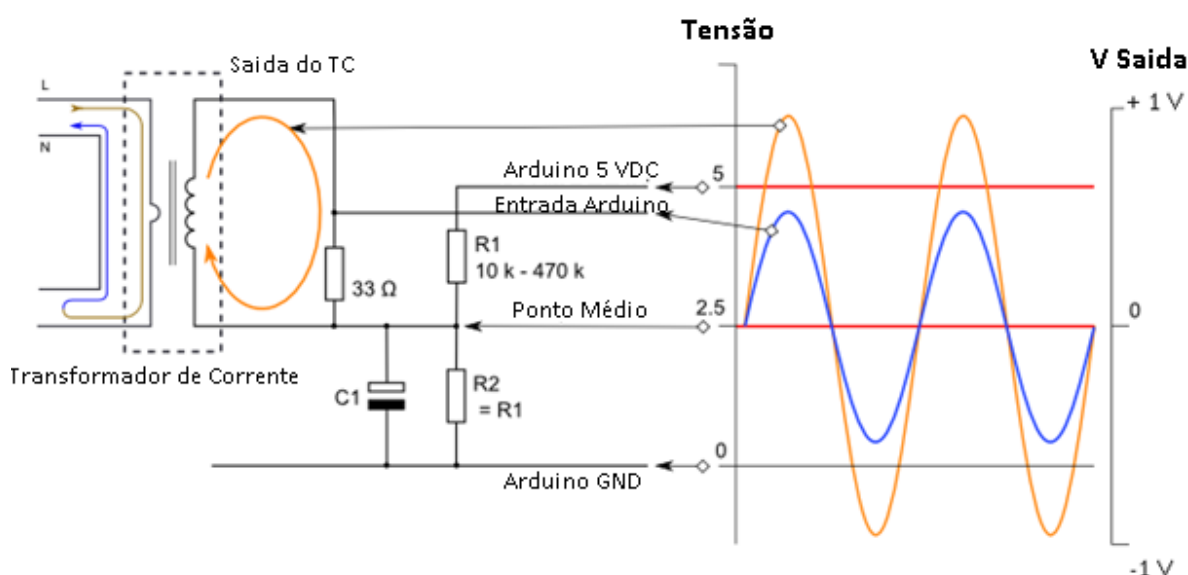


FIGURA 13. Esquema de Ligação do Sensor SCT

Fonte: OpenEnergyMonitor, 2017.<sup>8</sup>

<sup>8</sup> Ligação Sensor SCT 013 30 - OPENENERGYMONITOR, 2017 <<https://openenergymonitor.org/>>

Para que a medição seja precisa, devemos realizar várias amostragens ao longo de pelo menos um período inteiro (comprimento de onda formado pela semi-onda positiva e negativa) e realizar uma média quadrática dos valores obtidos. Para uma tensão na frequência de 60Hz a duração é de 0,016 segundos por ciclo, portanto em tão pouco tempo, o Arduino tem que fazer toda uma série de amostragens e cálculos. O Arduino, de um ponto de vista estritamente teórico, tem um conversor analógico-digital (ADC) capaz de amostragem a 9,6 kHz, ou seja, em apenas um segundo, ele pode executar 9600 medições, cada uma das quais dura 104 $\mu$ s (microssegundos). Mas na realidade, além da aquisição real do sinal, também temos que lidar com os cálculos e com os dados obtidos fazendo com que a velocidade real diminua, a tabela a seguir (Tabela 01) mostra os resultados das medições realizadas através do Arduino.

TABELA 01. Precisão de Medição dos Sensores de Corrente – Erro Percentual

Corrente Aplicada em (A)	Fase A		Fase B		Fase C		Neutro N	
	Valor Medido (A)	Erro (%)	Valor Medido (A)	Erro (%)	Valor Medido (A)	Erro (%)	Valor Medido (A)	Erro (%)
5	4,88	2,40	4,87	2,60	4,84	3,20	4,88	2,40
10	9,79	2,10	9,48	5,20	9,62	3,80	9,74	2,60
15	14,51	3,27	14,09	6,07	14,35	4,33	14,37	4,20
20	19,43	2,85	19,02	4,90	19,29	3,55	19,48	2,60
25	24,25	3,00	23,55	5,80	24,07	3,72	24,23	3,08
30	28,96	3,47	28,17	6,10	28,73	4,23	28,97	3,43
Erro Médio (%)		2,27	4,09		3,10		2,48	

Fonte: Própria, 2017.

Pela tabela observa-se que em relação as medições obtidas, estão totalmente aceitáveis para equipamentos de proteção, onde os TC's podem apresentar erros de medição de até 10%. Tendo em vista que a parte de medição do circuito apresentou erro médio abaixo de 5% em todos os sensores, o dispositivo se mostra bastante preciso e aceitável para a finalidade proposta. O fato de as medições da fase B se mostrarem sempre um pouco acima das demais se deve ao circuito divisor de tensão, que pode haver uma variação nas características construtivas dos valores de resistência e capacitância dos componentes usados, fato que foi confirmado trocando o sensor da fase B com o de outra fase, mesmo assim o erro continuou acima das demais, ficando claro que o problema não era no sensor, e sim o circuito eletrônico.

A simulação da funcionalidade do circuito foi feita utilizando-se o *software* ISIS Proteus versão 7.8sp2 onde foi desenvolvido o diagrama esquemático de interconectividade do Arduino com os dispositivos de entrada e saída, como pode ser visto no diagrama (Figura 14).

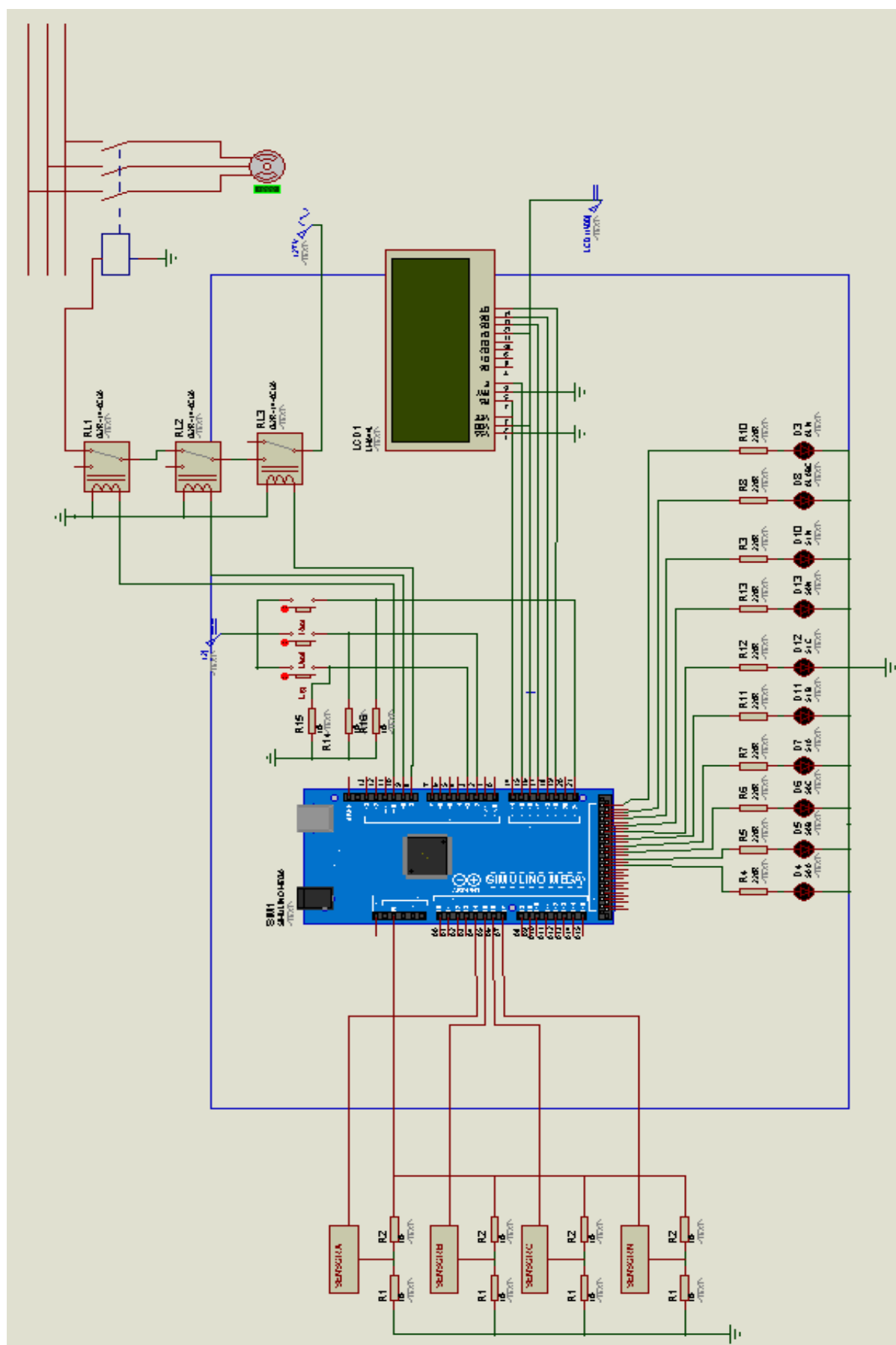


FIGURA 14. Esquemático do Projeto Montado no *Software* Proteus  
 Fonte: Própria, 2017.

Basicamente o circuito é constituído por uma placa Arduino Mega 2560, que é o responsável por toda a manipulação das variáveis e controle das entradas e saídas digitais e analógicas seguindo a lógica de funcionamento através do código desenvolvido no IDE e inserido na memória do Arduino APÊNDICE A e APÊNDICE B, com uma breve explicação das funções utilizadas. Os quatro sensores SCT 013 30 que fazem as medições de corrente das fases A, B, C e de Neutro estão ligados às entradas analógicas do Arduino seguindo o esquema (Figura 13) como foi visto anteriormente. O circuito ainda possui os três botões que são usados para os comandos de Ligar, Desligar a carga e o botão de *Reset* das condições de entrada e saída do Arduino, como por exemplo reiniciar os LED's de indicação de *trip* e deixar a carga em condição de partida. Para mostrar as medições das correntes foi conectado um *display* LCD do tamanho 20 colunas por 4 linhas, onde são exibidas além das medições de corrente, a potência correspondente a cada fase e as mensagens de *trip* quando da ocorrência de alguma falha no circuito de potência e atuação do relé. Ainda foram utilizados 9 LED's para indicação das proteções atuadas, sendo 3 para sinalização de *trip* das funções ANSI 51 das fases A, B e C, outros 3 para sinalização de *trip* pela função ANSI 50 das fases A, B e C, 1 LED para sinalização de *trip* pela função ANSI 51 de Neutro, 1 LED para sinalização de *trip* pela função ANSI 50 de Neutro, e outros dois LED's para sinalização de alarme quando a corrente de Fase e de Neutro atingir valores acima dos nominais.

Para a parte de acionamento e comando da carga foi utilizado o módulo relé *shield* (Figura 15), esse modulo tem como característica um sinal de acionamento de entrada de 5V, que é acionado por uma saída digital do Arduino, comutando assim seus contatos de saída (1 contato normal aberto e 1 normal fechado) que podem ser usados para alimentar cargas de até 250V / 10A.

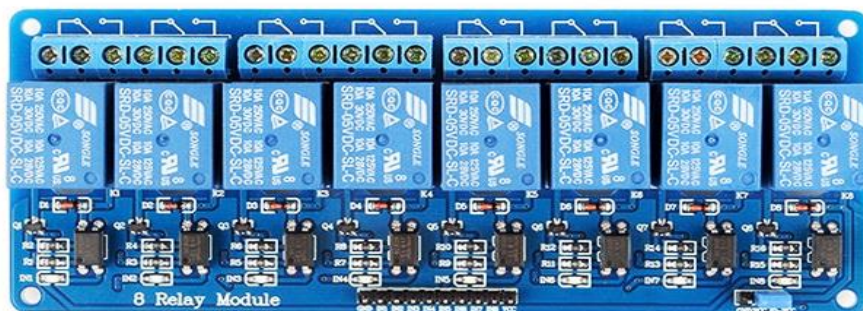


FIGURA 15. Módulo relé 5V, 8 canais

Fonte: Multilógica-Shop, 2017.<sup>9</sup>

<sup>9</sup> Modulo ReléShield - MULTILÓGICA, 2017 <<https://multilogica-shop.com/modulo-rele-5v-8-canais>>

Foram usados apenas 3 módulos dos 8 existentes, um para ligar a carga quando o botão “Liga” for pressionado, um para desligar a carga quando o botão “Desliga” for pressionado, e o terceira que é o módulo de *Trip*, que é acionado quando o relé entende que há um defeito no circuito atuando alguma das proteções de sobrecorrente ajustadas, comandando assim desligamento da carga afim de eliminar o defeito.

Após a realização das simulações de funcionalidade do dispositivo no *Software* Proteus foi desenvolvida a parte de montagem física do relé, ficando o *Lay Out* como segue (Figura 16 e Figura 17). O circuito que foi montado fisicamente segue o digrama simulado, mudando apenas a disposição de alguns componentes para facilitar conexões entre os mesmos e reduzir espaço.



FIGURA 16. Montagem do Relé – Parte de Trás  
Fonte: Própria, 2017.



FIGURA 17. Frente do Relé Usando Arduino

Fonte: Própria, 2017.

Para os ensaios de atuação do dispositivo foi ajustado os valores de alarme e de *pick up* das unidades de sobrecorrente do relé (ambos ajustados em 3A) além do tipo de curva a ser usada e o Dial de Tempo. Para o teste foi ajustada a curva IEC Muito Inversa no DT 1.0, em seguida foi utilizada uma Unidade de Ensaio de Relé *Mult Test*, do fabricante Instronic, modelo UER 100, série 122 (Figura 18) para injeção de corrente nos sensores das fases A, B, C e de Neutro, onde foi gerada uma corrente acima do valor de ajuste para que fosse possível verificar o comportamento do dispositivo.





FIGURA 18. Unidade de Ensaio de Relés

Fonte: Própria.

Para as funções de sobrecorrente temporizada ANSI 51, o dispositivo não se comportou de forma esperada, onde os tempos de atuação não seguiu os tempos correspondentes à curva ajustada. Foram feitos vários testes com diversos valores de corrente e inclusive feitas alterações na programação do Arduino para corrigir alguns problemas e tentar fazer com que o dispositivo acompanhasse a inclinação da curva ajustada, entretanto a parte de temporização da função ANSI 51 utilizando a curva de tempo inverso não foi possível corrigir todos os erros devido algumas limitações do Arduino, onde seria exigido funções de programação um pouco mais complexas e velocidade de processamento maior. Após todas as correções feitas foi obtida uma curva de atuação bem próxima da especificada, porém existe um desvio entre as duas curvas devido ao tempo de processamento e atualização das entradas e saídas do Arduino. A seguir (Tabela 02) estão os dados dos ensaios realizados onde mostra os valores obtidos e os especificados através da equação característica da curva utilizada.

TABELA 02. Tempo de Atuação – Temporização por Curva de Tempo Inverso

Tempos de Atuação Curva IEC Muito Inversa no DT 1.0			
	Múltiplos de Corrente	Tempo Especificado (s)	Tempo Obtido (s)
1	2,00	13,50	15,03
2	3,00	6,75	8,05
3	4,00	4,50	6,46
4	5,00	3,38	4,26
5	6,00	2,70	3,61
6	7,00	2,25	3,14
7	8,00	1,93	2,63

Fonte: Própria, 2017.

A partir da tabela anterior foi gerado o gráfico (GRAF. 01) que mostra a comparação entre a curva ajustada no relé e a curva que foi gerada através dos valores obtidos nos ensaios.

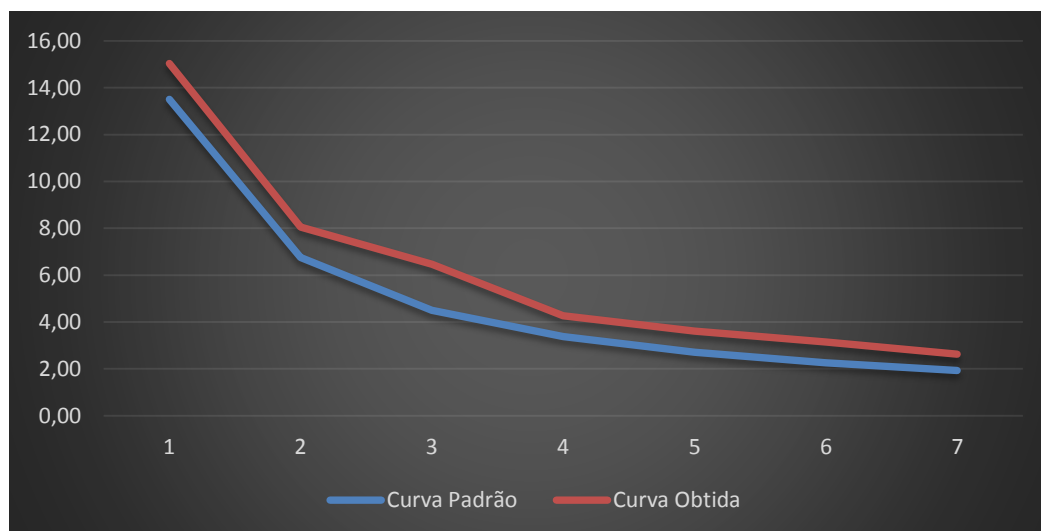


GRÁFICO 01. Curva de Atuação de Tempo Muito Inverso  
Fonte: Própria, 2017.

Através do gráfico fica fácil comparar o desempenho do relé com Arduino, onde a curva em vermelho foi gerada através dos valores obtidos, e a curva em azul foi gerada com os valores de referência da curva utilizada.

Para melhorar a funcionalidade do dispositivo foi implantada a temporização por tempo definido, que pode desempenhar uma função semelhante e bastante eficiente, principalmente quando tem apenas dispositivos de proteção microprocessados, onde pode ser feita a coordenação dos dispositivos intercalando os tempos de desarme. Foi ajustado os tempos de desarme na programação do relé e em seguida aplicada uma corrente superior ao *pick up* da função ANSI 51 para verificar se o dispositivo atuaria no tempo ajustado. A seguir (Tabela 03) estão os resultados de alguns ensaios realizados para verificar a eficácia do dispositivo, e a curva de atuação resultante (GRAF. 02).

TABELA 03: Tempo de Atuação - Temporização por Tempo Definido

	Tempo Ajustado (s)	Tempo de Atuação (s)	Erro (%)
1	10	10,40	4
2	5	5,40	8
3	2	2,09	4,5
4	1	1,04	4

Fonte: Própria



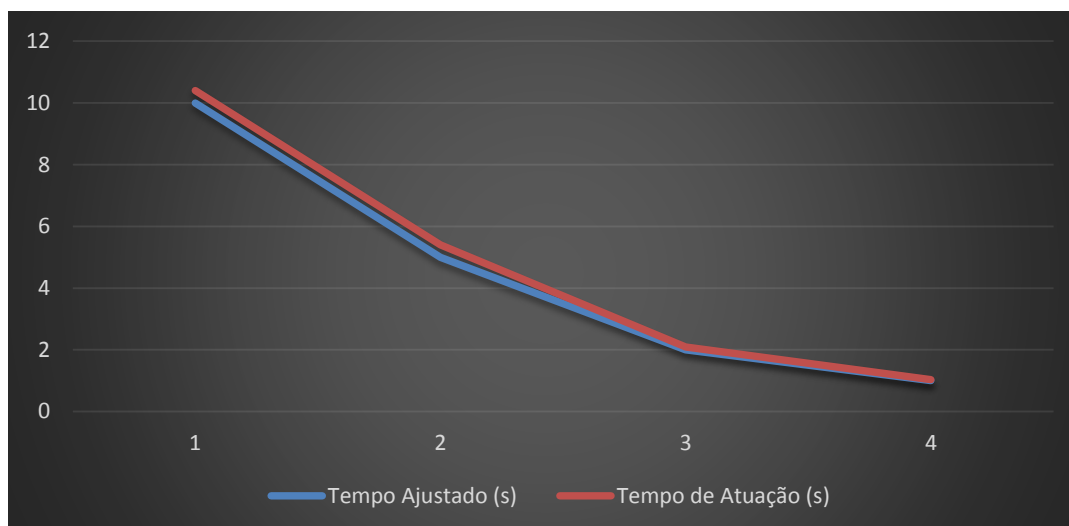


GRÁFICO 02. Curva de Atuação por Tempo Definido

Fonte: Própria, 2017.

Como pode ser observado a temporização por tempo definido se mostrou bastante eficiente tendo em vista que os tempos de atuação obtidos (curva em Vermelho) se assemelham aos tempos ajustados (curva em Azul), sendo a diferença existente devido ao tempo de processamento que o Arduino leva para manipular as variáveis e atualizar as entradas e saídas do dispositivo. É importante lembrar que dependendo da aplicação do relé, através de intercalação da temporização podemos obter tempos de atuação bem parecidos com os da curva de tempo inverso, assim podemos simular a curva através da temporização por tempo definido, tornando a coordenação mais eficiente.

Para testar as unidades de sobrecorrente instantâneas ANSI 50 de fase e de neutro, foram ajustados os valores de *pick up* das mesmas, que para o teste foram ajustadas em 10 vezes o ajuste da função ANSI 51, ou seja 30 A, em seguida gerada uma corrente ligeiramente superior ao valor ajustado. O tempo de desarme obtido foi em torno de 0,74 s ou 740 ms, que para uma unidade de proteção instantânea pode ser considerado alto, visto que determinadas cargas requerem tempos de desarme da ordem de algumas poucas dezenas de milissegundos. Tal tempo se deve ao período em que o Arduino gasta para processar os dados e comutar o relé de *Trip* para que possa desarmar a unidade de testes.

Além dos ensaios realizados com a unidade de testes foi montado um circuito prático no laboratório de elétrica do Centro de Ensino Superior de Conselheiro Lafaiete, onde foi possível simular uma sobrecarga no circuito fazendo com que a proteção atuasse, comprovando sua funcionalidade. O circuito montado (Figura 19) é composto por dois

motores iguais do fabricante EBERLE, modelo B 71 R2 de 1 CV, sendo alimentados em 220 V, através de um contator e um disjuntor. A bobina do contator está ligada ao relé onde é possível comandar o fechamento e abertura do mesmo acionando o motor. Foi ajustada no relé uma corrente de *pick up* ligeiramente acima da corrente nominal de apenas um motor ( $I_N = 1,2 \text{ A}$  e  $I_{\text{ajuste}} = 1,5 \text{ A}$ ), que permite o funcionamento normal quando apenas um motor estiver operando. Ao acionar o segundo motor através do disjuntor, a corrente do circuito dobra e ultrapassa o valor ajustado, o relé então entende que há uma falha no circuito e executa os cálculos de acordo com a equação característica da curva ajustada no relé, para saber quanto tempo deve esperar para comandar o desarmar caso a falha permaneça, após passar o tempo calculado, como a corrente ainda se mantém acima do valor ajustado ele comanda a abertura do contator interrompendo o circuito e eliminando a corrente de falha, e sinaliza através dos LED's qual a função de proteção atuou.

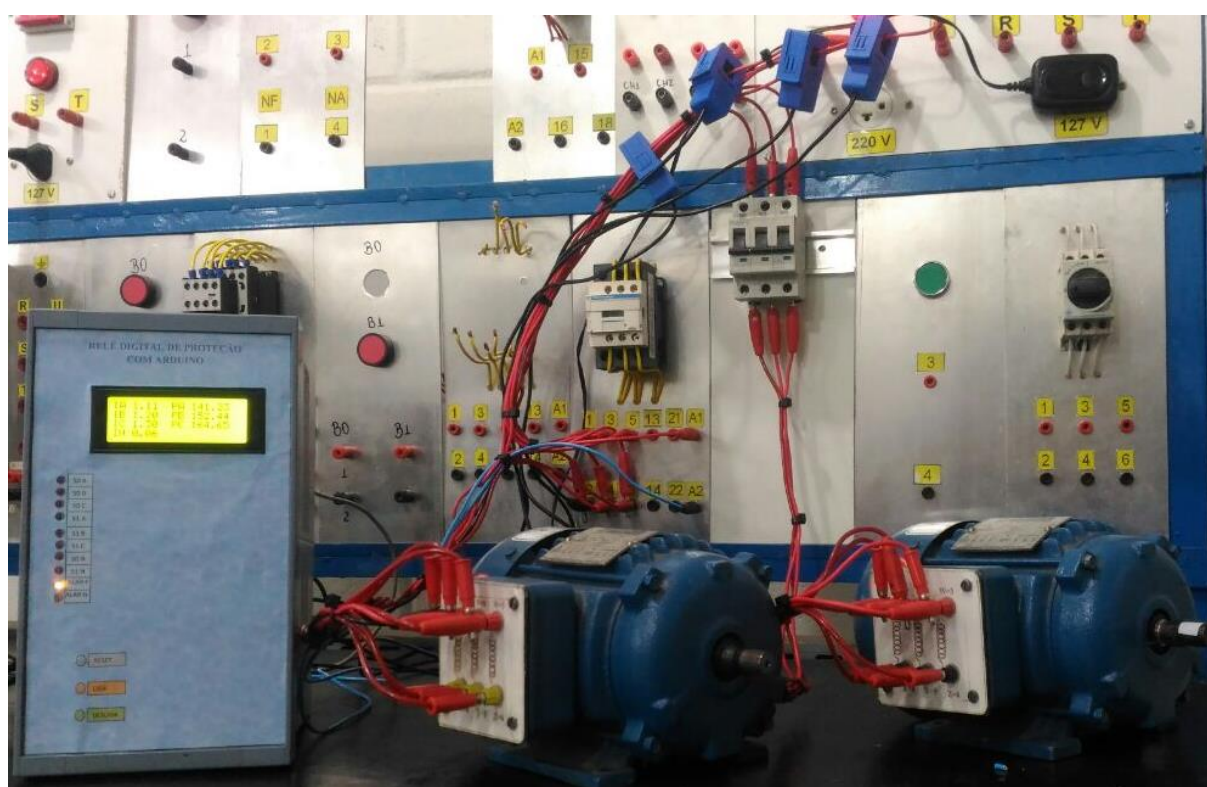


FIGURA 19: Circuito Prático Montado para Simulação de Sobrecarga e Verificação de Atuação do Relé.

Fonte: Própria.

## 5 CONSIDERAÇÕES FINAIS

Do ponto de vista prático é possível observar que o projeto desenvolvido é totalmente funcional e desempenhou as funções propostas seguindo a lógica de funcionamento inserida no microcontrolador Arduino, se mostrando inclusive aplicável na proteção de determinados circuitos e equipamentos elétricos que não necessitem de tempos de desarme muito pequeno (na casa de milissegundos) conforme teste prático realizado no laboratório. Entretanto a eficiência do dispositivo de uma forma geral se torna inviável para sistemas de proteção que exijam elevada confiabilidade e alta velocidade de resposta, uma vez que o mesmo se tornou razoavelmente lento devido as diversas variáveis manipuladas simultaneamente fazendo com que o ciclo de processamento do Arduino ficasse alto, o que interferiu diretamente nos tempos de atuação.

De uma forma geral é possível concluir que com algumas alterações na parte de programação, utilizando técnicas mais avançadas e elaboradas, além de uma maior disponibilidade de recursos, como por exemplo uma substituição do *Hardware*, trocando o Arduino por outro microcontrolador com capacidade de processamento maior, possa chegar a um dispositivo de alta eficiência e confiabilidade, podendo ser implantado em diversos circuitos e equipamentos elétricos de forma a reduzir custos com a aquisição de dispositivos de proteção, visto que os relés microprocessados existentes no mercado atual são de alto valor comercial.

## REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, Pedro Afonso de Oliveira. **Transdutores para Medida de Deslocamentos Lineares**. Escola Politécnica da Universidade de São Paulo, São Paulo, 2004.

ARDUINO PARA MONITORAR CORRENTE ELÉTRICA, disponível em: <<http://www.mcmajan.com/mcmajanwpr/blog/2015/04/02/arduino-per-il-monitoraggio-del-consumo-elettrico-parte-1-introduzione-it/>> Acesso em: 10 de julho de 2017.

CAMINHA, Amadeu.C. **Introdução à Proteção dos Sistemas Elétricos**. Edgard Blucher, São Paulo, 1977.

CARVALHO, B. A. S. **Condições Operacionais de filtros em 69KV de uma usina siderúrgica**. Monografia – Centro de Ensino Superior de Conselheiro Lafaiete, Conselheiro Lafaiete, 2015.

CASANOVA, Renata Alves. **Análise do Sistema Elétrico de uma Unidade de Produção de Petróleo**. Dissertação de Mestrado – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2008.

ENGELETRICA, Disponível em :<<http://www.engeletrica.com.br/fatordepotencia-manual-fatordepotencia.html>>. Acesso em: 21 abril 2017.

EVANS, M.; NOBLE, Joshua.; HOCHENBAUM, Jordan.**Arduino em Ação**. 1. ed., São Paulo. Novatec, 2013.

JUSTEN, Álvaro. **Curso de Arduino**, Disponível em:<<http://cursodearduino.com.br>>. Acesso em: 20 maio 2017.

KINDERMANN, Geraldo.**Proteção de Sistemas Elétricos de Potência**. Florianopolis, Santa Catarina, 1999.

LIGAÇÃO SENSOR SCT 013 30, disponível em: <<https://openenergymonitor.org/>>. Acesso em: 30 de outubro de 2017.

MAMEDE FILHO, J.; MAMEDE, D. R. **Proteção de Sistemas Elétricos de Potência**. Rio de Janeiro: LTC, 2011.

MONK, S. **Programação com Arduino: Começando com Sketches**. Rio Grande do Sul: Bookman, 2013.

MOTOR *SHIELD*, disponível em: <<https://www.arduino.cc/en/Main/ArduinoMotorShieldR3>>. Acesso em: 30 de maio de 2017.

NAZARETH, P. R. P. **Proteção de Sistemas Elétricos**. Notas de Aula I, Centro de Ensino Superior de Conselheiro Lafaiete, Conselheiro Lafaiete, 2016.

NISKIER, J.; MANCINTYRE, A.J. **Instalações Elétricas**. 5. ed. Rio de Janeiro: LTC, 2008.

POTÊNCIA ATIVA, Disponível em: <<http://www.engeletrica.com.br/fatordepotencia-manual-fatordepotencia.html>>. Acesso em: 20 março 2017.

RELÉ*SHIELD*, disponível em: <<https://multilogica-shop.com/modulo-rele-5v-8-canais>>. Acesso em: 30 de outubro de 2017.

SANTOS, Kelly Vinente dos. **Fundamentos de Eletricidade**. Centro de Educação Tecnológica do Amazonas, Amazonas, 2011.

SATO, F.; FREITAS, W. **Análise De Curto-Circuito E Princípios De Proteção Em Sistemas De Energia Elétrica**. 1.ed. Campus, 2014.

SATO, Fujio. **Noções Básicas de Proteção de Sistemas de Energia Elétrica**. Universidade Estadual de Campinas, Campinas, 2005.

TABELA ANSI, disponível em: < <https://selinc.com/pt/products/tabelas/ansi/> > Acesso em: 28 de novembro de 2017.

## ANEXO A – TABELA ANSI

<b>Nr</b>	<b>Denominação</b>
1	Elemento Principal
2	Relé de partida ou fechamento temporizado
3	Relé de verificação ou interbloqueio
4	Contator principal
5	Dispositivo de interrupção
6	Disjuntor de partida
7	Relé de taxa de variação
8	Dispositivo de desligamento da energia de controle
9	Dispositivo de reversão
10	Chave comutadora de sequência das unidades
11	Dispositivo multifunção
12	Dispositivo de sobrevelocidade
13	Dispositivo de rotação síncrona
14	Dispositivo de subvelocidade
15	Dispositivo de ajuste ou comparação de velocidade e/ou frequência
16	Dispositivo de comunicação de dados
17	Chave de derivação ou descarga
18	Dispositivo de aceleração ou desaceleração
19	Contator de transição partida-marcha
20	Válvula operada eletricamente
21	Relé de distância
22	Disjuntor equalizador
23	Dispositivo de controle de temperatura
24	Relé de sobreexcitação ou Volts por Hertz

Nr	Denominação
25	Relé de verificação de Sincronismo ou Sincronização
26	Dispositivo térmico do equipamento
27	Relé de subtensão
28	Detector de chama
29	Contator de isolamento
30	Relé anunciador
31	Dispositivo de excitação
32	Relé direcional de potência
33	Chave de posicionamento
34	Dispositivo master de sequência
35	Dispositivo para operação das escovas ou curto-circuitar anéis coletores
36	Dispositivo de polaridade ou polarização
37	Relé de subcorrente ou subpotência
38	Dispositivo de proteção de mancal
39	Monitor de condições mecânicas
40	Relé de perda de excitação ou relé de perda de campo
41	Disjuntor ou chave de campo
42	Disjuntor / chave de operação normal
43	Dispositivo de transferência ou seleção manual
44	Relé de sequência de partida
45	Monitor de condições atmosféricas
46	Relé de reversão ou desbalanceamento de corrente
47	Relé de reversão ou desbalanceamento de tensão
48	Relé de sequência incompleta / partida longa
49	Relé térmico
50	Relé de sobrecorrente instantâneo

<b>Nr</b>	<b>Denominação</b>
51	Relé de sobrecorrente temporizado
52	Disjuntor de corrente alternada
53	Relé para excitatriz ou gerador CC
54	Dispositivo de acoplamento
55	Relé de fator de potência
56	Relé de aplicação de campo
57	Dispositivo de aterramento ou curto-circuito
58	Relé de falha de retificação
59	Relé de sobretensão
60	Relé de balanço de corrente ou tensão
61	Sensor de densidade
62	Relé temporizador
63	Relé de pressão de gás (Buchholz)
64	Relé detetor de terra
65	Regulador
66	Relé de supervisão do número de partidas
67	Relé direcional de sobrecorrente
68	Relé de bloqueio por oscilação de potência
69	Dispositivo de controle permissivo
70	Reostato
71	Dispositivo de detecção de nível
72	Disjuntor de corrente contínua
73	Contator de resistência de carga
74	Relé de alarme
75	Mecanismo de mudança de posição
76	Relé de sobrecorrente CC



<b>Nr</b>	<b>Denominação</b>
77	Dispositivo de telemedição
78	Relé de medição de ângulo de fase / proteção contra falta de sincronismo
79	Relé de religamento
80	Chave de fluxo
81	Relé de frequência (sub ou sobre)
82	Relé de religamento de carga de CC
83	Relé de seleção / transferência automática
84	Mecanismo de operação
85	Relé receptor de sinal de telecomunicação (teleproteção)
86	Relé auxiliar de bloqueio
87	Relé de proteção diferencial
88	Motor auxiliar ou motor gerador
89	Chave seccionadora
90	Dispositivo de regulação (regulador de tensão)
91	Relé direcional de tensão
92	Relé direcional de tensão e potência
93	Contator de variação de campo
94	Relé de desligamento
95	Usado para aplicações específicas
96	Relé auxiliar de bloqueio de barra
97 à 99	Usado para aplicações específicas
150	Indicador de falta à terra
AFD	Detector de arco voltaico
CLK	Clock
DDR	Sistema dinâmico de armazenamento de perturbações
DFR	Sistema de armazenamento de faltas digital

<b>Nr</b>	<b>Denominação</b>
ENV	Dados do ambiente
HIZ	Detector de faltas com alta impedância
HMI	Interface Homem-Máquina
HST	Histórico
LGC	Esquema lógico
MET	Medição de Subestação
PDC	Concentrador de dados de fasores
PMU	Unidade de medição de fasores
PQM	Esquema de monitoramento de potência
RIO	Dispositivo Remoto de Inputs/Outputs
RTU	Unidade de terminal remoto / Concentrador de Dados
SER	Sistema de armazenamento de eventos
TCM	Esquema de monitoramento de Trip
SOTF	Fechamento sob falta

Fonte: SEL (Schweitzer Engineering Laboratories), 2017

### **Complementação da Tabela ANSI:**

50N - sobrecorrente instantâneo de neutro

51N - sobrecorrente temporizado de neutro ( tempo definido ou curvas inversas)

50G - sobrecorrente instantâneo de terra (comumente chamado 50GS)

51G - sobrecorrente temporizado de terra (comumente chamado 51GS e com tempo definido ou curvas inversas)

50BF - relé de proteção contra falha de disjuntor (também chamado de 50/62 BF)

51Q - relé de sobrecorrente temporizado de sequência negativa com tempo definido ou curvas inversas

51V - relé de sobrecorrente com restrição de tensão

51C - relé de sobrecorrente com controle de torque

50PAF - sobrecorrente de fase instantânea de alta velocidade para detecção de arco voltaico

50NAF - sobrecorrente de neutro instantânea de alta velocidade para detecção de arco voltaico

59Q - relé de sobretensão de sequência negativa

59N - relé de sobretensão residual ou sobretensão de neutro (também chamado de 64G) ,

calculado ou TP em delta aberto

64 - relé de proteção de terra pode ser por corrente ou por tensão. Os diagramas unifilares devem indicar se este elemento é alimentado por TC ou por TP, para que se possa definir corretamente. Se for alimentado por TC, também pode ser utilizado como uma unidade 51 ou 61. Se for alimentado por TP, pode-se utilizar uma unidade 59N ou 64G. A função 64 também pode ser encontrada como proteção de carcaça, massa-cuba ou tanque, sendo aplicada em transformadores de força até 5 MVA.

67N - relé de sobrecorrente direcional de neutro (instantâneo ou temporizado)

67G - relé de sobrecorrente direcional de terra (instantâneo ou temporizado)

67Q - relé de sobrecorrente direcional de sequência negativa

78 - Salto vetorial (Vector Shift)

### **Proteção Diferencial - ANSI 87:**

O relé diferencial 87 pode ser de diversas maneiras:

87T - diferencial de transformador (pode ter 2 ou 3 enrolamentos)

87N - diferencial de neutro

REF – falta restrita à terra

87Q - diferencial de sequência negativa (aplicado para detecção de faltas entre espiras em transformadores)

87G - diferencial de geradores

87GT - proteção diferencial do grupo gerador-transformador

87SP - proteção diferencial de fase dividida de geradores

87V – Diferencial de tensão de fase

87VN – Diferencial de tensão de neutro

87B - diferencial de barras. Pode ser de alta, média ou baixa impedância

Pode-se encontrar em circuitos industriais elementos de sobrecorrente ligados num esquema diferencial, onde os TC's de fases são somados e ligados ao relé de sobrecorrente.

Também encontra-se um esquema de seletividade lógica para realizar a função diferencial de barras.

Pode-se encontrar em algumas documentações o relé 68 sendo referido à função de seletividade lógica.

87M - diferencial de motores - Neste caso pode ser do tipo percentual ou do tipo autobalanceado.

O percentual utiliza um circuito diferencial através de 3 TC's de fases e 3 TC's no neutro do motor. O tipo autobalanceado utiliza um jogo de 3 TC's nos terminais do motor, conectados de forma a obter a somatória das correntes de cada fase e neutro. Na realidade, trata-se de um elemento de sobrecorrente, onde o esquema é diferencial e não o relé.

## APÊNDICE A - CÓDIGO UTILIZADO NO ARDUINO PARA TEMPORIZAÇÃO COM CURVA DE TEMPO INVERSO

```
#include "EmonLib.h" //inlui as bibliotecas necessárias
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F,2,1,0,4,5,6,7,3, POSITIVE); // Inicializa o display no endereço 0x3F

int rede = 127.0; //Valor da tensão em cada fase

EnergyMonitor emon1; //Variáveis a serem manipuladas referente a fase A,B,C e neutro N
EnergyMonitor emon2;
EnergyMonitor emon3;
EnergyMonitor emon4;

int IA = 0;
int IB = 0;
int IC = 0;
int IN = 0;

int Irms1 = 0;
int Irms2 = 0;
int Irms3 = 0;
int Irms4 = 0;

const int Alar = 1.5; //Valor ajustado para alarme de sobrecorrente de fase
const int AlarN = 1.5; //Valor ajustado para alarme de sobrecorrente de neutro
const int ICC = 6.0; //Valor de ajuste de corrente de pick up unidade instantânea de fase
const int ICCN = 3.0; //Valor de ajuste de corrente de pick up unidade instantânea de neutro
const int IPK = 1.8; //Valor de ajuste de corrente de pick up unidade temporizada de fase
const int IPKN = 1.0; //Valor de ajuste de corrente de pick up unidade temporizada de neutro
const int Dt = 1.0; //Dial de Tempo selecionado para a curva
const int Beta = 13.5; //Constante de acordo com o tipo de curva selecionado (Muito Inversa)
const int Alfa = 1.0; //Constante de acordo com o tipo de curva selecionado (Muito Inversa)

int pino_sct = A7; //pino sensor de corrente fase A
int pino_sct2 = A6; //pino sensor de corrente fase B
```

```

int pino_sct3 = A5;          //pino sensor de corrente fase C
int pino_sct4 = A4;          //pino sensor de corrente neutro N

int LED50APin = 40;          //pino led de indicação de trip unidade instantânea fase A
int LED50BPin = 38;          //pino led de indicação de trip unidade instantânea fase B
int LED50CPin = 36;          //pino led de indicação de trip unidade instantânea fase C
int LED51A = 34;             //pino led de indicação de trip unidade temporizada fase A
int LED51B = 32;             //pino led de indicação de trip unidade temporizada fase B
int LED51C = 30;             //pino led de indicação de trip unidade temporizada fase C
int LED50NPin = 28;          //pino led de indicação de trip unidade instantânea neutro N
int LED51N = 26;             //pino led de indicação de trip unidade temporizada neutro N
int LEDALPin = 24;           //pino de alarme sobrecorrente de fase
int LEDALNPin = 22;          //pino de alarme sobrecorrente de neutro

int releTripPin = 5;         //pino do relé que da trip na carga
int releLigPin = 7;          //pino do relé que liga a carga
int releDesPin = 6;          //pino do relé que desliga a carga

int Desl = 2;                //pino do botão que desliga a carga
int Lig = 3;                 //pino do botão que liga a carga
int Reset = 18;              //pino do botão que reseta as condições das entradas e saídas do Arduino

volatile int STATE1=LOW;     //indica estado inicial dos botões em nível baixo
volatile int STATE2=LOW;
volatile int STATE3=LOW;

long previousMillis1 = 0;    //constantes usadas para temporização da fase A
long IntervalA = 0;
long currentMillis1 = 0;

long previousMillis2 = 0;    //constantes usadas para temporização da fase B
long IntervalB = 0;
long currentMillis2 = 0;

long previousMillis3 = 0;    //constantes usadas para temporização da fase c
long IntervalC = 0;
long currentMillis3 = 0;

long previousMillis4 = 0;    //constantes usadas para temporização de neutro N
long IntervalN = 0;

```

```

long currentMillis4 = 0;
void setup()
{
  lcd.begin (20,4);           //define LCD como sendo do tamanho 20 colunas por 4 linhas

  pinMode (pino_sct,INPUT);    //define pino dos sensores como entradas
  pinMode (pino_sct2,INPUT);
  pinMode (pino_sct3,INPUT);
  pinMode (pino_sct4,INPUT);

  pinMode (LED50APin,OUTPUT);  //define pino dos LED's como saída
  pinMode (LED50BPin,OUTPUT);
  pinMode (LED50CPin,OUTPUT);
  pinMode (LED50NPin,OUTPUT);
  pinMode (LEDALPin,OUTPUT);
  pinMode (LEDALNPin,OUTPUT);
  pinMode (LED51A,OUTPUT);
  pinMode (LED51B,OUTPUT);
  pinMode (LED51C,OUTPUT);
  pinMode (LED51N,OUTPUT);

  pinMode (releTripPin,OUTPUT); //define pino dos relés como saída
  pinMode (releLigPin,OUTPUT);
  pinMode (releDesPin,OUTPUT);

  pinMode (Desl, INPUT);       //define os botões como entradas
  pinMode (Lig, INPUT);
  pinMode (Reset, INPUT);

  //Pino, calibracao - Cur Const= Ratio/BurdenR. 1800/62 = 29

  emon1.current(pino_sct, 29); //calibra os pinos dos sensores de corrente
  emon2.current(pino_sct2, 29);
  emon3.current(pino_sct3, 29);
  emon4.current(pino_sct4, 29);
}

```

```

void loop()

{
    //Calcula as correntes
    double Irms1 = emon1.calcIrms(1480);           //calcula corrente da fase A
    double Irms2 = emon2.calcIrms(1480);           //calcula corrente da fase B
    double Irms3 = emon3.calcIrms(1480);           //calcula corrente da fase C
    double Irms4 = emon4.calcIrms(1480);           //calcula corrente de N

    IA=Irms1;           //atribui os valores das correntes calculadas para as variáveis criadas para cada fase
    IB=Irms2;
    IC=Irms3;
    IN=Irms4;

    //Mostra o valor da corrente
    lcd.clear ();           //limpa o LCD

    lcd.setCursor (0,0);           //seta o cursor na coluna 0 e linha 0
    lcd.print ("IA");           //escreve IA(corrente fase A)
    lcd.setCursor (3,0);           //seta cursor coluna 3 e linha 0
    lcd.print (Irms1);           //escreve a corrente referente a fase A

    lcd.setCursor (0,1);
    lcd.print ("IB");
    lcd.setCursor (3,1);
    lcd.print (Irms2);

    lcd.setCursor (0,2);
    lcd.print ("IC");
    lcd.setCursor (3,2);
    lcd.print (Irms3);

    lcd.setCursor (0,3);
    lcd.print ("IN");
    lcd.setCursor (3,3);
    lcd.print (Irms4);

    //Calcula e mostra o valor da potência
    lcd.setCursor (9,0);           //seta o cursor na coluna 9 e linha 0
    lcd.print ("PA");           //escreve PA(potência fase A)

```

```

lcd.setCursor (12,0);           //seta o cursor na coluna 12 e linha 0
lcd.print (Irms1*rede);         //calcula a potência e escreve o resultado

lcd.setCursor (9,1);
lcd.print ("PB");
lcd.setCursor (12,1);
lcd.print (Irms2*rede);

lcd.setCursor (9,2);
lcd.print ("PC");
lcd.setCursor (12,2);
lcd.print (Irms3*rede);

//função para acender LED de alarme de sobrecorrente de fase

if (Irms1 > Alar || Irms2 > Alar || Irms3 > Alar)           //Se corrente da fase A,B ou C for maior que o valor
de ajuste de alarme
{
    digitalWrite (LEDALPin, HIGH);                         // Acende o LED de alarme
}
else                                                         // Se não for maior mantem LED de alarme apagado
{
    digitalWrite(LEDALPin,LOW);                             //mantem LED apagado
}

//função de sobrecorrente temporizada da fase A

if (Irms1 >= IPK)                                             //Se corrente da fase A for maior que o valor de ajuste sobrecorrente de fase
{
    IntervalA = (((Dt*Beta) / ((IA / IPK)^Alfa - 1)) *1000); //calcula tempo de atuação através da
equação da curva característica
    if (previousMillis1==0)                                  //se tempo anterior igual a zero
    {
        previousMillis1= millis();                          //atribui tempo atual
    }
    else                                                      //senão
    {
        unsigned long currentMillis1 = millis();            //atribui tempo atual
    }
}

```



```

    if(currentMillis1-previousMillis1 >=IntervalA) //se tempo atual menos tempo anterior for maior ou
    igual ao intervalo de desarme

```

```

    {
        previousMillis1 = 0; //atribui zero ao tempo anterior
        digitalWrite (releTripPin, HIGH); //aciona o pino que atuará o relé desarmando - o
        digitalWrite (LED51A, HIGH); //aciona led correspondente a função que deu trip (51 A)
        lcd.clear ();
        lcd.setCursor (0,0);
        lcd.print (" Sobrecorrente A"); //escreve mensagem no lcd falando em qual fase foi o defeito
    }
}

```

//função de sobrecorrente temporizada da fase B

```

if (Irms2 >= IPK)
{
    IntervalB = (((Dt*Beta) / ((IB / IPK)^Alfa - 1)) *1000);
    if (previousMillis2==0)
    {
        previousMillis2= millis();
    }
    else
    {
        unsigned long currentMillis2 = millis();
        if(currentMillis2-previousMillis2 >=IntervalB)
        {
            previousMillis2 = 0;
            digitalWrite (releTripPin, HIGH
            digitalWrite (LED51B, HIGH
            lcd.clear ();
            lcd.setCursor (0,0);
            lcd.print (" Sobrecorrente B");
        }
    }
}

```

//função de sobrecorrente temporizada da fase C

```

if (Irms3 >= IPK)
{

```

```

IntervalC = (((Dt*Beta) / ((IC / IPK)^Alfa - 1)) *1000);
if (previousMillis3==0)
{
    previousMillis3= millis();
}
else
{
    unsigned long currentMillis3 = millis();
    if(currentMillis3-previousMillis3 >=IntervalC)
    {
        previousMillis3 = 0;
        digitalWrite (releTripPin, HIGH);
        digitalWrite (LED51C, HIGH);
        lcd.clear ();
        lcd.setCursor (0,0);
        lcd.print (" Sobrecorrente C");
    }
}
}

```

//função para controlar LED de alarme de sobrecorrente de neutro

```

if (Irms4 > AlarN)           //Se corrente de neutro for maior que o valor de ajuste de alarme de neutro
{
    digitalWrite (LEDALNPin, HIGH);           // Acende o LED de alarme de neutro
}
else                           //se não for
{
    digitalWrite(LEDALNPin,LOW);           //mantem LED apagado
}

```

//função de sobrecorrente temporizada de neutro N

```

if (Irms4 >= IPKN)
{
    IntervalN = (((Dt*Beta) / ((IN / IPKN)^Alfa - 1)) *1000);
    if (previousMillis4==0)
    {
        previousMillis4= millis();
    }
}

```

```

else
{
    unsigned long currentMillis4 = millis();
    if(currentMillis4-previousMillis4 >=IntervalN)
    {
        previousMillis4 = 0;
        digitalWrite (releTripPin, HIGH);
        digitalWrite (LED51N, HIGH);
        lcd.clear ();
        lcd.setCursor (0,0);
        lcd.print (" Sobrecorrente N");
    }
}
}

```

//Sobrecorrente fase A Instantânea

```

if (Irms1 > ICC) //Se corrente da fase A for maior que o valor de ajuste de
pick up da unidade instantânea de fase

```

```

{
    digitalWrite (releTripPin, HIGH); //aciona o pino que atuará o relé desarmando - o
    digitalWrite (LED50APin, HIGH); //aciona led correspondente a função que deu trip (50 A)
    lcd.clear ();
    lcd.setCursor (0,0);
    lcd.print (" Sobrecorrente A"); //escreve mensagem no lcd falando em qual fase foi o defeito
}

```

//Sobrecorrente fase B Instantânea

```

if (Irms2 > ICC)
{
    digitalWrite (releTripPin, HIGH);
    digitalWrite (LED50BPin, HIGH);
    lcd.clear ();
    lcd.setCursor (0,1);
    lcd.print (" Sobrecorrente B");
}

```

//Sobrecorrente fase C Instantânea

```

if (Irms3 > ICC)
{
    digitalWrite (releTripPin, HIGH);
    digitalWrite (LED50CPin, HIGH);
    lcd.clear ();
    lcd.setCursor (0,2);
    lcd.print (" Sobrecorrente C");
}

```

//Sobrecorrente Neutro Instantânea

```

if (Irms4 > ICCN)
{
    digitalWrite (releTripPin, HIGH);
    digitalWrite (LED50NPIn, HIGH);
    lcd.clear ();
    lcd.setCursor (0,3);
    lcd.print (" Sobrecorrente N ");
}

```

```

STATE1 = digitalRead ( Desl);           //lê o estado do botão desliga
STATE2 = digitalRead ( Lig);           //lê o estado do botão liga
STATE3 = digitalRead (Reset);          //lê o estado do botão reset

```

```

if (STATE1==HIGH)                       //se o botão desliga estiver em nível logico alto
{
    digitalWrite(releDesPin,HIGH);      //aciona o rele que desliga a carga
}

```

```

if (STATE2==HIGH)                       //se o botão liga estiver em nível logico alto
{
    digitalWrite(releLigPin,HIGH);      //aciona o rele que liga a carga
}

```

```

if (STATE3==HIGH)                       //se o botão reset estiver em nível logico alto
{
    lcd.clear ();                       //limpa LCD
}

```

```

digitalWrite(releTripPin,LOW);           //muda todas as saídas de LED's e reles para nível logico baixo
digitalWrite(releLigPin,LOW);
digitalWrite(releDesPin,LOW);
digitalWrite(LED50APin,LOW);
digitalWrite(LED50BPin,LOW);
digitalWrite(LED50CPin,LOW);
digitalWrite(LEDALPin,LOW);
digitalWrite(LEDALNPin,LOW);
digitalWrite(LED50NPin,LOW);
digitalWrite(LED51A,LOW);
digitalWrite(LED51B,LOW);
digitalWrite(LED51C,LOW);
digitalWrite(LED51N,LOW);
previousMillis1 = 0;                     //seta as variáveis de controle com valor zero
currentMillis1 = 0;
previousMillis2 = 0;
currentMillis2 = 0;
previousMillis3 = 0;
currentMillis3 = 0;
previousMillis4 = 0;
currentMillis4 = 0;
IntervalA = 0;
IntervalB = 0;
IntervalC = 0;
IntervalN = 0;
}
}

```

## APÊNDICE B - CÓDIGO UTILIZADO NO ARDUINO PARA TEMPORIZAÇÃO COM TEMPO DEFINIDO

```
#include "EmonLib.h" //Inclui as Bibliotecas Necessárias
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Crescer.h>

LiquidCrystal_I2C lcd(0x3F,2,1,0,4,5,6,7,3, POSITIVE); // Inicializa o display no endereço 0x3F

int rede = 127.0; //Valor da tensão em cada fase

EnergyMonitor emon1; //Variáveis a serem manipuladas referente a fase A,B,C e neutro N
EnergyMonitor emon2;
EnergyMonitor emon3;
EnergyMonitor emon4;

int IA = 0;
int IB = 0;
int IC = 0;
int IN = 0;

int Irms1 = 0;
int Irms2 = 0;
int Irms3 = 0;
int Irms4 = 0;

const int Alar = 1.5; //Valor ajustado para alarme de sobrecorrente de fase
const int AlarN = 1.5; //Valor ajustado para alarme de sobrecorrente de neutro
const int ICC = 7.0; //Valor de ajuste de corrente de pick up unidade instantânea de fase
const int ICCN = 5.0; //Valor de ajuste de corrente de pick up unidade instantânea de neutro
const int IPK = 3.0; //Valor de ajuste de corrente de pick up unidade temporizada de fase
const int IPKN = 2.0; //Valor de ajuste de corrente de pick up unidade temporizada de neutro

int pino_sct = A7; //pino sensor de corrente fase A
int pino_sct2 = A6; //pino sensor de corrente fase B
int pino_sct3 = A5; //pino sensor de corrente fase C
int pino_sct4 = A4; //pino sensor de corrente neutro N
```

```

int LED50APin = 40;      //pino led de indicação de trip unidade instantânea fase A
int LED50BPin = 38;      //pino led de indicação de trip unidade instantânea fase B
int LED50CPin = 36;      //pino led de indicação de trip unidade instantânea fase C
int LED51A = 34;         //pino led de indicação de trip unidade temporizada fase A
int LED51B = 32;         //pino led de indicação de trip unidade temporizada fase B
int LED51C = 30;         //pino led de indicação de trip unidade temporizada fase C
int LED50NPin = 28;      //pino led de indicação de trip unidade instantânea neutro N
int LED51N = 26;         //pino led de indicação de trip unidade temporizada neutro N
int LEDALPin = 24;       //pino de alarme sobrecorrente de fase
int LEDALNPin = 22;      //pino de alarme sobrecorrente de neutro

int releTripPin = 5;      //pino do relé que da trip na carga
int releLigPin = 7;       //pino do relé que liga a carga
int releDesPin = 6;       //pino do relé que desliga a carga

int Desl = 2;            //pino do botão que desliga a carga
int Lig = 3;            //pino do botão que liga a carga
int Reset = 18;          //pino do botão que reseta as condições das entradas e saídas do arduino

volatile int STATE1=LOW;  //indica estado inicial dos botões em nível baixo
volatile int STATE2=LOW;
volatile int STATE3=LOW;

Tempora tempA;           //variáveis usadas para temporização
Tempora tempB;
Tempora tempC;
Tempora tempN;

void setup()
{
  lcd.begin (20,4);      //define LCD como sendo do tamanho 20 colunas por 4 linhas

  pinMode (pino_sct,INPUT); //define pino dos sensores como entradas
  pinMode (pino_sct2,INPUT);
  pinMode (pino_sct3,INPUT);
  pinMode (pino_sct4,INPUT);

  pinMode (LED50APin,OUTPUT); //define pino dos LED's como saída
  pinMode (LED50BPin,OUTPUT);

```

```

pinMode (LED50CPin,OUTPUT);
pinMode (LED50NPin,OUTPUT);
pinMode (LEDALPin,OUTPUT);
pinMode (LEDALNPin,OUTPUT);
pinMode (LED51A,OUTPUT);
pinMode (LED51B,OUTPUT);
pinMode (LED51C,OUTPUT);
pinMode (LED51N,OUTPUT);

pinMode (releTripPin,OUTPUT);           //define pino dos relés como saída
pinMode (releLigPin,OUTPUT);
pinMode (releDesPin,OUTPUT);

pinMode (Desl, INPUT);                   //define os botões como entradas
pinMode (Lig, INPUT);
pinMode (Reset, INPUT);

tempA.defiSP(5000);                      //define o tempo de desarme em 5 segundos
tempB.defiSP(5000);
tempC.defiSP(5000);
tempN.defiSP(5000);

//Pino, calibração - Cur Const= Ratio/BurdenR.  $1800/62 = 29$ .
emon1.current(pino_sct, 29);    //calibra os pinos dos sensores de corrente
emon2.current(pino_sct2, 29);
emon3.current(pino_sct3, 29);
emon4.current(pino_sct4, 29);
}

void loop()
{
  double Irms1 = emon1.calcIrms(1480);    //calcula corrente da fase A
  double Irms2 = emon2.calcIrms(1480);    //calcula corrente da fase B
  double Irms3 = emon3.calcIrms(1480);    //calcula corrente da fase C
  double Irms4 = emon4.calcIrms(1480);    //calcula corrente de N

  IA=Irms1;    //atribui os valores das correntes calculadas para as variáveis criadas para cada fase
  IB=Irms2;
  IC=Irms3;
  IN=Irms4;

```



```

//Mostra o valor da corrente
lcd.clear ();           //limpa o LCD
lcd.setCursor (0,0);    //seta o cursor na coluna 0 e linha 0
lcd.print ("IA");       //escreve IA(corrente fase A)
lcd.setCursor (3,0);    //seta cursor coluna 3 e linha 0
lcd.print (Irms1);      //escreve a corrente referente a fase A

lcd.setCursor (0,1);
lcd.print ("IB");
lcd.setCursor (3,1);
lcd.print (Irms2);

lcd.setCursor (0,2);
lcd.print ("IC");
lcd.setCursor (3,2);
lcd.print (Irms3);

lcd.setCursor (0,3);
lcd.print ("IN");
lcd.setCursor (3,3);
lcd.print (Irms4);

//Calcula e mostra o valor da potencia
lcd.setCursor (9,0);    //seta o cursor na coluna 9 e linha 0
lcd.print ("PA");       //escreve PA(potência fase A)
lcd.setCursor (12,0);   //seta o cursor na coluna 12 e linha 0
lcd.print (Irms1*rede); //calcula a potência e escreve o resultado

lcd.setCursor (9,1);
lcd.print ("PB");
lcd.setCursor (12,1);
lcd.print (Irms2*rede);

lcd.setCursor (9,2);
lcd.print ("PC");
lcd.setCursor (12,2);
lcd.print (Irms3*rede);

```

//função para acender LED de alarme de sobrecorrente de fase

```
if (Irms1 > Alar || Irms2 > Alar || Irms3 > Alar)      //Se corrente da fase A,B ou C for maior que o valor de
ajuste de alarme
```

```
{
    digitalWrite (LEDALPin, HIGH);      // Acende o LED de alarme
}
else                                    // Se não for maior mantém LED de alarme apagado
{
    digitalWrite(LEDALPin,LOW);        //mantem LED apagado
}
```

//função de sobrecorrente temporizada da fase A

```
if (Irms1 >= IPK)                            //Se corrente da fase A for maior que o valor de ajuste sobrecorrente de fase
{
    if(tempA.Saida(1))                        //conta o tempo especificado
    {
        digitalWrite (releTripPin, HIGH);    //aciona o pino que atuará o relé desarmando - o
        digitalWrite (LED51A, HIGH);          //aciona led correspondente a função que deu trip (51 A)
        lcd.clear ();                          //limpa LCD
        lcd.setCursor (0,0);                  //seta cursor na coluna 0 e linha 0
        lcd.print (" Sobrecorrente A");        //escreve mensagem no lcd falando em qual fase foi o defeito (A)
        tempA.Saida(0);                      //zera a função de temporização
    }
}
```

//função de sobrecorrente temporizada da fase B

```
if (Irms2 >= IPK)                            //Se corrente da fase A for maior que o valor de ajuste sobrecorrente de fase
{
    if(tempB.Saida(1))                        //conta o tempo especificado
    {
        digitalWrite (releTripPin, HIGH);    //aciona o pino que atuará o relé desarmando - o
        digitalWrite (LED51B, HIGH);          //aciona led correspondente a função que deu trip (51 B)
        lcd.clear ();                          //limpa LCD
        lcd.setCursor (0,0);                  //seta cursor na coluna 0 e linha 0
        lcd.print (" Sobrecorrente B");        //escreve mensagem no lcd falando em qual fase foi o defeito (B)
        tempB.Saida(0);                      //zera a função de temporização
    }
}
```

//função de sobrecorrente temporizada da fase C

```

if (Irms3 >= IPK)           //Se corrente da fase A for maior que o valor de ajuste sobrecorrente de fase
{
    if(tempC.Saida(1))      //conta o tempo especificado
    {
        digitalWrite (releTripPin, HIGH);    //aciona o pino que atuará o relé desarmando - o
        digitalWrite (LED51C, HIGH);          //aciona led correspondente a função que deu trip (51 C)
        lcd.clear ();
        lcd.setCursor (0,0);
        lcd.print (" Sobrecorrente C");        //escreve mensagem no lcd falando em qual fase foi o defeito
        tempC.Saida(0);                        //zera a função de temporização
    }
}

```

//função para controlar LED de alarme de sobrecorrente de neutro

```

if (Irms4 > AlarN)          //Se corrente de neutro for maior que o valor de ajuste de alarme
{
    digitalWrite (LEDALNPin, HIGH); // Acende o LED de alarme de neutro
}
else                          //se não for
{
    digitalWrite(LEDALNPin,LOW);    //mantem LED apagado
}

```

//função de sobrecorrente temporizada de neutro N

```

if (Irms4 >= IPKN)          //Se corrente da fase A for maior que o valor de ajuste CC de fase
{
    if(tempN.Saida(1))      //conta o tempo especificado
    {
        digitalWrite (releTripPin, HIGH);    //aciona o pino que atuará o relé desarmando - o
        digitalWrite (LED51N, HIGH);          //aciona led correspondente a função que deu trip (50 A)
        lcd.clear ();
        lcd.setCursor (0,0);
        lcd.print (" Sobrecorrente N");        //escreve mensagem no lcd falando em qual fase foi o defeito
        tempN.Saida(0);                        //zera a função de temporização
    }
}

```

//Sobrecorrente fase A Instantânea

```

if (Irms1 > ICC)                                //Se corrente da fase A for maior que o valor de ajuste CC de fase
{
    digitalWrite (releTripPin, HIGH);           //aciona o pino que atuará o relé desarmando - o
    digitalWrite (LED50APin, HIGH);             //aciona led correspondente a função que deu trip (50 A)
    lcd.clear ();
    lcd.setCursor (0,0);
    lcd.print (" Sobrecorrente A");              //escreve mensagem no lcd falando em qual fase foi o defeito
}

```

//Sobrecorrente fase B Instantânea

```

if (Irms2 > ICC)                                //Se corrente da fase B for maior que o valor de ajuste CC de fase
{
    digitalWrite (releTripPin, HIGH);           //aciona o pino que atuará o relé desarmando - o
    digitalWrite (LED50BPin, HIGH);             //aciona led correspondente a função que deu trip (50 B)
    lcd.clear ();
    lcd.setCursor (0,1);
    lcd.print (" Sobrecorrente B");              //escreve mensagem no lcd falando em qual fase foi o defeito
}

```

//Sobrecorrente fase C Instantânea

```

if (Irms3 > ICC)                                //Se corrente da fase C for maior que o valor de ajuste CC de fase
{
    digitalWrite (releTripPin, HIGH);           //aciona o pino que atuará o relé desarmando - o
    digitalWrite (LED50CPin, HIGH);             //aciona led correspondente a função que deu trip (50 C)
    lcd.clear ();
    lcd.setCursor (0,2);
    lcd.print (" Sobrecorrente C");              //escreve mensagem no lcd falando em qual fase foi o defeito
}

```

//Sobrecorrente Neutro Instantânea

```

if (Irms4 > ICCN)                               //Se corrente de neutro for maior que o valor de ajuste de CC de neutro
{
    digitalWrite (releTripPin, HIGH);           //aciona o pino que atuará o relé desarmando - o
    digitalWrite (LED50NPin, HIGH);             //aciona led correspondente a função que deu trip (50 N)
}

```

```

        lcd.clear ();
        lcd.setCursor (0,3);
        lcd.print (" Sobrecorrente N ");           //escreve mensagem no lcd falando que defeito foi no neutro
    }

STATE1 = digitalRead ( Desl);                     //lê o estado do botão desliga
STATE2 = digitalRead ( Lig);                      //lê o estado do botão liga
STATE3 = digitalRead (Reset);                    //lê o estado do botão reset

if (STATE1==HIGH)                                //se o botão desliga estiver em nível logico alto
{
    digitalWrite(releDesPin,HIGH);                //aciona o rele que desliga a carga
}

if (STATE2==HIGH)                                //se o botão liga estiver em nível logico alto
{
    digitalWrite(releLigPin,HIGH);                //aciona o rele que liga a carga
}

if (STATE3==HIGH)                                //se o botão reset estiver em nível logico alto
{
    lcd.clear ();                                //limpa LCD
                                                //muda todas as saídas de LED's e reles para nível logico baixo

    digitalWrite(releTripPin,LOW);
    digitalWrite(releLigPin,LOW);
    digitalWrite(releDesPin,LOW);
    digitalWrite(LED50APin,LOW);
    digitalWrite(LED50BPin,LOW);
    digitalWrite(LED50CPin,LOW);
    digitalWrite(LEDALPin,LOW);
    digitalWrite(LEDALNPin,LOW);
    digitalWrite(LED50NPin,LOW);
    digitalWrite(LED51A,LOW);
    digitalWrite(LED51B,LOW);
    digitalWrite(LED51C,LOW);
    digitalWrite(LED51N,LOW);

}

}

```