

**UNIVERSIDADE DO ESTADO DO AMAZONAS**  
**ESCOLA SUPERIOR DE TECNOLOGIA**  
**GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**EDUARDO LIMA MANGUEIRA**

**SISTEMA REMOTO DE ENSAIOS DE ROTINA EM TRANSFORMADORES EM  
SUBESTAÇÃO ABRIGADA COM GERAÇÃO DE RELATÓRIO NO EXCEL**

Manaus

2018

**EDUARDO LIMA MANGUEIRA**

**SISTEMA REMOTO DE ENSAIOS DE ROTINA EM TRANSFORMADORES EM  
SUBESTAÇÃO ABRIGADA COM GERAÇÃO DE RELATÓRIO NO EXCEL**

Projeto de Pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentado à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletricista.

Orientador: Prof, Me. Raimundo Cláudio Souza Gomes

**Manaus**

**2018**

**Universidade do Estado do Amazonas – UEA**  
**Escola Superior de Tecnologia - EST**

*Reitor:*

***Cleinaldo de Almeida Costa***

*Vice-Reitor:*

***Cleto Cavalcante de Souza Leal***

*Diretor da Escola Superior de Tecnologia:*

***Roberto Higino Pereira da Silva***

*Coordenador do Curso de <Nome do Curso>:*

***Ingrid Sammyne Gadelha Figueiredo***

*Banca Avaliadora composta por:*

*Data da defesa: 06/12/2018.*

***Prof. Raimundo Cláudio Souza Gomes (Orientador)***

***Prof. Israel Gondres Torné***

***Prof. Carlos A. De Leon Benítez***

## **CIP – Catalogação na Publicação**

Mangueira, Eduardo Lima

Sistema Remoto De Ensaio De Rotina Em Transformadores Em Subestação Abrigada Com Geração De Relatório No Excel/ Eduardo Lima Mangueira; [orientado por] Raimundo Cláudio Souza Gomes. – Manaus: 2018.

81 p.: il.

Trabalho de Conclusão de Curso (Graduação em <nome do curso>). Universidade do Estado do Amazonas, 2018.

1. Arduino. 2. Transformador. 3. Ensaio. 4. Subestação. 5. Excel.  
I. Gomes, Raimundo Cláudio Souza.

EDUARDO LIMA MANGUEIRA

SISTEMA REMOTO DE ENSAIOS DE ROTINA EM TRANSFORMADORES EM  
SUBESTAÇÃO ABRIGADA COM GERAÇÃO DE RELATÓRIO NO EXCEL

Projeto de pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletricista.

Nota obtida: 9,50 ( \_\_\_\_\_ )

Aprovada em 06 / 12 / 2018.

Área de concentração: Subestação

BANCA EXAMINADORA

\_\_\_\_\_  
Orientador: Raimundo Cláudio Souza Gomes, Me.

\_\_\_\_\_  
Avaliador: Israel Gondres Torné, Dr.

\_\_\_\_\_  
Avaliador: Carlos A. De Leon Benitez, Dr.

Manaus 2018

## **DEDICATÓRIA**

Ao nosso bom Deus.

Aos meus pais Emerson e Fernanda, que batalharam para dar ferramentas de estudos aos seus filhos. Aos meus avós, tios e irmãos que estavam presentes apoiando.

Ao professor Charles, que fica na memória como grande professor e amigo.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, presente em todo momento para que eu possa crescer bastante, dando-me sabedoria, discernimento e força durante todas as dificuldades.

Agradeço aos meus pais Emerson e Fernanda por estarem presentes na minha vida, dando suporte e condições para minhas lutas e me amando bastante todo dia.

Agradeço aos meus irmãos Mateus, Filipe, Gabriele e Joel por serem pessoas indispensáveis na minha vida, por me proporcionar momentos incríveis e me fazer se sentir amado todo dia.

Agradeço aos meus avós e tios por fazerem parte das minhas lutas diárias e estarem me apoiando e dando força em todos os momentos, e em especial as minhas vós que não me deixaram emagrecer nos períodos difíceis.

Agradeço aos meus amigos, que a universidade trouxe para minha vida, e sem eles a vida universitária seria muito chata e difícil.

Agradeço ao meu orientador que teve um papel importante para a conclusão deste trabalho, auxiliando, apoiando e transmitindo conhecimentos.

Agradeço a todos os professores da UEA, que me fizeram desenvolver a capacidade de realizar um trabalho científico.

*” Deus não deixa ninguém esperando, a menos que julgue  
que essa espera seja benéfica ”*

C.S. Lewis

## RESUMO

O presente trabalho propõe construir um controlador automatizado para ensaios de transformadores em subestações abrigadas, possibilitando o controle das medições sem a necessidade de um operador dentro da subestação e reduzindo o tempo de uma manutenção preventiva completa. O trabalho surgiu com a necessidade de tornar o processo rápido, mais fácil de controlar e mais seguro, pois o mesmo é considerado um serviço com risco alto. O componente principal que é responsável pelo controle é um microcontrolador Arduino MEGA. Ele irá controlar a conexão do protótipo com uma interface de controle, os módulos de medição e enviará os resultados para o usuário através do Excel. O chaveamento entre os módulos de medição e os barramentos do transformador será realizado por relés, alguns deles com fabricação mais específica para operar com isolamento superior a 15 kV. O diferencial do projeto é a otimização dos ensaios em transformadores quando o mesmo já se encontra em uso e sem a necessidade de retirá-lo do local ou colocar técnicos para realizar o serviço. Além de apresentar os dados de forma organizada no Excel, possibilitando gerar banco de dados e adaptações para outros modelos de relatório.

Palavras-chave:

Arduino – Transformadores – Ensaio – Subestação – Excel



## **ABSTRACT**

This work was done with the aim of automating the tests of transformers in substations, allowing the control of the measurement without the need of an operator inside the substations and reducing the time of complete preventive maintenance. The work came with the intention of becoming fast, easier to control and more secure because it is a high risk service. The main component is an Arduino MEGA microcontroller. It controls a prototype connection with a control interface, the measurement modules and selecting results for the user through Excel. The switching between measurement modules and the transformer busbar is performed by relay, some of them with a more specific manufacturing to operate with insulation superior higher than 15 kV. The innovation of this project is the optimization of the tests on transformers when it is already in use and without the need of moving it or hire technicians to do the service. Besides showing the data in organized form in Excel, making it possible to generate the database and adaptations for other report templates.

Keywords: Arduino – Transformer – Test – Substation – Excel

## ÍNDICE DE ILUSTRAÇÃO

Figura 1: Curva performance do equipamento x manutenção corretiva .....	19
Figura 2: Curva performance do equipamento x custo para reparo .....	20
Figura 3: Ensaio de medição da resistência de isolamento .....	23
Figura 4: Ensaio de medição da relação de espiras do transformador.....	25
Figura 5: Modelos de placas Arduino .....	29
Figura 6: Exemplos de módulos de Arduino.....	29
Figura 7: ARDUINO IDE .....	30
Figura 8: Fluxograma de operação do sistema de ensaios de transformadores.....	33
Figura 9: Modelo de relatório de ensaio técnico .....	36
Figura 10: Arduino MEGA simulado no Proteus.....	37
Figura 11: Representação do relé no Proteus .....	38
Figura 12: Polarização do transistor .....	38
Figura 13: Esquema de comutação dos barramentos do transformador.....	39
Figura 14: Esquema de chaveamento para controlar o módulo megômetro .....	40
Figura 15: Interface do Virtual Serial Port.....	40
Figura 16: Linhas de código para armazenar dados na EEPROM do arduino.....	41
Figura 17: Linhas de código para leitura da EEPROM do Arduino.....	41
Figura 18: Interface de controle.....	42
Figura 19: Linha de código para declaração dos botões.....	43
Figura 20: Linha de código para declaração da área de trabalho .....	43
Figura 21: Linha de código para busca das portas serial.....	44
Figura 22: Linha de código com as funcionalidades do botão open .....	45
Figura 23: Linhas de código para os botões Ligar, Abrir e Iniciar.....	46
Figura 24: Interface de controle do Excel .....	47
Figura 25: Linhas de comando para gerar a conexão entre o Excel e o Arduino.....	48
Figura 26: Linhas de código para desativar a conexão do Excel com o Arduino .....	48
Figura 27: Linhas de código para receber os dados do Arduino .....	49
Figura 28: Linhas de código para gerar um novo relatório e salvar .....	49
Figura 29: Simulação da comunicação serial da interface .....	51
Figura 30: Componente que estabelece conexão do Arduino com a serial.....	52
Figura 31: Conectando o Excel com a serial .....	53
Figura 32: Geração de novos relatórios no Excel.....	53
Figura 33: Relatório final com os dados coletados do protótipo.....	54
Figura 34: Fluxograma da coleta de dados e montagem do relatório.....	55

## SUMÁRIO

<b>INTRODUÇÃO</b> .....	11
<b>1 REFERENCIAL TEÓRICO</b> .....	13
1.1 SUBESTAÇÃO.....	13
<b>1.1.1 Classificação das Subestações</b> .....	13
1.1.1.1 Quanto a Função: .....	13
<b>1.1.2 Quanto ao nível de tensão</b> .....	15
1.1.2.1 Tipo de instalação.....	15
1.1.2.2 Forma de operação .....	17
1.2 MANUTENÇÃO .....	17
<b>1.2.1 Manutenção Corretiva</b> .....	18
<b>1.2.2 Manutenção Preventiva</b> .....	20
<b>1.2.3 Manutenção Preditiva</b> .....	21
1.3 ENSAIOS EM SUBESTAÇÃO.....	21
<b>1.3.1 Ensaios em transformador</b> .....	22
1.3.1.1 Resistência de isolamento .....	22
1.3.1.2 Relação de Espiras .....	24
1.3.1.3 Resistência elétrica dos enrolamentos.....	25
1.3.1.4 Resistência de aterramento.....	26
1.3.1.5 Análise do óleo isolante .....	27
1.4 MICROCONTROLADORES .....	28
<b>1.4.1 Arduino</b> .....	28
<b>2 METODOLOGIA</b> .....	31
2.1 DESENVOLVIMENTO DO PROTÓTIPO.....	34
2.2 DESENVOLVIMENTO DA INTERFACE DE CONTROLE .....	35
<b>3 IMPLEMENTAÇÃO</b> .....	37
3.1 MONTAGEM DO PROTÓTIPO.....	37
3.2 MONTAGEM DA INTERFACE.....	42

<b>4 TESTES E RESULTADOS .....</b>	<b>50</b>
4.1 TESTE.....	50
4.2 RESULTADOS.....	53
<b>CONCLUSÃO .....</b>	<b>57</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>59</b>
<b>OBRAS CONSULTADAS.....</b>	<b>61</b>
<b>APÊNDICE A – CÓDIGO FONTE DO PROJETO (ARDUINO) .....</b>	<b>62</b>
<b>APÊNDICE B – CÓDIGO DA INTERFACE (SCREEN).....</b>	<b>69</b>
<b>APÊNDICE C – CÓDIGO DA INTERFACE (MAIN) .....</b>	<b>71</b>
<b>APÊNDICE D – CÓDIGO FONTE PARA GERAÇÃO DO RELATÓRIO NO EXCEL .....</b>	<b>73</b>
<b>APÊNDICE E – PROTÓTIPO MONTADO NO PROTEUS .....</b>	<b>78</b>

## INTRODUÇÃO

A segurança no trabalho em qualquer atividade vem crescendo cada vez mais e ficando mais exigentes dentro do ambiente industriário. A priorização das empresas é garantir a segurança e a saúde dos seus colaboradores e de terceiros. O setor de manutenção executa as atividades que apresentam riscos maiores dentro de uma empresa, pois geralmente são atividades envolvendo reparos de máquinas ou manutenção da infraestrutura do prédio, e os cuidados são redobrados quando os trabalhos estão envolvendo eletricidade em média tensão.

A manutenção preventiva é um serviço realizado com o objetivo de coletar informações importantes para diagnósticos de possíveis falhas funcionais ou falhas potenciais. A manutenção preventiva de uma subestação particular é um serviço exigido pelos bombeiros e serve para gerar e manter o alvará de funcionamento de qualquer empresa. Essa manutenção deve acontecer pelo menos uma vez ao ano. A maioria dos consumidores são de média tensão, ou seja, alimentados com 13.000 volts ou 69.000 volts. Quando uma manutenção na subestação é solicitada, uma sequência de procedimentos de segurança é exigida pelo Serviço Especializado em Engenharia de Segurança e em Medicina do Trabalho e é emitido documentação com assinatura de todos os responsáveis com a apresentação das pessoas capacitadas para realizar a atividade. E para realizar o serviço é necessário desligar totalmente a subestação, atividade solicitada à concessionária de energia, deixando o estabelecimento totalmente sem energia elétrica num período de 8 horas, prazo estimado para realização de uma sequência de ensaios dentro da subestação. O processo em si é um pouco demorado devido a quantidade de equipamentos e a quantidade de ensaios que devem ser realizados.

Os ensaios são os serviços mais importantes realizados dentro de uma manutenção preventiva de uma subestação e geram informações importantes para definir a qualidade dos equipamentos da subestação. Para realizar ensaio é necessário a utilização de diversos instrumentos de medição e uma equipe qualificada. Os instrumentos são aplicados no transformador para medição de parâmetros essenciais para emissão de laudos.

Diante da complexidade do serviço, das exigências de segurança e com o grande crescimento da Internet das Coisas, é possível realizar os ensaios dentro de uma subestação de maneira automática, sem a necessidade de uma equipe no local. Neste trabalho apresenta uma proposta com o objetivo de realizar os ensaios de rotinas em transformadores controlados por apenas um operador qualificado e sem a necessidade do mesmo está dentro da subestação e agilizando o processo de manutenção para que o estabelecimento volte a receber energia elétrica o mais rápido possível. Com um sistema de controle, os serviços dentro de uma subestação

podem ser reduzidos, e controlando um risco existente dentro da empresa, pois não será necessário tráfego de pessoas dentro de um ambiente perigoso. Este sistema realizara todos os ensaios de maneira automática com apenas um operador na parte externa da subestação.

Será feita uma revisão teórica abrangendo os seguintes assuntos: subestação, manutenção, ensaios em transformadores e microcontroladores. Em seguida, no capítulo 2 é mostrado a metodologia aplicada no projeto, nesta etapa é dividido em duas seções, com as metodologias aplicadas para o desenvolvimento do protótipo e para o desenvolvimento da interface de controle. No capítulo 3 contém a montagem e execução do projeto, e nesta etapa é mostrado a montagem do protótipo, programação da interface e programação do Excel para elaboração do relatório. Por fim, os testes e resultados são apresentados no capítulo 4.

# 1 REFERENCIAL TEÓRICO

## 1.1 SUBESTAÇÃO

Subestações são um conjunto de equipamentos, máquinas, aparelhos e circuitos destinados a modificar os níveis de tensão e corrente. São responsáveis pela transmissão e distribuição da energia elétrica. Durante o percurso entre os centros de geração até os centros de consumo, a energia elétrica passa por diversas subestações, elevando a tensão elétrica para reduzir as perdas elétricas e transmitir para longas distâncias, e abaixando até o consumo final (TECNOGERA, 2015).

Segundo a ABNT NBR 5460:1992, subestação é:

Parte de um sistema de potência, concentrada em um dado local, compreendido primordialmente as extremidades de linhas de transmissão e/ou distribuição, com os respectivos dispositivos de manobra, controle e proteção, incluindo as obras civis e estruturas de montagem, podendo incluir também transformadores, equipamentos conversores e/ou outros equipamentos.

As subestações podem ser classificadas segundo a função, nível de tensão, tipo de instalação e forma de operação. A escolha da subestação depende de diversos fatores, como por exemplo a sua função (classificação segundo a função), local onde será empregada (classificação tipo a instalação), quanto ao potencial de entrada e a ser transformado (classificação tipo nível de tensão) (UNIVERSO ELÉTRICO, 2016).

### 1.1.1 Classificação das Subestações

#### 1.1.1.1 Quanto a Função:

##### a) Subestação Elevadora

São subestações transformadoras onde o potencial elétrico na saída é maior que do da entrada. São localizadas na saída das usinas geradoras, elevando a tensão de saída dos geradores para níveis de transmissão e sub-transmissão. São essenciais para transmitir a energia elétrica para longas distâncias, pois quando se eleva a tensão elétrica, a corrente elétrica abaixa como consequência e assim reduz as perdas Joules nas redes elétricas. Estas subestações apresentam transformadores elevadores de potência elevadas e uma serie de equipamentos auxiliares. (NBR 5460, 1992).

#### b) Subestação Abaixadora

São subestações que fazem o papel inverso das subestações elevadoras. A tensão de saída é menor que tensão de entrada. São localizadas dentro ou próximo aos centros urbanos, e quando a energia elétrica não precisa percorrer grandes distâncias até o próximo ponto. Dentro dos centros urbanos, linhas elétricas devem ter potencial menor que das linhas de transmissão por vários motivos, tanto econômico quanto de segurança. Para manter redes elétricas com alta tensão, o sistema de proteção é mais caro, e as torres de sustentação dos cabos são maiores e consequentemente mais caros, além do campo elétrico gerado pelas linhas de alta tensão afetarem outros sistemas, como, por exemplo, sistemas de telecomunicações. As subestações possuem transformadores abaixadores para reduzir a tensão para potenciais padronizados pela concessionária local (NBR 5460, 1992).

#### c) Subestação de Distribuição

As subestações de distribuição têm a função parecida com as subestações abaixadora, abaixa a tensão para níveis de tensão adequado para distribuição. Estas subestações encontram-se dentro dos centros urbanos e podem pertencer a concessionária ou a grandes consumidores. Estas subestações abaixam a tensão para média tensão que é classificada como nível primário de distribuição, ou seja, 13,8 kV à 34,5 kV (NBR 5460, 1992).

#### d) Subestação de Manobra

É uma subestação com o objetivo principal é modificar a configuração do sistema elétrico, com as modificações das interligações entre linhas de transmissão. Esta subestação não altera os níveis de tensão, apenas realiza manobras inserindo ou retirando partes do sistema de transmissão (NBR 5460, 1992).

#### e) Subestação Conversora

As subestações conversoras têm a finalidade de realizar uma mudança na forma de onda que circula nas linhas elétricas. Elas podem ser subestação conversora de frequência onde é realizado a mudança da frequência da energia elétrica, ou subestação conversora de corrente,



onde a corrente é alterada de alternada pra contínua ou vice-versa, ou seja, podem ser retificadoras ou inversoras (NBR 5460, 1992).

### **1.1.2 Quanto ao nível de tensão**

As subestações podem ser classificadas quanto ao nível tensão, podendo ser de média tensão, alta tensão ou extra alta tensão e essa classificação pode variar de acordo com a concessionária. Para a Agência Nacional de Energia, Elétrica, ANEEL, classifica como extra alta tensão (superior a 230 kV), alta tensão (superior a 69 kV e inferior a 230 kV) e média tensão (inferior a 69 kV) (ANEEL, 2015).

#### **1.1.2.1 Tipo de instalação**

Existem vários tipos de instalações de subestações, a construção de uma subestação está associada a fatores econômicos, confiabilidade e continuidade requeridas pela carga, flexibilidade de manobra operacional e facilidade de execução da manutenção. Os tipos de instalações podem ser divididos em três grupos principais: instalação abrigada ou instalação ao tempo.

##### **a) Instalação abrigada**

São subestações onde os equipamentos e aparelhos são instalados em dependência abrigadas das condições climáticas. A seguir será feita uma descrição sucinta dos tipos de subestações abrigadas existentes.

Subestação em Alvenaria é um tipo mais comum de subestação, utilizada em grandes empreendimentos de comércio, indústrias, condomínios e etc. É um dos modelos de baixo custo e de fácil manutenção. Todos os equipamentos são instalados dentro da subestação que o tamanho pode variar de acordo com a capacidade, a área onde será instalada, o investimento e as especificações da NBR 14039 (2005) que estabelece padrões seguindo a ANEEL. A subestação de alvenaria é dividida em cabines (ou cubículos, ou postos) onde cada uma desempenha uma função bem definida, e as principais cabines são: cabine de transformação, onde contém o transformador de força, cabine de medição, onde é destinado a equipamentos de medição e cabine de proteção de média tensão, onde apresenta um disjuntor de média tensão e em alguns casos relé de proteção (MAMEDE, 2017).

Além das subestações abrigadas de alvenaria, existe as subestações em invólucro metálico, também conhecida como subestação modular metálica, são subestações aplicadas onde o espaço é limitado e seu uso pode ser tanto interno quanto ao tempo. Assim como as subestações de alvenaria, as subestações metálicas são divididas em cabines com as suas funções bem definidas. Em cada cabine é instalado os equipamentos pertencentes a função da cabine, por exemplo, na cabine de proteção de média tensão é instalada equipamentos como o disjuntor de média tensão, chave seccionadora, fusíveis e etc. Os fabricantes deste tipo de subestação trazem a vantagem de ampliar a subestação instalando módulos separadamente ou retirando dependendo da necessidade (MAMEDE, 2017).

#### b) Instalação ao tempo

São subestações instaladas a intempéries atmosférica e, normalmente, os aparelhos abrigados. Todos os equipamentos em área externa devem ser resistentes às intempéries em conformidade com a NBR 14039 (2005) – Instalações elétricas de média tensão 1,0 kV a 36,2 kV. Essas subestações podem ser classificadas em dois tipos de acordo com a montagem dos equipamentos (NBR 14039, 2005).

A subestação aérea em plano elevado é o primeiro tipo de subestação externa e neste tipo de subestação o transformador é fixado em torre ou plataforma onde geralmente são fabricadas de em concreto armado, aço ou madeira. A preocupação com as partes vivas é maior e estas devem estar situadas a uma altura mínima de 5 metros e quando esta altura não for possível, pode ser tolerado o limite de 3,5 metros, desde que o local seja provido de sistema de proteção de tela metálica ou equivalentes. Os equipamentos podem ser instalados em postes ou torres, em plataformas elevadas sobre estruturas de concreto, aço ou madeira adequada, em áreas sobre cobertura de edifício, desde que seja inacessível a pessoas não qualificadas e os transformadores não devem ser de óleo isolante. Algumas concessionárias limitam a potência dos transformadores instalados em postes em 150 kVA e quando for superior a esse valor deve ser instalado em dois postes (MAMEDE, 2017).

O outro tipo de subestação externa é a instalada no nível do solo em que disjuntores de média tensão, transformadores são instalados em bases de concreto construídas ao nível do solo e equipamentos como para-raios, chaves seccionadoras e fusíveis são instalados em estruturas aéreas. Esse tipo de subestação aplicada em local urbano é de custo elevado devido ao espaço ocupado pelos equipamentos, pois não são subestações compactas e conseqüentemente o preço do próprio terreno é elevado e os equipamentos devem atender os parâmetros mencionados no

item 4.3 da NBR 14039 (2005) que fala sobre classificação das influências externas. Já em áreas rurais essa subestação não se torna tão cara devido ao grande espaço disponível (MAMEDE, 2017).

#### 1.1.2.2 Forma de operação

As subestações podem ser classificadas de acordo com a forma de operação, ou seja, a presença de um operador ou não e isso pode agregar qualidade, melhorando o tempo de operação e outras qualidades. Essa classificação pode ser:

##### a) Subestação com operador

Exige um nível de treinamento alto para a equipe de operação com todos os cursos e treinamentos exigidos em normas como os cursos da NR 10 – Segurança em Instalações e Serviço em Eletricidade (Módulo I e Módulo II). Não necessita de controladores automáticos, mas a aplicação de um sistema supervisionado garante a segurança a confiabilidade da subestação e é importante em quando está é de grande porte.

##### b) Subestação semiautomáticas

Esta subestação possui computadores locais que impedem operações indevida por parte do operador local. Tem sistema de supervisão para sinalização quando aparece alguma ocorrência alertando o operador local para tomar uma ação.

##### c) Subestação automatizadas

São supervisionadas à distância por intermédio de computadores, este tipo de subestação não necessita de um operador local para tomar ações, uma equipe em outro local pode controlar sem a necessidade de ir até o local dependendo do nível da ocorrência. Apresenta todos os parâmetros online para supervisão e controle.

## 1.2 MANUTENÇÃO

Segundo o dicionário Aurélio (FERREIRA, 2010), manutenção pode ser definida como: “Medidas necessárias para a conservação ou permanência, de alguma coisa ou situação”. A NBR 5462 (1994) define: “Combinação de todas as ações técnicas e administrativas, incluindo as de supervisão, destinadas a manter ou recolocar um item em um estado no qual possa

desempenhar uma função requerida”. Como a norma informa, a manutenção é um conjunto de ações tanto técnica quanto administrativa para que um determinado equipamento exerça sua função como projetado para que este não pare de funcionar ou quando o mesmo está parado com defeito e precise voltar a funcionar (PAULINO, 2005).

Para realizar uma manutenção é preciso de um quadro de mão de obra qualificada e da implementação de um sistema, a qual integre todas as áreas da empresa. Paulino da revista Portal O Setor Elétrico fala que a missão da manutenção é garantir a disponibilidade da função dos equipamentos e instalações de modo a atender a um processo de produção ou de serviço, com confiabilidade, segurança, preservação do meio ambiente e custo (KARDEC E LAFRAIA, 2002).

Os benefícios gerados por um plano de manutenção são aumento da confiabilidade, melhoria na qualidade, diminuição dos custos, aumento da vida útil dos equipamentos e melhoria da segurança. Com um plano bem executado, os equipamentos podem durar mais tempo, os operadores estão seguros quanto a situações geradas em equipamentos danificados, benefícios para a empresa no sentido econômico e qualidade dos produtos ou serviços ofertados (PAULINO, 2005).

A manutenção pode ser dividida quanto a sua aplicação e em quatro tipos clássicos e básicos, a corretiva, preventiva, preditiva e a sistemática. Essas manutenções fazem parte do plano de gestão de manutenção no cotidiano do ambiente industrial. O plano de manutenção é capaz de detectar quando um equipamento ou peça está desgastado ou necessita de certos reparos. Acompanhado do plano de manutenção tem o histórico do equipamento que é um fator importante para a equipe de manutenção (ALMEIDA, 2017).

### **1.2.1 Manutenção Corretiva**

A manutenção corretiva é aquela que acontece após a ocorrência de algum defeito, ou seja, é uma manutenção sem programação e geralmente ocorre em situação de urgência. A NBR 5462 (1994) define manutenção corretiva sendo aquela efetuada após uma ocorrência de uma pane destinado a realocar um item em condição de executar uma função requerida. Uma pesquisa realizada pela Engeteles (Teles, 2018) a corretiva é o tipo de manutenção mais cara aplicada na indústria e que toma mais tempo e traz mais prejuízo. Pode chegar ser sete vezes mais cara que as demais manutenções existentes. A manutenção corretiva sempre vai atuar em uma falha e a falha pode ser dividida em dois tipos comuns, falha potencial ou falha funcional. A falha potencial é aquela que se encontra em fase inicial, já demonstra que algo está errado,

interfere ou não no funcionamento do equipamento e este ainda pode operar. A falha funcional ocorre quando o equipamento não pode mais funcionar, ou seja, a falha já atingiu níveis que interfere totalmente no processo da máquina (TELES, 2018).

Figura 1: Curva performance do equipamento x manutenção corretiva



Fonte: Teles (2018)

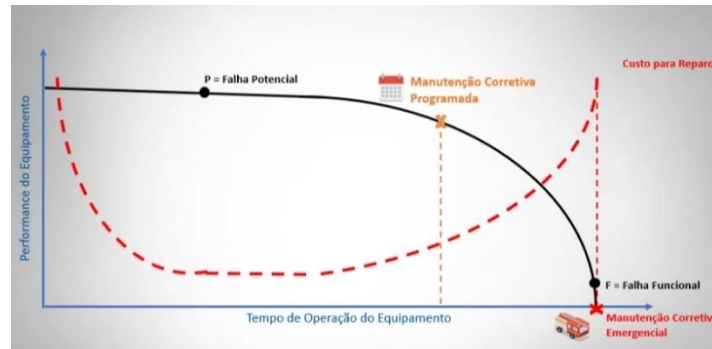
A manutenção corretiva pode se enquadrar em dois tipos, a programável e a não programável. A manutenção programável deve ocorrer antes de uma parada do equipamento e está relacionada a falha potencial, já a manutenção não programável acontece está relacionada a falha funcional, como é ilustrado na Figura 1. Podemos definir a manutenção não programável como sendo a correção de uma falha ou defeito após a ocorrência do fato e de maneira aleatória. A manutenção corretiva não programável é de classe emergencial e é aplicada quando apresenta a falha funcional do equipamento e por esse motivo o equipamento deve ser reparado em caráter de urgência (PAULINO, 2005).

A manutenção corretiva programada é o tipo de manutenção que opera em cima das falhas potenciais antes que ela evolua para uma falha funcional. Consiste numa atividade gerada pelo acompanhamento preditivo, detectivo, ou até uma decisão gerencial. Inicia-se analisando a falha, verificando a possibilidade de agendamento, decorrente de uma falha não prejudicial ao processo ou a segurança de um colaborador, e programada para um ser eliminada no momento conveniente para a empresa (PAULINO, 2014).

O tempo da detecção de um problema até a manutenção corretiva está diretamente relacionada ao custo gerado para a empresa. A Figura 2 mostra um gráfico do custo em função de uma intervenção corretiva. A partir do momento que foi identificado o problema até a programação para correção da falha o custo tende a crescer e se chegar a um ponto onde

necessite de uma correção emergência, a curva do custo de correção pode chegar a um ponto crítico. Por isso, é importante realizar a manutenção antes de uma parada emergencial.

Figura 2: Curva performance do equipamento x custo para reparo



Fonte: Teles (2018)

### 1.2.2 Manutenção Preventiva

A manutenção preventiva está relacionada a todas as atividades sistemáticas e monitoramento em máquinas que não se encontra em falha e realizada em intervalos predeterminados ou de acordo com critérios prescritos de acordo a reduzir a probabilidade de falha ou a degradação do funcionamento de um item. Junto com a preventiva tem o plano de manutenção preventivo, que serve para organizar as manutenções nos intervalos pré-determinado. Tem como objetivo a redução ou eliminação de falhas ou defeitos (NBR 5462, 1994).

Com um plano bem executado resultará na redução da necessidade de se adotar ações corretivas, evitando quebras e paradas não planejadas do processo, garantindo confiabilidade e qualidade e, podendo até aumentar a produtividade dentro da indústria. Resumidamente, a manutenção preventiva age antes da ocorrência de alguma falha. Para isso, deve-se realizar as medidas tecnicamente adequadas, as quais devem ser selecionadas entre uma grande variedade de alternativas.

Com uma programação preventiva, será possível identificar problemas e solucionar antes que se torne algo grande. Poderá identificar a necessidade de troca de uma peça a gerência de manutenção terá tempo para pesquisar e comprar a peça quando é identificado uma falha no início. E, assim como a manutenção corretiva, a manutenção preventiva atua nas falhas potências e falhas funcionais e o custo para reparar uma falha é proporcional ao tempo em que ela ocorreu, assim como é mostrado na Figura 2.

### 1.2.3 Manutenção Preditiva

A NBR 5462 (1994) defini a manutenção preditiva como sendo a manutenção que permite garantir uma qualidade de serviço desejada, com base na aplicação sistemática de técnicas de análise, utilizando-se de meios de supervisão centralizados ou de amostragem, para reduzir ao mínimo a manutenção preventiva e diminuir a manutenção corretiva. A manutenção preditiva também é conhecida pela norma como sendo manutenção controlada. A preditiva é composta por tarefas de manutenção preventiva que visa acompanhar a máquina ou peça, por monitoramento, por medição ou por controle estatístico (PAULINO, 2005).

A preditiva é considerada um acompanhamento periódico das variáveis ou parâmetros que indicam o desempenho do equipamento por meio de monitores e inspeções. Com ela poderá prever o tempo de vida útil do equipamento ou peça.

Entre as principais vantagens é a possibilidade de antecipar serviços de manutenção do equipamento, eliminar a chance de desmontagem desnecessária, aumentar a vida útil do equipamento, reduzir as paradas imprevistas, aumentar a confiabilidade e segurança do equipamento.

## 1.3 ENSAIOS EM SUBESTAÇÃO

Ensaio são medições realizadas nos equipamentos ou componentes para realizar análise segundo mais de fabricantes e normas de referência. Existem diversos ensaios e cada equipamento tem ensaios específicos para determinar as condições do mesmo. Com o resultado dos ensaios, pode-se analisar a vida útil, a qualidade, e a eficiência do equipamento em questão, e, em caso extremos, pode-se gerar laudos de obsolescência em caso que os ensaios mostram condições não aceitáveis pelas normas.

Com os ensaio elétricos pode-se tirar informações como a característica dos materiais como a condutibilidade dos condutores, quase condutores e dielétricos, medir a isolação de um determinado equipamento, a permissividade, resistividade, relaxação dielétrica, comportamento do material com frequência, geometria do campo elétrico aplicado, a capacidade de corrente, descarga, deslocamento, fator de potência, dissipação e outros. Pode-se perceber que existe diversos ensaios e cada um para uma aplicação diferente, como a medição da resistência de isolação que aponta a isolação de equipamento e em seguida é comparado com valores de fábrica e apontados pelas normas.

Em uma subestação, realiza-se ensaios em transformadores de força, transformadores de potência, transformadores de corrente, muflas, buchas, disjuntor de média tensão, chave seccionadora, chave fusível, para-raios, circuito de aterramento, cabos, isoladores e outros equipamentos não comuns em subestações. A ABNT NBR 14039 (2005) informa que os ensaios da instalação devem incluir no mínimo os seguintes itens:

1. Continuidade elétrica dos condutores de proteção e das ligações equipotenciais e suplementares;
2. Resistência de isolamento da instalação elétrica;
3. Ensaio de tensão aplicada;
4. Ensaio para determinação da resistência de aterramento;
5. Ensaio recomendados pelos fabricantes dos equipamentos;
6. Ensaio de funcionamento.

Dentro de uma subestação de 13,8 kV do tipo consumidor os equipamentos que são essenciais para inspeção são: transformador de força, disjuntor de média tensão, chave seccionadora e mufla. Será citado para cada um deles os tipos de ensaios realizados e descrever tais ensaios.

### **1.3.1 Ensaio em transformador**

O transformador é o equipamento principal de uma subestação e o mais complexo. Existem vários ensaios realizados em um transformador, para efeito de uma manutenção preditiva é realizado o ensaio de isolamento, o ensaio de medição da relação de espiras, medição da resistência das espiras, medição da resistência de aterramento do transformador e análise do óleo para transformadores com isolamento à óleo. A norma de referência para ensaios em transformadores é a ABNT NBR 5356-1:2007 Versão Corrigida: 2010 - Transformadores de potência.

#### **1.3.1.1 Resistência de isolamento**

Este ensaio é realizado com o objetivo de detectar falhas na isolamento do transformador, ou seja, diagnosticar a condição dielétrica do material isolante. A resistência de isolamento representa a dificuldade da corrente elétrica passa pelo material isolante e seus valores podem



variar de acordo com a umidade e com a sujeira (Paulino, 2014). O ensaio consiste em aplicar uma tensão de 500 V até 2 kV em corrente contínua e isso provocará um fluxo pequeno de corrente. Utiliza-se um megaohmímetro de 1 kV no mínimo, para enrolamentos de tensão máxima de equipamento igual ou inferior a 72,5 kV, e de 2 kV no mínimo, para enrolamentos de tensão máxima do equipamento superior a 72,5 kV (NBR 5356-1, 2007).

Curto-circuitar os terminais de cada enrolamento do transformador sob ensaio, ou seja, ligar todos os terminais do primário através de um condutor e todos os terminais do secundário, incluindo o neutro, através de outro condutor. Ligar o megaohmímetro, mantendo-se a tensão constante durante no mínimo 1 minutos e fazer a leitura. Em seguida fazer uma outra medida em 5 minutos e fazer anotação dos valores. No transformador, deve-se medir a isolação do primário com o secundário, do primário com a carcaça do transformador, e do secundário com a carcaça. Para as duas primeiras medições, utiliza-se o megaohmímetro deve estar no mínimo em 1 kV e para a última medição o megaohmímetro deve estar no mínimo em 500 V e cada uma das medições deve-se realizar as medições em 1 minuto e em 5 minutos. Na Figura 3 é ilustrada uma medição utilizando um megaohmímetro em um transformador para medição da resistência de isolamento (PAULINO, 2014a).

É importante que o transformador esteja totalmente isolado do circuito, se possível desconecta os cabos do secundário e abrir a chave seccionadora do lado primário do transformador. Caso não seja possível desconectar os cabos do secundário, deve-se desaterrar o neutro (caso seja sistema neutro aterrado) e abrir o disjuntor geral do QGBT (quadro geral de baixa tensão).

Figura 3: Ensaio de medição da resistência de isolamento



Fonte: Própria

A avaliação é realizada comparando os valores obtidos na medição de 1 minuto e 5 minutos e calculando o índice de absorção. Este índice deve apresentar valor superior a 1,0 para uma condição de isolamento, caso contrário o transformador pode estar com a isolação danificada (Paulino, 2014). O cálculo do índice de absorção é feito pela fórmula abaixo:

$$I_{AB} = R_{1min} / R_{5min} \quad (1)$$

onde:

$I_{AB}$  é índice de absorção;

$R_{1min}$  é resistência medida em 1 minuto;

$R_{5min}$  é resistência medida em 5 minutos.

O ensaio da resistência de isolamento, basicamente, informa a degradação da isolação do transformador e pode ser acompanhada com testes periódicos e quando a isolação está baixa pode ocasionar problemas sérios como curto-circuito e por isso o ensaio é um dos mais importantes para uma manutenção preditiva em uma subestação.

#### 1.3.1.2 Relação de Espiras

Neste ensaio se mede a relação de transformação em cada derivação (ou TAP) e este ensaio é padronizado como um ensaio de rotina pela ABNT NBR 5356-1:2007 sendo um teste básico em programas de manutenção preditiva e preventiva, reparos ou quando os transformadores são submetidos a reformas ou, ainda, no comissionamento. É um recurso valioso para se verificar a existência de espiras em curto-circuito, falhas nos comutadores de derivação (TAP) e ligação erradas de derivação. Para transformadores trifásicos, apresentando fases independentes e com terminais acessíveis, procede-se, indiferentemente, usando-se corrente monofásica ou trifásica, como for mais conveniente. Se o enrolamento onde a tensão é aplicada estiver ligado em estrela com neutro inacessível, usa-se corrente trifásica, procedendo-se como para transformadores monofásicos (NBR 5356-1, 2007).

A norma mostra diversos métodos utilizados para realizar a medição da relação de transformação e entre eles tem o método do voltímetro, do transformador-padrão, do transformador de referência de relação variável e do potenciométrico. Além destes métodos, existe um outro método, ou melhor, um equipamento que realiza a medição da relação de transformação se baseando em algum método listado na normal. Este equipamento é

denominado de *Transformer Turn Ratio* (TTR) que é específico para medição da relação de transformação e é mostrado na Figura 4 em um ensaio no transformador. Os valores obtidos por qualquer método devem ser suficientemente precisos onde o erro não possa passar de  $\pm 0,5\%$  em todos os TAPs de comutação (PAULINO, 2014c).

O erro percentual é calculado em função dos valores medidos e valores nominais, sendo:

$$E_{\%} = \frac{R_{med} - R_{nom}}{R_{nom}} \times 100 \quad (2)$$

onde:

$E_{\%}$  é o erro percentual;

$R_{med}$  é a relação medida;

$R_{nom}$  é a relação nominal.

Figura 4: Ensaio de medição da relação de espiras do transformador



Fonte: Própria

### 1.3.1.3 Resistência elétrica dos enrolamentos

O ensaio de medição elétrica dos enrolamentos mede a resistência ôhmica de cada enrolamento do transformador e tem como objetivo detectar com precisão a vida útil do transformador com o envelhecimento das bobinas e detectar curto-circuito que ocorreram anteriormente na rede. Neste ensaio devem ser registrar a resistência elétrica de cada

enrolamento, os terminais entre as quais é medida a resistência elétrica e a temperatura dos enrolamentos para correção dos valores posteriormente (PAULINO, 2014b).

A medição deve ser realizada com corrente contínua por um método de ponte ou pelo método de queda de tensão. Caso a corrente nominal do enrolamento for menor que 1 A, deve utilizar o método de ponte. A corrente de ensaio não deve superar 15% da corrente nominal do enrolamento. O valor da resistência é muito baixo na escala de miliohms, e com o método de ponte pode ser obtida com precisão o valor da resistência.

O equipamento utilizado para medição é o microhmímetro, ideal para medição de resistência em escala baixa e com uma precisão alta. O microhmímetro funciona com a metodologia de Kelvin para evitar erros na medição provocados pelos cabos auxiliares e as resistências de contato (PAULINO, 2014b).

Os resultados obtidos devem ser comparados com os valores do fabricante e os valores de ensaios anteriores e realizando a correção de temperatura quando for analisar o resultado final. Para transformadores, a temperatura de referência normalmente é 75 °C, mas esse valor pode mudar de acordo com as especificações do fabricante. O valor obtido do ensaio após a correção de temperatura tem que apresentar uma variação de  $\pm 5\%$  e caso exceda esse valor, devem ser pesquisadas anomalias no enrolamento tais como: espira em curto circuito, número incorreto de espiras, dimensões incorretas do condutor e outros. É importante ter o histórico de medidas para comparação e determinação de defeitos no enrolamento. Com o histórico é possível determinar a vida útil de um transformador e saber quando realizar manutenção.

#### 1.3.1.4 Resistência de aterramento

Este ensaio é importante para determinar a resistência de aterramento aplicado a carcaça do transformador e essa resistência deve ser suficientemente baixa para garantir a segurança do empreendimento onde a subestação se encontra e operadores no local. A norma que estabelece os critérios e métodos da medição de resistência de sistemas de aterramento definindo as características gerais dos equipamentos que podem ser utilizados nas medições e também os conceitos para avaliação dos resultados é a ABNT NBR 15749:2009 – Medição de resistência de aterramento e de potenciais na superfície do solo em sistemas de aterramento.

A norma define aterramento como sendo uma ligação intencional de parte eletricamente condutiva à terra, através de um condutor elétrico. A resistência de aterramento é uma resistência resultante medida no ponto onde um determinado equipamento está aterrado até o solo. Para a medição, pode-se utilizar o termômetro alicate, um dispositivo onde realiza a

medição sem a necessidade de desconectar o cabo de aterramento. O funcionamento do alicate terrômetro consistem em um gerador de corrente alternada que aplica uma tensão numa bobina onde o núcleo envolve o circuito fechado a qual representa uma única espira e neste circuito fechado será produzido uma força eletromotriz conhecida e basta apenas medir a corrente circulando por este circuito e por fim calcular a resistência. O alicate terrômetro realiza todo esse procedimento, bastando apenas o operador colocar o alicate na posição adequada para medição da resistência de aterramento no transformador.

#### 1.3.1.5 Análise do óleo isolante

Para transformadores à óleo, é importante verificar as condições físico-químicas e cromatográficas do óleo isolante. O óleo isolante tem duas principais funções em um transformador, serve como dielétrico e agente de transferência de calor e para isso o mesmo deve possuir certas propriedades, tais como, rigidez dielétrica suficiente para suportar as tensões impostas pelo serviço e viscosidade adequada para sua capacidade de circular e transferir calor (DATALINK ENGENHARIA ELÉTRICA, 201?).

A deterioração do óleo é causada por fatores como a temperatura elevadas ocasionada pelo transformador, presença de oxigênio, presença de umidade e a aplicação em metais também contribui na deterioração. A qualidade do óleo pode ser avaliada através de ensaios laboratoriais tais como a análise físico-química e cromatográfica. Com a análise físico-química, normatizada pela ABN NBR 7036:1990 – Recebimento, instalação e manutenção de transformadores de potência para distribuição, imersos em líquidos isolantes, pode-se retirar dados quanto o teor de água, a densidade do óleo, tensão interfacial, rigidez dielétrica, fator de potência, índice de neutralização e a cor. O resultado do ensaio determina a capacidade de isolamento e o envelhecimento do óleo. Os resultados devem ser comparados com os resultados anteriores e estabelecido por normas e caso apresente valores fora dos padrões necessitará de tratamento como o termo-vácuo, substituição ou regeneração (NBR 7036, 1990).

A análise cromatográfica informa a concentração dos gases presentes no óleo do transformador tais como, hidrogênio, oxigênio, nitrogênio e outros. A formação dos gases dentro do transformador caracteriza problemas como mau contato entre as bobinas, fuga de energia, esforço à altas correntes de curto e tempo de trabalho prolongado com cargas elevadas. Este ensaio é normatizado pela ABNT NBR 7070:2006 – Amostragem de gases e óleo mineral isolantes de equipamentos elétricos e análise dos gases livres dissolvidos e a norma que é

utilizada para analisar os dados é a ABNT NBR 7274:2012 – Interpretação da análise dos gases de transformadores em serviço.

## 1.4 MICROCONTROLADORES

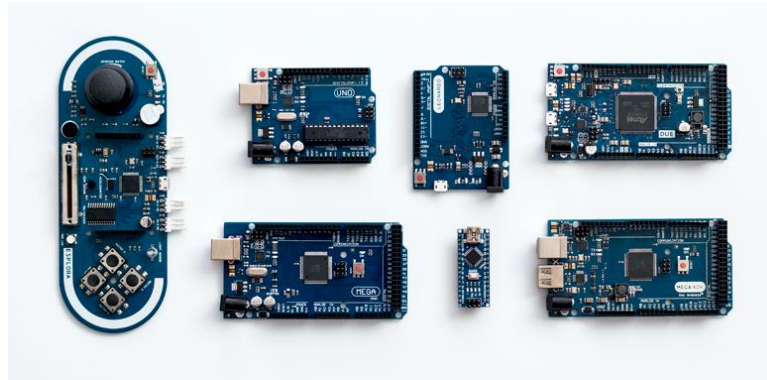
Microcontroladores são dispositivos que utilizam a união de *hardware* com *software* para realizar um conjunto de atividade. É um pequeno computador, a qual tem um núcleo de processamento, memória, e periféricos programáveis de entrada e saída. Em geral, os microcontroladores são usados para controlar circuitos específicos seguindo linhas de algoritmos escritos em sua memória.

### 1.4.1 Arduino

O Arduino é uma plataforma eletrônica de hardware e software de código aberto e livre. Uma placa Arduino é considerada como sendo um minicomputador, e podemos definir um computador como sendo uma máquina que processa instruções e essas instruções são processadas no processador do computador. Todo computador possui um processador, o processador do Arduino é o ATmega, também chamado de microcontroladores. O Atmega é tem suporte para entradas e saídas analógicas e digitais embutido, linguagem de programação parecido com a linguagem C/C++ e de fácil interpretação, interface serial ou USB para interligar-se ao hospedeiro que pode ser utilizado para programar o Arduino e realizar interações em tempo real (ARDUINO, 2018).

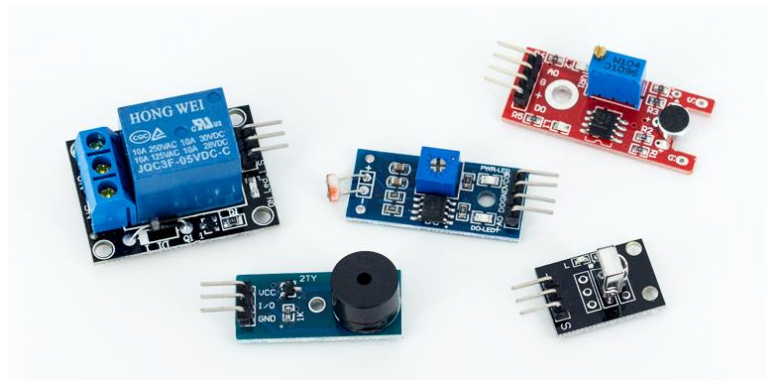
Uma placa de Arduino pode ser utilizado para projetos independentes de controle de atividades simples e algumas complexas, onde depende da necessidade de portas de entrada e saída. O Arduino pode ser conectado a outro computador e realizar comandos e troca de informações. O microcontrolador tem no mercado em diversos modelos como é ilustrado na Figura 5, variando a quantidade de portas e algumas funcionalidades. Fora suas funcionalidades embutidas, existe no mercado uma gama de sensores e atuadores com aplicação direta com o Arduino e a Figura 6 mostra alguns exemplos, aumentando a vasta área de atuação do processador. Grandes partes dos acessórios são disponíveis em pequenas placas chamadas de módulos, deixando o acessório mais fácil de ser utilizado. Além dos sensores e atuadores, existem também as Shields, que são placas que se encaixam no Arduino aumentando algumas funcionalidades específicas.

Figura 5: Modelos de placas Arduino



Fonte: Thomsen, 2014

Figura 6: Exemplos de módulos de Arduino



Fonte: Thomsen, 2014

O campo de atuação para controle de sistema é vasto, tendo aplicações com robótica, smartgrid, sistemas supervisionados, sistemas de controle, engenharia médica, engenharia clínica, sistemas de telecomunicação e entre outros. Hoje sua aplicação atende qualquer tipo de atividade, se limitando pelo processamento e a quantidade de portas de entrada e saída.

A programação é feita pela plataforma própria do Arduino que é gratuito no site oficial e tem uma estrutura parecida com a linguagem de programação C++, trabalhando como algoritmo. O algoritmo, ou programa, é a forma de informar ao computador as atividades que ele deve realizar de forma que um humano consiga entender facilmente. O IDE do Arduino apresenta duas estruturas principais, o *setup()* que encontra-se no início do programa e serve para inicializar configurações e o *loop()* que é seguido do *setup* é um comando que se repete infinitamente até que seja desativado. A Figura 7 mostra a interface do Arduino IDE com as duas principais funções (CHAVIER, 2018).

Figura 7: ARDUINO IDE



```
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

Fonte: Arduino, 2018



## 2 METODOLOGIA

O presente trabalho apresentado é uma Pesquisa Aplicada, e tem como objetivo uma pesquisa exploratória sobre o material bibliográfico e do assunto abordado. Tem utilizado método técnico de pesquisas bibliográficas. O método de abordagem utilizado será o hipotético-dedutivo e o método de procedimento de elaboração será o monográfico. O desenvolvimento do protótipo é realizado em uma plataforma de simulação e os dados simulados são retirados de manutenções preventivas realizada em diversas subestações e colocado na entrada de dados do protótipo. O relatório da manutenção realizada pelo protótipo será emitido em uma planilha no Excel contendo todos os dados e comparativos com as normas regulamentadoras. O relatório será realizado com base nas normas regulamentadoras, fabricantes, profissionais da área com experiencia de campo e livros de referência.

Foram feitas pesquisas bibliográficas voltadas para subestação, manutenções em subestações, ensaios em transformadores, instalações em média tensão, sistemas de proteção contra descargas atmosférica, sistema de aterramento, normas que regulamentam as instalações de média tensão e instalação de subestação. Foram realizadas visitas técnicas em subestações e em indústria que fabrica transformadores a fim de conhecer os processos de manutenção preventiva em transformadores de subestação abrigada. O projeto do protótipo foi simulado no Proteus e para desenvolvimento do circuito, foram realizadas pesquisas nas matérias de eletrônica analógica, eletrônica digital, eletrônica de potência, microcontroladores, sistemas microprocessados e redes de computadores.

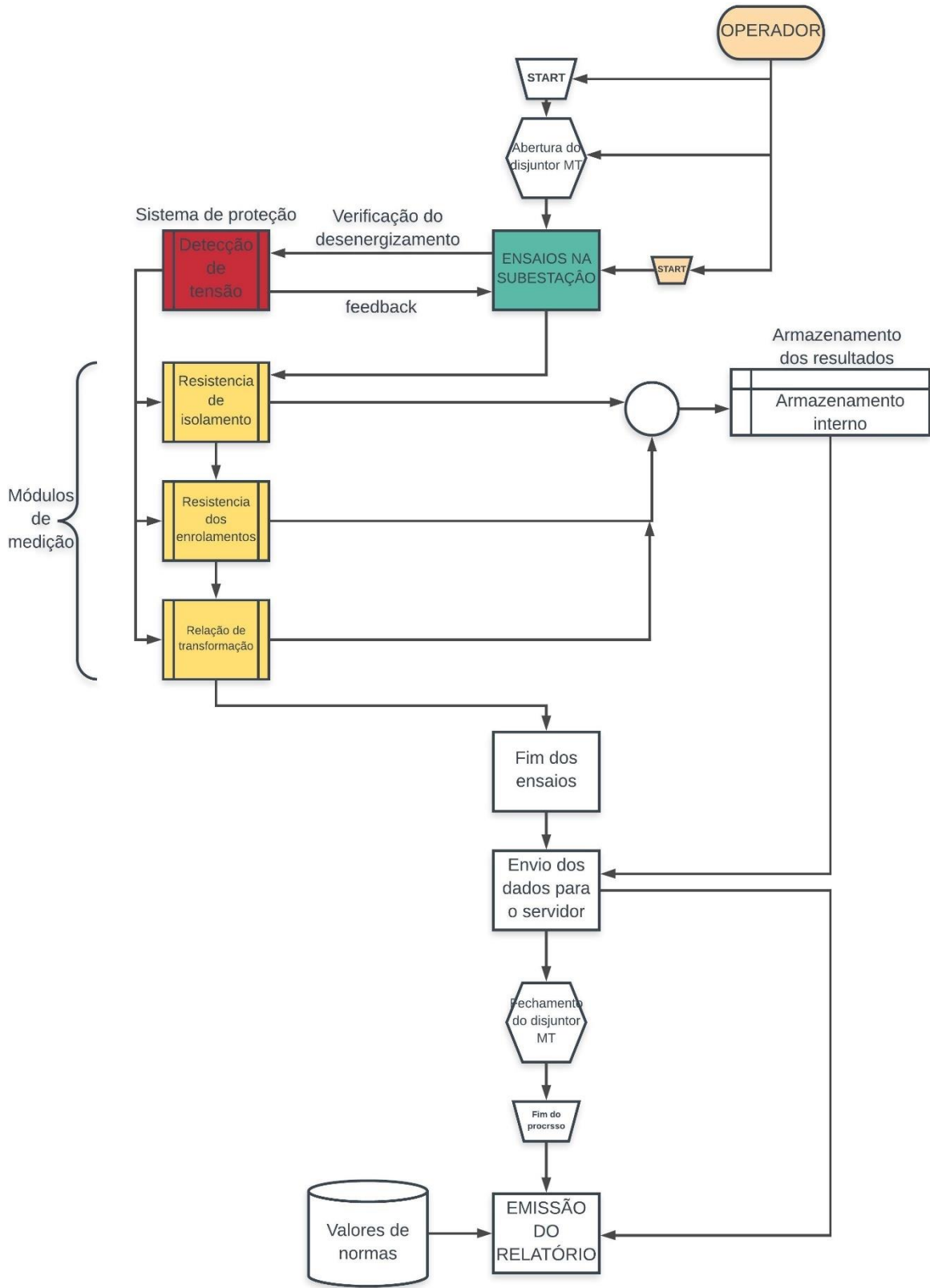
Em seguida, foi desenvolvido uma interface de operação e controle para o protótipo, esta interface é operada por um operador e tem a função de controlar o protótipo e emitir o relatório final com os dados coletados. Para desenvolver a interface, foi utilizado bibliografias voltadas para rede de computadores, linguagem de programação, normas regulamentadoras da ABTN e programação em microcontroladores.

O processo inicia-se com um comando do operador de início, um comando para desenergizar a subestação (abrir o disjuntor de média tensão) e um comando para iniciar os testes. Os ensaios são feitos um por vez para que um não interfira nos resultados dos demais, os valores são armazenados e quando finalizado os testes os resultados são enviados para o computador do operador e o sistema gera um relatório comparando com valores tabelados e em seguida emite para o usuário. Esse processo está ilustrado na Figura 8 como fluxograma do sistema de ensaio em subestação.

Como a subestação é responsável pela alimentação de energia elétrica do estabelecimento onde está sendo aplicado, o protótipo deve ser alimentado por bateria e o computador onde o operador controla os ensaios deve ter um nobreak com duração mínima suficiente para segurar a energia do computador no início que a subestação é desligada até sua reenergização.

O circuito deve controlar a abertura e o fechamento do disjuntor de média tensão, receber comandos de um usuário externo a subestação e realizar os ensaios de medição da resistência de isolamento do transformador, medição da relação de espiras e medição da resistência das espiras do transformador. Para isso foi considerado três módulos de medição, o megômetro, micromímetro e o *Transformer Turn Ratio* (TTR).

Figura 8: Fluxograma de operação do sistema de ensaios de transformadores



Fonte: Própria

## 2.1 DESENVOLVIMENTO DO PROTÓTIPO

O protótipo é um conjunto de componentes eletrônicos com a finalidade de realizar a leitura de dados dentro da subestação e apresentar os dados ao operador. Os componentes principais são: Um microcontrolador para processar os dados e realizar a comutação dos módulos de medição, os módulos de medição que têm a função de realizar medições específicas (resistência de isolamento, resistência dos enrolamentos, relação de transformação, etc.) e um sistema de comutação responsável por realizar a troca dos módulos de medição e é controlado pelo microcontrolador. O microcontrolador utilizado é um Arduino MEGA que tem o processamento adequado para esse tipo de atividade e a quantidade de I/O necessárias. O chaveamento é responsável por alternar entre os módulos de medição, são utilizados relé e transistor para o chaveamento. Para o módulo do megômetro que realiza e medição da resistência de isolamento deve-se utilizar um relé com isolamento de 10 kV e todo o sistema deve ter uma isolação adequada. Para os demais instrumentos, é utilizado relé de aplicação simples com apenas um contato. É importante observar que a leitura de dados no transformador é feita através de cabos e estes cabos estão conectados a todo momento, para isso, alguns critérios de isolação na parte de alta tensão devem ser seguidos. O projeto exige a utilização de barramentos auxiliares para o lado de alta tensão do transformador, este barramento deve ter isolação de no mínimo 15 kV. A isolação deve ser observada nos cabos que ligam os barramentos até o transformador e dos relés que iram fazer o chaveamento, ambos deve ter isolação superior a 15 kV. É recomendado que o relé tenha uma isolação acima de 36,4 kV para garantir uma confiabilidade maior.

O conjunto barramento e relé devem estar dentro do cubículo de transformação, dentro de um painel de estrutura de ferro e todo aterrado, com o cabo de aterramento com dimensão mínima de 50 milímetros quadrado, e o painel deve ter dimensões previstas em normas de instalação de média tensão. Os barramentos devem ter um espaçamento mínimo de 30 centímetros e não devem estar totalmente expostos para garantir a confiabilidade. Recomenda-se utilizar cabos com isolação de 15 kV como barramento, com terminais a compressão na saída dos cabos e estes ligados diretamente a entrada dos relés.

Além do Arduino mega, é utilizado um Arduino nano para realizar uma operação de segurança em paralelo aos ensaios. Essa operação fica verificando a presença de tensão nos barramentos principais da subestação para caso aconteça alguma falha na comunicação entre o controlador principal e o disjuntor de média tensão ou o controlador do disjuntor de média

tensão e este realiza o fechamento do circuito energizando o transformador. Este sistema de segurança tem como objetivo principal ficar acompanhando todo momento os barramentos da subestação e em caso de falha realizar a desconexão do protótipo nos barramentos do transformador. Outra função importante do sistema de proteção é realizar a comutação dos relés dos barramentos do transformador e abrir após finalizar a atividade. Ambos os projetos são constituídos basicamente por um microcontrolador e relé, sendo os relés ativados por transistor npn com polarização simples.

## 2.2 DESENVOLVIMENTO DA INTERFACE DE CONTROLE

A interface de controle tem sua função principal controlar e apresentar os resultados para um operador. Deve ser fácil de manusear e com poucos comandos, deve apresentar um sistema de proteção e bloqueio para que os ensaios na subestação aconteçam sem nenhum imprevisto. Um dos sistemas de proteção obrigatório associado a interface é a detecção de tensão que se encontra dentro da subestação e tem o papel de apontar para o operador que a subestação se encontra desenergizada e pronta para iniciar os ensaios. É importante que o status de “subestação desenergizada” esteja ativado para iniciar o processo. O desenergizamento da subestação acontece com a abertura do disjuntor de média tensão e é realizado pela interface, quando o operador seleciona a opção de abrir o disjuntor de média tensão. Outro ponto importante para o desenvolvimento deste projeto é que o disjuntor de média tensão deve ser motorizado e seu controle será associado à interface de controle de ensaio da subestação.

Como mostra no fluxograma (Figura 8), o operador necessitará dar início ao processo e acompanhar em um qualquer lugar, desde que o computador esteja ligado ao módulo de controle e ambos ligados à energia elétrica. A interface completa foi desenvolvida em Python e em dois códigos diferentes, cada código realizará uma função diferente. O primeiro, chamado de *screen*, tem a função da arte gráfica da interface e é ilustrado no Apêndice B. O segundo realiza o controle dos comandos, é chamado de *main* e o código completo é apresentado no Apêndice C.

O relatório final está ilustrado na Figura 9 e será preenchido pelo programa e emitido em forma de Excel para possibilitar salvar e imprimir. O relatório contém os três principais ensaios de rotinas realizados em um transformador, apresentando os valores e comentários quanto aos resultados, comparados com valores tabelados e estabelecidos por normas e uma breve descrição do transformador informando a potência, relação de transformação de tensão, número de série, frequência e o tipo de isolamento (à seco ou à óleo). As nomenclaturas de X0,

X1, X2 e X3 representam os terminais de baixa tensão do transformador e as nomenclaturas H1, H2 e H3 representam os terminais de média tensão. Na medição da resistência de isolamento as nomenclaturas MT representam os terminais de média tensão curto-circuitados, BT baixa tensão curto-circuitados e a Massa representa a carcaça do transformador.

Figura 9: Modelo de relatório de ensaio técnico

RELATÓRIO DE ENSAIOS TÉCNICOS						
Equipamento:	Transformador		Potência (kVA):		Tensões (kV~V):	13,8 ~ 127/220
Nº de série:	xxxxxx		Frequência (Hz):	60	Isolação/ Resfriamento:	
RESULTADO DOS TESTES						
Medição de resistência de enrolamento			Medição de resistência de isolação (MΩ)			
Enrolamentos de baixa tensão (mΩ)		Enrolamentos de média tensão (Ω)		Conexões	Resistência 1 minuto	Resistência 5 minutos
X0-X1		H1-H3		MT - BT		
X0-X2		H1-H2		MT - Massa		
X0-X3		H2-H3		BT - Massa		
Comentários			Comentários			
Medição de relação de espiras						
Fase	TAP	Relação medida	Relação padrão	Erro (%)	Comentários	
R-r	1					
S-s	1					
T-t	1					

Fonte: Própria

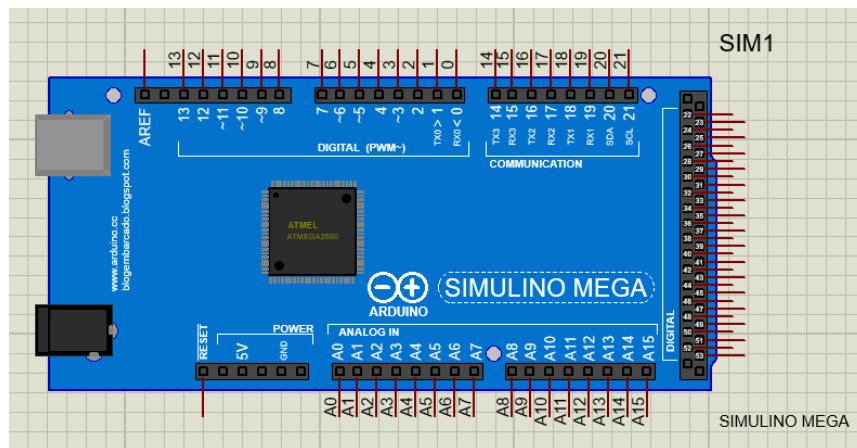
É utilizada a programação do código no *Visual Basic for Applications* (VBA) para receber os dados do Arduino e organizar no modelo ilustrado acima (Figura 9), o código realizado se encontra no Apêndice D. Os dados são enviados pelo Arduino por uma serial onde o Excel está conectado e o mesmo realiza a leitura dos dados inseridos na serial. A serial é um canal de comunicação utilizada pela plataforma Arduino e sem dúvidas a mais poderosa, pode-se utilizar vários tipos de meios de comunicação, como os meios *wireless* ou qualquer outro tipo de comunicação via cabo, e basta apenas acrescentar um módulo específico.

### 3 IMPLEMENTAÇÃO

#### 3.1 MONTAGEM DO PROTÓTIPO

A montagem do protótipo foi realizada no Proteus para simular o circuito e a implementação da programação do Arduino foi realizada pela IDE do próprio Arduino. Inicialmente foi escolhido o Arduino adequado para realizar a atividade. O Arduino escolhido foi o Arduino Mega pois será preciso utilizar 21 portas digitais de saída e entrada de dados. O Arduino Uno contém apenas 13 portas sendo duas específicas para comunicação serial.

Figura 10: Arduino MEGA simulado no Proteus

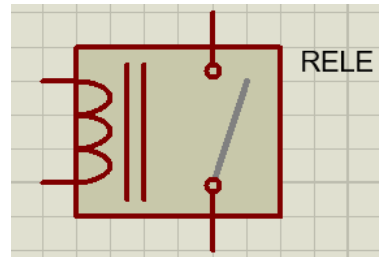


Fonte: Própria

Na Figura 10 é mostrado o Arduino mega na plataforma do Proteus e é possível verificar a semelhança com um Arduino físico. Na simulação, as portas de GND e fonte de tensão não são simulados, mas basta adicionar uma fonte de 5 V e o GND do Proteus.

Para realizar o chaveamento dos módulos de medição é utilizado relé. O simulador não apresenta muitos modelos de relé e nem o relé com isolamento de 15 kV. Para simulação, foi utilizado relé simples de um contato para todos os chaveamentos pois todos os relés têm o princípio de funcionamento idênticos, onde os contatos são acionados por uma bobina. O relé utilizado está ilustrado na Figura 11.

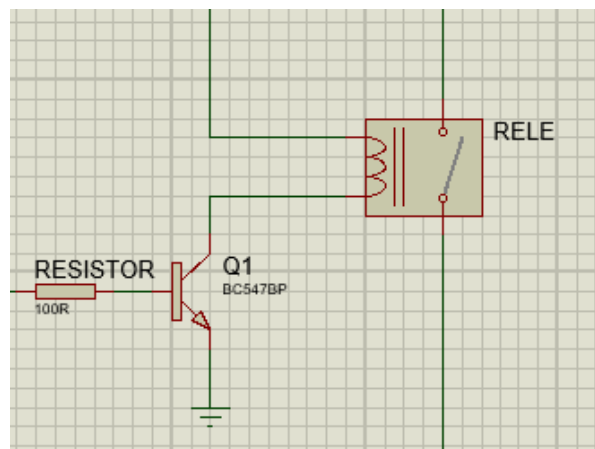
Figura 11: Representação do relé no Proteus



Fonte: Própria

O relé é acionado com 24 V em sua bobina, e para isso é utilizado um banco de bateria que forneça essa tensão e segure isso por um pouco mais de uma hora, tempo suficiente para realizar os ensaios e reenergizar a subestação. Para ativar as bobinas pelo Arduino é utilizado transistor bipolar npn associado a uma bateria. O transistor utilizado foi o BC547BP com polarização direta, ou seja, colocando um resistor para limitar a corrente na base do transistor e ligando a base direto com o terminal do Arduino como é ilustrado na Figura 12.

Figura 12: Polarização do transistor

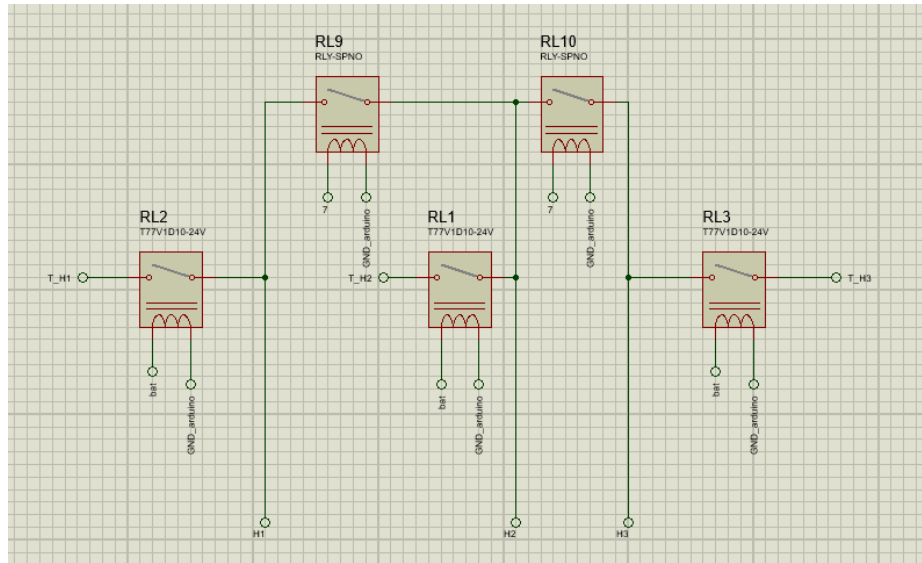


Fonte: Própria

Basicamente todo o circuito vai conter esse conjunto transistor e relé e, por isso, esse conjunto será chamado de Célula do circuito. Existirá uma célula dessa para cada barramento do transformador, células para realizar o curto-circuito dos terminais e células para realizar o chaveamento nos módulos de medição. Para realizar a comutação dos barramentos do transformador para o protótipo é utilizado o esquemático ilustrado na Figura 13, e cada relé é uma célula no circuito.



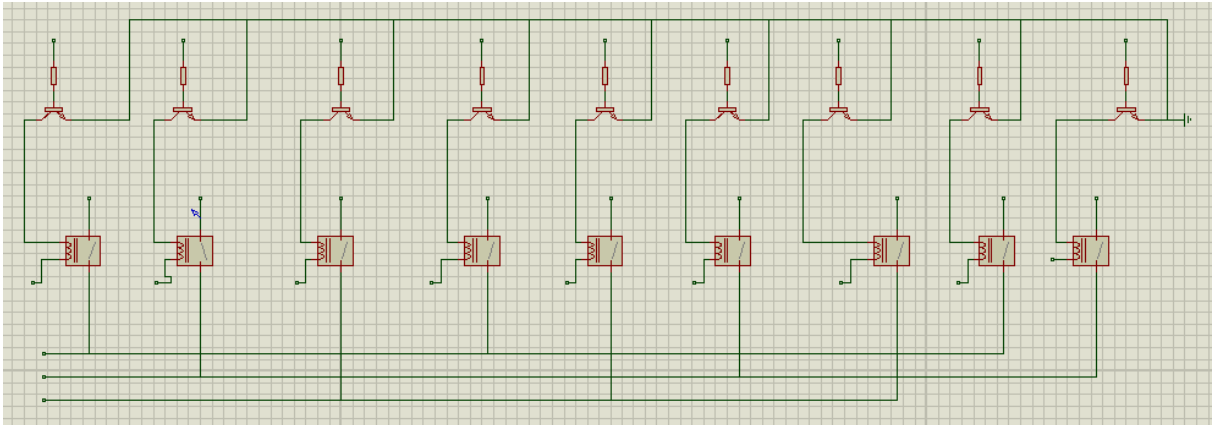
Figura 13: Esquema de comutação dos barramentos do transformador



Fonte: Própria

A parte de chaveamento dos módulos de medição funciona com o mesmo princípio da Figura 13, mudando apenas as ordens das células e aumentando sua quantidade. Por exemplo, o módulo de medição da resistência de isolamento vai precisar de nove células dessas para realizar o chaveamento no transformador e gerar resultados da resistência de isolamento entre as bobinas de média tensão e as bobinas de baixa tensão, entre as bobinas de média tensão e a carcaça do transformador e as bobinas de baixa tensão e a carcaça do transformador. A Figura 14 mostra o esquema de chaveamento para realizar as medições de resistência de isolamento. No Apêndice E são apresentados todos os esquemas de chaveamento de maneira mais otimizada. Pode-se observar que é possível realizar três tipos de ensaios diferentes utilizando apenas o módulo de medição megômetro e essa quantidade pode ser aumentada caso um cliente solicite outras medições na subestação. E para isso será necessário acoplar um conjunto de células para realizar as medições específicas e acrescentar no código do Arduino a função.

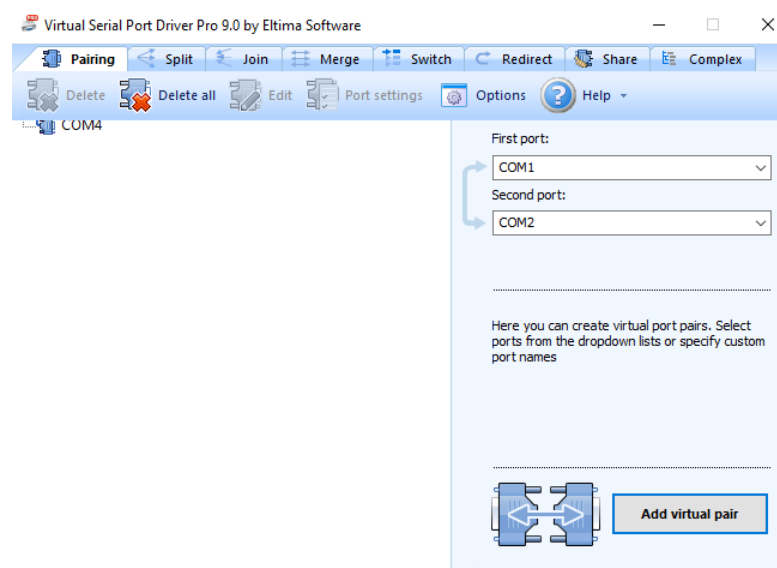
Figura 14: Esquema de chaveamento para controlar o módulo megômetro



Fonte: Própria

A última parte da implementação do projeto é a comunicação serial. Para isso foi necessário utilizar um componente denominado COMPIM. Este componente simula um meio de comunicação. Mas para realizar a comunicação virtual do Arduino com a interface de controles, foi necessário utilizar um simulador de comunicação, este simulador gera duas portas e realiza sua comunicação, sendo uma das portas direcionada ao Arduino e a outra a interface de controle. O programa utilizado para simular a comunicação é chamado de *Virtual Serial Port* e sua janela de comando está ilustrado na Figura 15 com uma comunicação entre a porta COM3 e a porta COM4.

Figura 15: Interface do *Virtual Serial Port*



Fonte: Própria

A COM3 foi direcionada para o Arduino deixando o outro lado da comunicação para a interface, ou seja, com a COM4. Após realizar a simulação da conexão, é possível conectar o Arduino do Proteus com a interface projetada no *Visual Studio Code* e com o Excel. Após finalizado toda a montagem do circuito, foi desenvolvido o código de controle das células na IDE do Arduino e por fim transferido para o mesmo. Os resultados obtidos durante um ensaio são armazenados na EEPROM do Arduino, possibilitando ter acesso em um outro momento. Outra vantagem de armazenar na EEPROM é a confiabilidade que o sistema gera, pois o microcontrolador pode ser desligado, mas os dados continuarão armazenados. O código para armazenamento dos dados na EEPROM do Arduino é ilustrado na Figura 16. Como o Arduino só consegue armazenar até 8 bits em cada endereço, ou seja, até o número 11111111 que equivale a 255 decimal, então, foi necessário realizar uma repartição do número em duas partes e salvo em posição de memória diferentes.

Figura 16: Linhas de código para armazenar dados na EEPROM do arduino

```
int inteiro = int(i11);  
int decimal = int((i11 - inteiro) * 100);  
EEPROM.write(0,inteiro);  
EEPROM.write(1,decimal);
```

Fonte: Própria

Para realizar o processo de leitura e escrita na EEPROM, deve-se adicionar a biblioteca EEPROM.h no código do programa. O comando de escrita solicita duas informações, a primeira é a posição da memória e a segunda é o valor que deve ser armazenado. Para acessar um valor na EEPROM do Arduino, é utilizado a função ilustrada na Figura 17 onde se deve informar apenas a posição na memória. O Apêndice A contém o programa completo com as funções comentadas.

Figura 17: Linhas de código para leitura da EEPROM do Arduino

```
Serial.print(EEPROM.read(22));  
Serial.print(".");  
Serial.println(EEPROM.read(23));
```

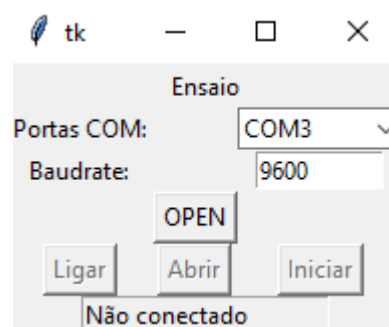
Fonte: Própria

### 3.2 MONTAGEM DA INTERFACE

A interface foi desenvolvida na plataforma *Visual Studio Code* utilizando a linguagem *Python* com um controle simples e fácil de manusear e é mostrado no Apêndice B e apêndice C. A comunicação com os demais subsistemas do projeto é por meio de serial, onde possibilita futuramente aplicação com qualquer meio de comunicação. Para iniciar a atividade, a interface, quando executada, solicita inicialmente a conexão com a serial. A interface já reconhece as portas conectadas e a respectiva velocidade de transmissão. Após o usuário escolher a porta que comunica com o Arduino do protótipo, o Arduino espera um sinal para sair do estado de segurança e este sinal é enviado pelo usuário quando pressiona o botão identificado como Ligar. O Arduino recebe o sinal e se prepara para começar o ensaio, mas antes deve acontecer o processo de desenergização do transformador, este processo ocorre abrindo o disjuntor de média tensão e por este motivo é recomendado que o disjuntor seja motorizado e controlado por um sistema remoto. O comando de comunicação para abrir o disjuntor de média tensão é o botão Abrir e este comando informa o controlador do disjuntor que deve realizar a abertura do mesmo.

Para iniciar os testes, o usuário deve selecionar o botão Iniciar, mas, por motivo de segurança, o botão se mantém inativo até que o controlador do disjuntor de média tensão informe que foi realizada a abertura e o detector de tensão presente dentro da subestação confirme o desenergizamento do transformador. A Figura 18 mostra a interface de controle simulada.

Figura 18: Interface de controle



Fonte: Própria

O código para criação da interface é dividido em duas partes. O primeiro é chamado de *Screen.py*, a qual contém toda a parte da arte gráfica da interface, onde é declarado os botões

com seus respectivos tamanhos, funções e posicionamento na área de trabalho. A Figura 19 mostra a parte do código que contém as declarações dos botões.

Figura 19: Linha de código para declaração dos botões

```
81 btnLigar = Button(Main, text='Ligar', command=main.ligarCmd, state = 'disable')
82 btnLigar.grid(row=6, column=1)
83 btnAbrir = Button(Main, text='Abrir', command=main.abrirCmd, state = 'disable')
84 btnAbrir.grid(row=6, column=2)
85 btnIniciar = Button(Main, text='Iniciar', command=main.iniciarCmd, state = 'disable')
86 btnIniciar.grid(row=6, column=3)
```

Fonte: Própria

A área de trabalho da interface contém os botões, o título da interface, o menu de portas de comunicação, menu da velocidade de comunicação e o status da conexão. A arte gráfica da área de trabalho da interface é declarado no programa com as linhas de comando mostrado na Figura 20. Os comandos *Label* são utilizados para escrever o texto e informar o posicionamento na área de trabalho. O posicionamento do texto é declarado pela função *.grid* e informado a posição da matriz em linha e coluna. Para o texto 'Interface de controle', foi utilizado a variável *row* (linha) igual a 2 e a variável *column* (coluna) igual a 1.

Figura 20: Linha de código para declaração da área de trabalho

```
34 Frame(Main, width = 150).grid(row=1, column=1, columnspan=3)
35 Label(Main, text='Interface de controle').grid(row=2, column=1, columnspan=3)
36 Label(Main, text="Portas COM:").grid(row=3, column=1)
37 cbCOM = StringVar(Main)
38 cbCOMs = ttk.Combobox(Main, textvar=cbCOM, width=10, state='readonly')
39 cbCOMs.grid(row=3, column=3)
40 Label(Main, text="Baudrate:").grid(row=4, column=1)
```

Fonte: Própria

A última parte do código contém a busca das portas de comunicação e apresentação da mesma para o usuário. Basta o usuário selecionar a porta que está sendo usada pelo Arduino e estabelecer a conexão. A Figura 21 mostra a parte do código que contém a busca da porta de comunicação e a apresentação da mesma. O número máximo de busca de portas de comunicação está limitado para 32, sendo que pode ser aumentado dependendo do nível de comunicação e da quantidade de equipamentos ligados a mesma rede. A busca é realizada por uma estrutura de repetição e verificando cada uma das portas, retornando apenas aquelas que estão abertas.

Figura 21: Linha de código para busca das portas serial

```

50 Entry(Main, textvar=enBaudrate, width=10, validate='key', validatecommand=(vcmd, '%P')).grid(row=4, column=3)
51 btOpen = Button(Main, text="OPEN", command=main.OpenCom)
52 btOpen.grid(row=5, column=2)
53
54 MAX_PORTAS = 32 # Número máximo de portas para verificação.
55 def getPortasEmUso () :
56     """
57     Método para buscar as portas em uso
58     """
59     # Faz um loop de 1 ao número máximo de portas para verificação e o concatena à string 'COM':
60     for p in ['COM%s' % (i + 1) for i in range(MAX_PORTAS)] :
61         try :
62             # Tenta abrir a conexão:
63             s = serial.Serial(p)
64             # Tenta fechar a conexão
65             s.close()
66             # Retorna um por um
67             yield s.name
68         except (OSError, serial.SerialException) :
69             # Caso der erro, a porta não está sendo utilizada. Ignoramos neste caso.
70             pass

```

Fonte: Própria

A segunda parte da programação é responsável pelos comandos e controle da interface, lá é declarado os status de cada botão inicialmente e o seu status após ser pressionado pelo usuário. Essa parte da programação é chamada de *main* e é considerado a parte de processamento, enquanto a parte *Screen* é mais voltada para a arte gráfica. No *main* é realizado os comandos de envio e recebimento de comandos para o Arduino e a interpretação dos comandos recebidos tanto pelo usuário como pelo Arduino. Também contém uma parte do sistema de proteção, onde o mesmo desabilita a função de iniciar os testes até que o sistema principal de segurança informe que a subestação está desligada. É importante informar que as duas partes da programação trabalham juntas, não havendo possibilidade de trabalhar com uma delas sem ter a outra. Quando for gerado um executável, a pasta de origem deve conter ambos os programas.

A primeira parte do código do *main* contém as funções do botão *open*, nesta parte é realizado a leitura da interface quanto ao status e, em caso de habilitado, a função realiza o comando de ligar a interface com a porta serial selecionada. Outra parte importante das funcionalidades da função *open* é a desativação dos demais botões. A Figura 22 mostra a linha de código do botão quando é ativado ou desativado ou quando não consegue estabelecer uma conexão com uma porta serial.

Figura 22: Linha de código com as funcionalidades do botão *open*

```

15 def OpenCom():
16     global isOpen
17     global s
18     try:
19         if (isOpen):
20             s.close()
21             isOpen = False
22             Screen.btOpen.config(text='OPEN')
23             Screen.edtStatus.set('Não conectado')
24             Screen.btnAbrir['state'] = 'disable'
25             Screen.btnLigar['state'] = 'disable'
26             Screen.btnIniciar['state'] = 'disable'
27         else :
28             s = serial.Serial(port=Screen.cbCOM.get(), baudrate=Screen.enBaudrate.get(), timeout=1000)
29             time.sleep(2)
30             isOpen = True
31             Screen.btOpen.config(text='CLOSE')
32             Screen.edtStatus.set('Conectado')
33             Screen.btnLigar['state'] = 'normal'
34     except:
35         Screen.PopUp('Porta não pode ser aberta')# + str(sys.exc_info()[0]))
36         isOpen = False
37         raise

```

Fonte: Própria

Logo após as linhas de código da tecla *open* vem as linhas de código para os botões Ligar, Abrir e Iniciar, ambas emitem sinais para o Arduino e esperam resposta para seguir para próxima etapa. A ordem de sequência de ativação dos botões é primeiro Ligar, depois Abrir e por último Iniciar. O botão seguinte sempre vai esperar um comando do botão anterior para ficar habilitado para ativação, e isso acontece para garantir a segurança e a confiabilidade do protótipo. O botão Ligar informa o Arduino que o mesmo deve se preparar para realizar os ensaios e o Arduino se retira de um status de segurança e espera confirmação do usuário, o próximo botão realiza a abertura do disjuntor de média tensão. É importante que o disjuntor esteja aberto para desligar o transformador da rede elétrica. Quando o transformador está desligado, o próximo passo é iniciar os testes, o Arduino fica aguardando o comando do usuário para iniciar, e este comando é gerado pelo botão Iniciar. A Figura 23 mostra as linhas de código dos botões, com a sequência de enviar o dado, habilitar o próximo botão e apresentar o status. Existe uma outra linha de código no final de cada conjunto dos botões que informa quando a conexão é perdida, essa linha estará monitorando a conexão e informará rapidamente ao usuário o seu status.

Figura 23: Linhas de código para os botões Ligar, Abrir e Iniciar

```

39 def ligarCmd():
40     if isOpen:
41         s.write(b'l')
42         Screen.btnAbrir['state'] = 'normal'
43         Screen.edtStatus.set('Sistema ligado')
44     else:
45         Screen.PopUp('Porta COM não aberta')
46
47 def abrirCmd():
48     if isOpen:
49         s.write(b'a')
50         Screen.btnIniciar['state'] = 'normal'
51         Screen.edtStatus.set('Sistema aberto')
52     else:
53         Screen.PopUp('Porta COM não aberta')
54
55 def iniciarCmd():
56     if isOpen:
57         s.write(b'i')
58         Screen.edtStatus.set('Sistema iniciado')
59         Screen.Main.update()
60         waitResposta()
61         Screen.edtStatus.set('Sistema finalizado')
62     else:
63         Screen.PopUp('Porta COM não aberta')

```

Fonte: Própria

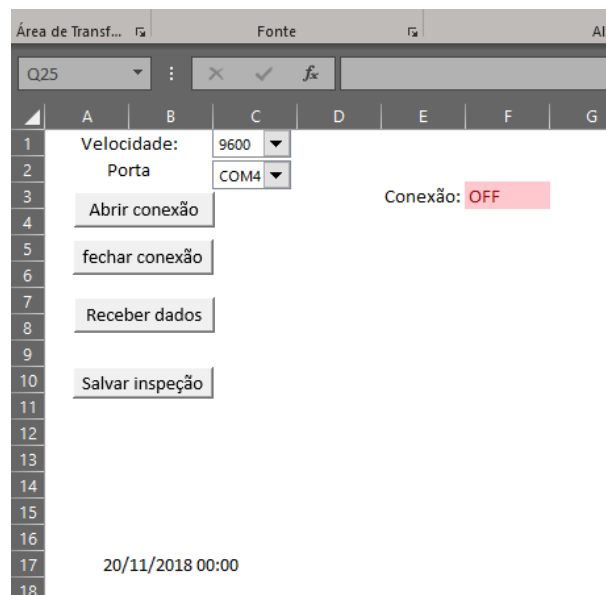
Com essa parte é finalizado a programação da interface de comunicação e controle entre o usuário e o Arduino. A próxima etapa é a geração do relatório. Quando o Arduino finaliza as medições, ele informa para a interface que os dados estão armazenados em sua memória e que já pode finalizar a manutenção. O usuário vai selecionar *ok* na interface e o Arduino vai mandar um sinal para o comando do disjuntor de média tensão que o mesmo já pode ser fechado e energizando a subestação. Após todo o processo, o Arduino faz a conferência de segurança e entra no modo de hibernação num *loop* de segurança aguardando ser chamado novamente.

A próxima etapa é configurar o Excel para receber os dados do Arduino, organizar e gerar um relatório final. Para organizar o relatório foi necessário conhecer a programação em *Visual Basic for Applications* e programação em Macro. Existe dois métodos para programar no Excel, a primeira é habilitando o modo 'desenvolvedor' e programar direto na ferramenta VBA e a outra é gravando Macros a partir de ações executadas por um usuário.



Para receber os dados do Arduino, o Excel deve realizar comunicação via serial, assim como a interface se conectou. Para isso, foi realizada a programação em VBA e criado macros para gravar ações e colocado em botões para a interação entre o usuário. Basicamente é necessário apenas 4 botões e dois seletores. Os seletores selecionam a porta de comunicação e a velocidade. Na Figura 24 é mostrado a interface de comando no Excel para realizar a coleta dos dados e gerar o relatório.

Figura 24: Interface de controle do Excel



Fonte: Própria

Além dos botões, a interface contém a data atual e o status da conexão com a serial. Quando é estabelecido uma conexão, o status passa de OFF para ON. Para iniciar a coleta de dados, primeiramente o usuário deve colocar a velocidade da conexão e a porta serial, em seguida, pressionar o botão “Abrir conexão” e aguardar a conexão ser estabelecida. Quando a conexão for estabelecida, o usuário deve pressionar o botão “Receber dados” e em seguida pressionar o botão “Salvar inspeção”. Um novo relatório é gerado numa planilha nova com os dados enviados da memória do Arduino e os resultados são comparados com valores estabelecidos por normas ou fabricantes.

O VBA divide as funções em grupos chamados de Sub e é onde é gravado as rotinas que o Excel deve realizar. A primeira rotina é para o botão de abrir conexão e este comando realiza a conexão entre o Excel e o Arduino. As linhas de código para gerar a função do botão Abrir conexão são descrita na Figura 25. As duas primeiras linhas buscam as informações da interface sobre a porta de comunicação e a velocidade de comunicação, as duas seguintes

configuram a comunicação e prepara o Excel para a conexão e a penúltima linha realiza a conexão com o Arduino. A função finaliza mandando uma informação para a interface de que a conexão foi realizada com sucesso. O outro botão é para fechar a conexão, a função criada contém apenas duas linhas de código, a primeira realiza o fechamento da conexão e a segunda informa para o usuário que a conexão foi finalizada. A Figura 26 mostra as linhas de código para desconectar o Excel com o Arduino.

Figura 25: Linhas de comando para gerar a conexão entre o Excel e o Arduino

```
com$ = UCase(Cells(2, 17))
baud$ = UCase(Val(Cells(3, 17)))
ReturnValue = Shell("mode.com " + com$ + " baud=" + baud$ + " parity=n data=8 stop=2 to=off xon=off dtr=off rts=off")
Application.Wait TimeSerial(Hour(Now()), Minute(Now()), Second(Now()) + 2)

Open com$ For Binary Access Read Write As #1
Sheets("Planilha1").Range("Q13") = "ON"
```

Fonte: Própria

Figura 26: Linhas de código para desativar a conexão do Excel com o Arduino

```
Close #1
Sheets("Planilha1").Range("Q13") = "OFF"
```

Fonte: Própria

O próximo comando é realizar a leitura dos dados do Arduino. Após o usuário pressionar o botão “Receber dados”, o Excel envia um sinal para o Arduino e em seguida o Arduino envia os dados pela serial de uma vez numa forma organizada e fácil de interpretação pelo Excel. Na Figura 27 é mostrado as linhas de código para leitura de dados. Como o Arduino manda todos os dados de uma vez, o Excel deve ler, separar e armazenar cada um deles. As primeiras linhas do código informam o Arduino que pode enviar os dados, realiza uma publicação no relatório o horário e o dia que recebeu os dados e em seguida começa a leitura. Os dados são lidos pela função Input do Excel e nesta função se deve informar o canal de conexão e colocar uma variável para receber os dados.

Figura 27: Linhas de código para receber os dados do Arduino

```

MyString$ = ""
MyString$ = "p"
Put #1, , MyString$

Sheets("Planilha2").Range("B6") = Now()
linha 1
MyString$ = ""

Do
    DoEvents
    char$ = Input(1, #1)
    If (char$ > Chr(31)) Then
        MyString$ = MyString$ + char$
    End If
Loop Until (char$ = Chr(44))
Sheets("Planilha2").Cells(10, 9) = Int(MyString)

```

Fonte: Própria

A última função gera o relatório de maneira organizada e salva a pasta completa para que o usuário possa realizar consultas posteriormente. Essa função apresenta três linhas, as duas primeiras são para gerar um novo relatório e a última para salvar os resultados. A Figura 28 mostra as linhas de código.

Figura 28: Linhas de código para gerar um novo relatório e salvar

```

Sub salvar_planilha()
'
' salvar_planilha Macro
    Sheets("Planilha2").Select
    Sheets("Planilha2").Copy After:=Sheets(2)

    ActiveWorkbook.Save
End Sub

```

Fonte: Própria

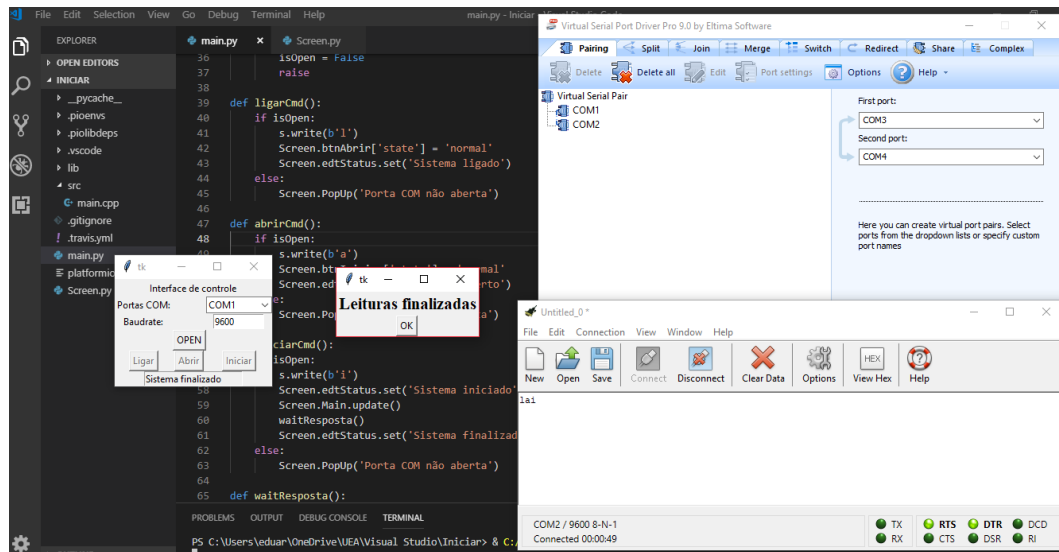
## 4 TESTES E RESULTADOS

### 4.1 TESTE

Para chegar em um resultado, foi necessário simular a atuação dos módulos de medição e simular resultados variados para comparação com os resultados gerados no relatório. Neste capítulo será apresentado, primeiramente, a simulação dos resultados realizada no Proteus e no *Visual Studio Code*. O capítulo apresentará três partes, a primeira é a simulação da interface e do protótipo, a segunda será a simulação da comunicação entre ambos e a última será a verificação dos resultados obtidos.

Iniciando com o teste da interface, foi necessário simular uma comunicação entre a interface e um simulador de serial. O simulador da serial utilizado foi o *CoolTermWin*, que tem a função de estabelecer uma conexão com portas seriais abertas. Para gerar as portas, foi utilizado o *Virtual Serial Port*, que é descrito na parte de implementação do protótipo. Com a conexão estabelecida, a minha interface está se comunicando o *CoolTermWin* e através deste aplicativo é possível verificar os dados enviados pela interface e, também, enviar dados através da serial. Os testes se iniciaram executando a interface, estabelecendo a comunicação e selecionando os comandos na ordem e verificando se os dados estavam chegando corretamente na outra ponta da comunicação. E, em algum momento, a interface esperava uma resposta da serial para realizar o próximo passo, a resposta era enviada pelo aplicativo e verificava-se se a interface receber corretamente. A comunicação é mostrada na Figura 29 com os aplicativos em operação para auxiliar. Com o fim do primeiro teste, a interface estava pronta para ser usada, e era preciso seguir para os próximos testes.

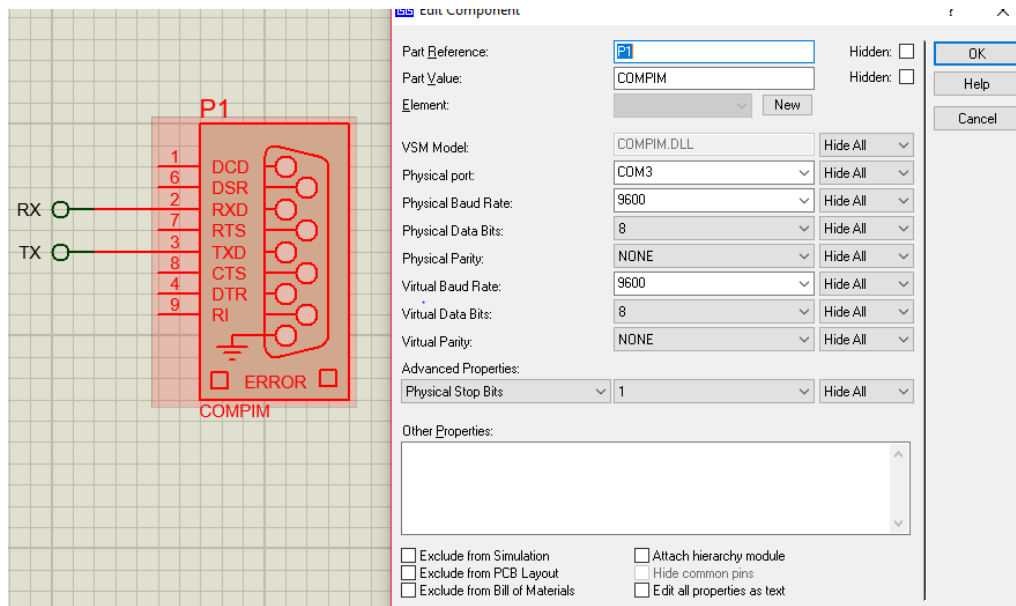
Figura 29: Simulação da comunicação serial da interface



Fonte: Própria

O teste seguinte consistia em verificar o funcionamento do circuito, para isso, foi projetado um circuito no Proteus, realizado a configuração e a programação do Arduino, e simulado uma conexão por serial virtual. A conexão do Arduino via Proteus se dá com a utilização de um componente onde é ilustrado na Figura 30. Na configuração do componente, denominado *COMPIM*, deve-se colocar a porta e a velocidade da transmissão. Para isso foram utilizados os mesmos programas anteriores, o *CoolTermWin* e o *Virtual Serial Porte*. Os testes consistiam do mesmo princípio dos testes da interface, e as respostas que o Arduino recebia deveria gerar uma modificação dos status do protótipo.

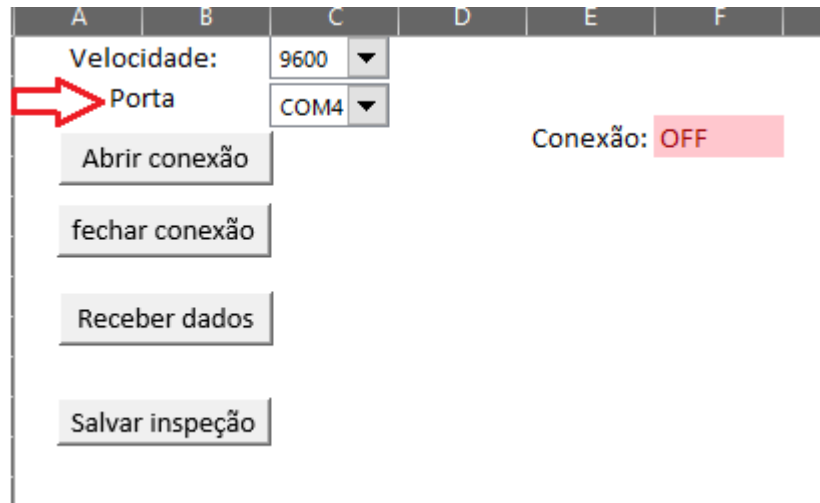
Figura 30: Componente que estabelece conexão do Arduino com a serial



Fonte: Própria

O último teste consistia em juntar a simulação do *hardware* com a simulação do *software* e para isso foi necessário realizar o mesmo processo dos testes anteriores. Como nos testes anteriores a interface e o protótipo individualmente responderam aos comandos corretamente, bastava apenas estabelecer a conexão entre eles e acompanhar o funcionamento. Para realizar a conexão, foi necessário gerar uma serial virtual, ligar o Arduino do Proteus em uma porta e a Interface no VSC na outra porta, por exemplo, foi gerado uma serial com as portas COM3 e COM4, o Arduino se conectava com a porta COM3 e a interface se comunicava através da porta COM4. A última parte do teste consistia em realizar a conexão do Excel com o Arduino e verificar os resultados. Para isso, foi necessário manter a serial aberta, mantendo o Arduino conectado na COM3, desativando a interface da COM4 e ligando o Excel na porta disponível, a COM4. O processo de conectar o Excel com uma serial é descrito na parte da Montagem da Interface e é ilustrado na Figura 31, onde o usuário deve selecionar a porta e em seguida pressionar o botão Abrir conexão

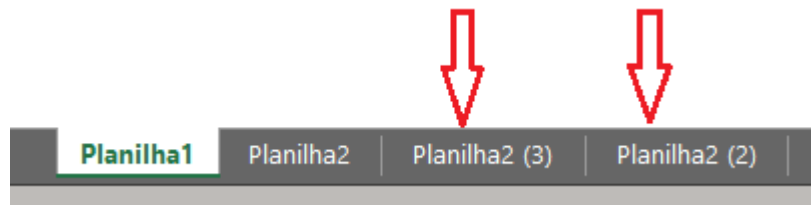
Figura 31: Conectando o Excel com a serial



Fonte: Própria

Após estabelecer a conexão do Arduino com o Excel, o usuário seleciona o botão Receber dados e por fim Salvar inspeção. O relatório é gerado numa nova planilha como é ilustrado na Figura 32, onde a Planilha1 é a interface principal, a Planilha2 é a base para montagem do relatório e a Planilha2 (2) ou Planilha2 (3) são os resultados obtidos. O

Figura 32: Geração de novos relatórios no Excel



Fonte: Própria

## 4.2 RESULTADOS

Os valores obtidos do Arduino são organizados e armazenados em uma planilha (ver Figura 33), possibilitando um usuário visualizar, editar, criar novos critérios, imprimir e copiar para colar em um outro padrão de relatório próprio.

Figura 33: Relatório final com os dados coletados do protótipo

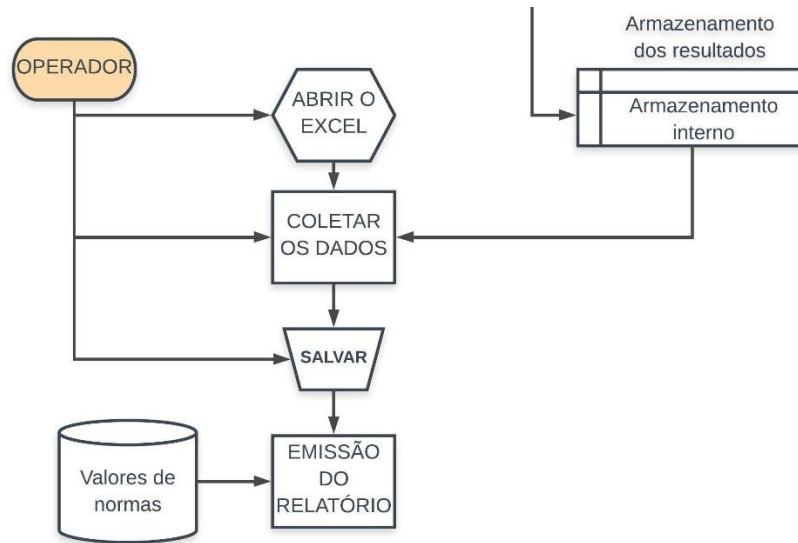
RELATÓRIO DE ENSAIOS TÉCNICOS							
<b>Equipamento:</b>	Transformador		<b>Potência (kVA):</b>	300	<b>Tensões (kV~V):</b>	13,8 ~ 127/220	
<b>N° de série:</b>	xxxxxx		<b>Frequência (Hz):</b>	60	<b>Isolação/ Resfriamento:</b>	Óleo	
RESULTADO DOS TESTES							
<b>Data:</b>	21/11/2018 16:13						
Medição de resistência de enrolamento				Medição de resistência de isolação (MΩ)			
Enrolamentos de baixa tensão (mΩ)		Enrolamentos de média tensão (Ω)		Conexões	Resistência	Resistência	Índice de absorção
					1 minuto	5 minutos	
X0-X1	2	H1-H3	10,1	MT - BT	100,00	120,00	1,2
X0-X2	2,4	H1-H2	10,19	MT - Massa	110,00	180,00	1,63636
X0-X3	2,5	H2-H3	10,3	BT - Massa	200,00	250,00	1,25
Comentários				Comentários			
Medição de relação de espiras							
Fase	TAP	Relação medida	Relação padrão	Erro (%)	Comentários		
R-r	1	108,69	108,664	-0,023926968	Valores dentro dos padrões		
S-s	1	108,8	108,664	-0,125156446	Valores dentro dos padrões		
T-t	1	108,3	108,664	0,334977545	Valores dentro dos padrões		

Fonte: Própria

O projeto final não seguiu completamente o fluxograma da Figura 8. Esse fluxograma pode ser dividido em três etapas, a primeira representa a preparação para iniciar os testes, a segunda é a parte dos ensaios e a terceira parte representa a finalização do processo com o envio dos dados para o operador. As duas primeiras partes do projeto seguem o modelo apresentado no fluxograma sem nenhuma alteração, a última parte tem uma alteração. O projeto deveria pegar os dados armazenados na memória do microcontrolador e enviar para interface e por fim montar os dados no Excel. Mas o projeto segue o fluxograma da Figura 34, onde representa a seguinte sequência: quando os ensaios são finalizados, o operador recebe uma informação que os ensaios foram executados com sucesso e finaliza a operação, o operador, em seguida, deve abrir a planilha de relatório, estabelecer uma conexão, extrair os dados do Arduino e, por fim, salvar os resultados.



Figura 34: Fluxograma da coleta de dados e montagem do relatório



Fonte: Própria

Foi realizado um levantamento financeiro para verificar a viabilidade do projeto. Neste estudo, foi considerado o valor comercial dos instrumentos de medição para representar os módulos de medição e esse valor pode ser ainda menor se for realizar a fabricação dos módulos. Para a construção dos módulos, deve seguir as normas de ensaios em transformadores. A Tabela 1 mostra o valor estimado do custo para construir um sistema de ensaios de transformadores, no item miscelânea é considerado todo o custo com materiais baratos e ferramentas, este valor é estimado.

Tabela 1: Custo para construir um sistema remoto de ensaios em transformadores

Descrição	QTD	Preço Uni.	Preço tot.
Arduino Mega	1	R\$ 89,90	R\$ 89,90
Megômetro	1	R\$ 7.790,00	R\$ 7.790,00
TTR	1	R\$ 14.615,00	R\$ 14.615,00
Microhmímetro	1	R\$ 11.110,00	R\$ 11.110,00
Relé com isolamento 15kV	37	R\$ 196,00	R\$ 7.252,00
Transistor bipolar	37	R\$ 6,05	R\$ 223,85
Resistores	100	R\$ 0,07	R\$ 7,00
Estanho para solda	1	R\$ 85,00	R\$ 85,00
Estação de solda	1	207	R\$ 207,00
Miscelânea	-	-	R\$ 7.000,00
Investimento:			R\$ 48.379,75

Fonte: Própria

Como uma manutenção preventiva numa subestação varia de R\$ 10.000,00 até 16.000,00, dependendo da quantidade de transformadores, da potência, e do tipo de subestação. A manutenção é obrigatória pelo menos uma vez ao ano, mas existe empresas mais rigorosas que aplicam essa manutenção de duas a três vezes. Considerando que uma empresa pague o valor mínimo e considerando que a mesma realize uma vez ao ano essa manutenção o retorno investido num sistema remoto é menor que cinco anos, como pode ser observado na Tabela 1.

## CONCLUSÃO

A elaboração do projeto visou construir um controlador automático dos ensaios de rotina em transformadores em subestação abrigada com o objetivo de tornar o processo mais seguro e mais rápido. Foi aplicado todo o conhecimento adquirido na faculdade e no estágio supervisionado.

Iniciou-se com uma breve apresentação de conceitos de subestação, manutenção, ensaios em transformador de energia elétrica e microcontrolador Arduino. Em seguida foi apresentado os métodos utilizados para desenvolver o projeto e por fim os testes e a simulação com obtenção dos resultados. A ideia central do projeto é construir um controlador para os ensaios para que o setor de manutenção de uma empresa realize serviços dentro da subestação sem a necessidade de pessoas dentro aumentando a segurança e a confiabilidade do sistema de energia elétrica da empresa.

Como o quesito de segurança é mais importante dentro de uma empresa, a realização de uma manutenção preventiva sem a presença de técnicos e eletricitas dentro da subestação dá um alívio para o setor de segurança do trabalho. O projeto necessitará apenas de um operador qualificado para controlar a interface e saber ler os dados emitidos no final. A principal diferença é a possibilidade de realizar os ensaios dentro de uma subestação sem a necessidade de um operador dentro.

Com os testes realizados com o circuito simulado Proteus, os comandos sendo realizados por uma interface criada em Python e o relatório sendo gerado no Excel, os resultados são positivos quanto ao funcionamento do protótipo e quanto a comunicação entre as interfaces. O resultado final é um relatório ilustrado na Figura 33 contendo todos os dados coletados pelo Arduino e enviado para uma planilha no Excel com os valores organizados e interpretados.

Como melhoria, realizar a geração do relatório no Excel através da interface de controle, sem a necessidade de abrir o Excel, realizar a conexão e coletar os dados do sistema. Em seguida, melhorar a interface para ter um controle mais amplo, possibilitando realizar um teste individualmente e apresentando os resultados na própria interface. Também a possibilidade de controlar os módulos de medição com mais precisão, variando os parâmetros para se obter outros resultados desejados.

É importante acrescentar ensaios exigidos pela norma de ensaios em subestação para atender todas as exigências, no trabalho atendeu apenas as exigências da norma regulamentadora para transformadores de potência e para garantir um sistema com uma confiabilidade maior, torna-se necessário a realização de todos os ensaios e os ensaios

recomendados por fabricantes. Entre esses ensaios, os mais importantes são os ensaios de análise de óleo para transformadores a óleo e resistência de aterramento que é um dos importantes.

Acrescentar um sistema de supervisão online para ter parâmetros de tensão, corrente, fator de potência e entre outros de maneira contínua. Ligar o sistema à uma rede de uma indústria, para que um supervisor acesse em qualquer lugar. Acrescentar mais módulos de chaveamento para que toda a subestação tenha seus ensaios garantida, ampliando para chaves seccionadoras, disjuntores de média tensão, muflas, isoladores, cabos e tudo mais.

Realizar estudos econômicos com mais aprofundamento, onde leva em conta a mão de obra dos técnicos e eletricitas, e a possibilidade de realizar mais ensaios no período de um ano e a possibilidade de não deixar os módulos de medição exclusivamente para a subestação, ou seja, os módulos serão acoplados ao sistemas apenas quando for realizar os testes dando a possibilidade de utiliza-los em outros ensaios dentro de uma indústria, tornando o sistema mais viável para o setor de manutenção.

## REFERÊNCIAS BIBLIOGRÁFICAS

AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA. **Regulação dos Serviços de Distribuição.** Disponível em: <<http://www.aneel.gov.br/regulacao-dos-servicos-de-distribuicao>>. Acesso em: 15 setembro 2018.

ALMEIDA, Tamires. **O que é manutenção industrial de máquinas e equipamentos?.** Disponível em: <<https://industria hoje.com.br/o-que-e-manutencao-industrial-de-maquinas-e-equipamentos>>. Acesso em: 25 outubro 2018.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 7070: **Amostragem de gases e óleo mineral isolantes de equipamentos elétricos e análise dos gases livres dissolvidos.** Rio de Janeiro. 2006.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 5462: **Confiabilidade e manutenibilidade.** Rio de Janeiro. 1994.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 14039: **Instalações elétricas de média tensão de 1,0 kV a 36,2 kV.** Rio de Janeiro. 2005.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 7274: **Interpretação da análise dos gases de transformadores em serviço.** Rio de Janeiro. 2002.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 15749: **Medição de resistência de aterramento e de potenciais na superfície do solo em sistemas de aterramento.** Rio de Janeiro. 2009.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 7036: **Recebimento, instalação e manutenção de transformadores de potência para distribuição, imersos em líquidos isolantes.** Rio de Janeiro. 1990.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 5460: **Sistema elétrico de potência.** Rio de Janeiro. 1992.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 5356 - 1: **Transformadores de potência – Parte 1: Generalidades.** Rio de Janeiro. 2010.

ARDUINO. **About Us.** Disponível em: <<https://www.arduino.cc/en/Main/AboutUs>>. Acesso em: 3 novembro 2018.

FERREIRA, Aurélio Buarque de Holanda. **Dicionário Aurélio Básico da Língua Portuguesa.** 8. ed. Rio de Janeiro, Editora Positivo, 2010.

CHAVIER, Luís Fernando. **Programação para Arduino – Primeiros Passos.** Disponível em: <<https://www.circuitar.com.br/tutoriais/programacao-para-arduino-primeiros-passos/>>. Acesso em: 29 outubro 2018.

DATALINK ENGENHARIA ELÉTRICA. **Análise de óleo transformador**. Disponível em: < <http://datalink.srv.br/servicos/analise-de-oleo-transformador/>>. Acesso em: 26 outubro 2018.

KARDEC, Alan; LAFRAIA, João. **Gestão estratégica e confiabilidade**. Rio de Janeiro: Qualitymark, ABRAMAN. 80 f., 2002.

MAMEDE FILHO, João. **Instalações Elétricas Industriais**. 9 ed. São Paulo: LTC, 2017, 976 p.

PAULINO, M. E. C. **Considerações sobre modelos de sistema integrado de manutenção e testes automatizados de proteção Elétrica**. Congresso Brasileiro de Manutenção – ABRAMAN, 2005.

PAULINO, M. E. C. **MANUTENÇÃO DE TRANSFORMADORES. Ensaio de resistência de isolamento e de rigidez dielétrica**. O Setor Elétrico, capítulo VII, 2014a.

\_\_\_\_\_. **MANUTENÇÃO DE TRANSFORMADORES. Ensaio de resistência ôhmica de enrolamentos e avaliação do comutador sob carga**. O Setor Elétrico, capítulo IV, 2014b.

\_\_\_\_\_. **MANUTENÇÃO DE TRANSFORMADORES. Polaridade e relação em transformadores de potência**. O Setor Elétrico, capítulo V, 2014c.

TECNOGERA. **SUBESTAÇÃO DE ENERGIA – O QUE É E COMO FUNCIONA?**. 2014. Disponível em: < <https://www.tecnogera.com.br/blog/o-que-e-e-como-funciona-uma-subestacao-de-energia/>>. Acesso em: 15 setembro 2018.

TELES, Jhonata. **Manutenção corretiva: O que é, quando fazer e como fazer**. Disponível em: < <https://engeteles.com.br/manutencao-corretiva/>>. Acesso em: 25 outubro 2018.

THOMSEN, Adilson. **O que é Arduino?**. 2014. Disponível em: < <https://www.filipeflop.com/blog/o-que-e-arduino/>>. Acesso em: 28 outubro 2018.

UNIVERSO ELÉTRICO. **O QUE SÃO SUBESTAÇÕES?**. 2016. Disponível em: < <https://universoeletrico.wordpress.com/2016/08/02/o-que-sao-subestacoes/>>. Acesso em: 15 setembro 2018.

## OBRAS CONSULTADAS

A GERADORA. **O QUE É UMA SUBESTAÇÃO?**. 2016. Disponível em: <<https://www.ageradora.com.br/o-que-e-uma-subestacao/>>. Acesso em: 15 setembro 2018.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 5410: **Instalações elétricas de baixa tensão**. Rio de Janeiro. 2005.

Companhia Energética de Minas Gerais. **Especificação geral para montagem de subestações convencionais**. 2007. 89f. Minas Gerais.

MUZY, Gustavo Luiz Castro de Oliveira. **Subestações Elétricas**. 2012. 122f. Monografia (Graduação em Engenharia Elétrica) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2012.

## APÊNDICE A – CÓDIGO FONTE DO PROJETO (ARDUINO)

```

#include <EEPROM.h>
// Detector de tensão
const int detector = 6;
// Disjuntor média tensão
const int abrirDJ = 8;

// Megometro
const int ativaMega = 7;
const int AltaBaixa = 3;
const int AltaMassa = 4;
const int BaixaMassa = 5;
const int LeituraMega = A0;
const int iniciarMega = 12;

// TTR
const int ativaTTR = 39;
const int relacaoH1X1 = 9;
const int relacaoH2X2 = 10;
const int relacaoH3X3 = 11;
const int LeituraTTR = A1;
const int iniciarTTR = 14;

// Micromímetro
const int ativaMicro = 22;
const int bobinaX0 = 23;
const int bobinaX1 = 25;
const int bobinaX2 = 27;
const int bobinaX3 = 29;
const int bobinaH1 = 31;
const int bobinaH2 = 33;
const int bobinaH3 = 35;
const int LeituraMicro = A2;
const int iniciarMicro = 24;

char comando;

void setup() {
  pinMode(detector, INPUT);
  pinMode(abrirDJ, OUTPUT);

  pinMode(ativaMega, OUTPUT);
  pinMode(AltaBaixa, OUTPUT);
  pinMode(AltaMassa, OUTPUT);
  pinMode(BaixaMassa, OUTPUT);
  pinMode(iniciarMega, OUTPUT);

  pinMode(ativaTTR, OUTPUT);
  pinMode(relacaoH1X1, OUTPUT);
  pinMode(relacaoH2X2, OUTPUT);
  pinMode(relacaoH3X3, OUTPUT);
  pinMode(iniciarTTR, OUTPUT);

```



```

pinMode(ativaMicro, OUTPUT);
pinMode(bobinaX0, OUTPUT);
pinMode(bobinaX1, OUTPUT);
pinMode(bobinaX2, OUTPUT);
pinMode(bobinaX3, OUTPUT);
pinMode(bobinaH1, OUTPUT);
pinMode(bobinaH2, OUTPUT);
pinMode(bobinaH3, OUTPUT);
pinMode(iniciarMicro, OUTPUT);

pinMode(30, OUTPUT);
Serial.begin(9600);

// posição de controle
EEPROM.write(31, 0);
EEPROM.write(32, 0);
}

float i11 = 0;
float i15 = 0;
float i21 = 0;
float i25 = 0;
float i31 = 0;
float i35 = 0;
float r1 = 0;
float r2 = 0;
float r3 = 0;
float r4 = 0;
float r5 = 0;
float r6 = 0;
float t1 = 0;
float t2 = 0;
float t3 = 0;

void loop() {
  if (Serial.available()>0)
  {
    comando = Serial.read();
    switch (comando)
    {
      case 'l':
        digitalWrite(LED_BUILTIN, HIGH);
        EEPROM.write(31, 1);
        break;
      case 'a':
        digitalWrite(abrirDJ, HIGH);
        break;
      case 'p':

        for (int i = 0; i<28; i = i+2){
          Serial.print(EEPROM.read(i));
          Serial.print(".");
          Serial.print(EEPROM.read(i+1));
          Serial.print(",");
        }
    }
  }
}

```

```

Serial.print(EEPROM.read(22));
Serial.print(".");
Serial.println(EEPROM.read(23));

break;
case 'i':
if(EEPROM.read(31) == 1 && EEPROM.read(32) == 1){

    // TESTE DE ISOLAÇÃO
    digitalWrite(ativaMega, HIGH); // Manda sinal para o módulo megômetro iniciar
    // Leitura 1
    digitalWrite(AltaBaixa, HIGH); // Alta para Baixa
    digitalWrite(iniciarMega, HIGH);
    delay(200); // Tempo de espera de 1 minuto
    //i11 = analogRead(LeituraMega);
    delay(200); //Tempo de espera de 5 minuto
    //i15 = analogRead(LeituraMega);
    digitalWrite(iniciarMega, LOW);
    //delay(60000);
    digitalWrite(AltaBaixa, LOW);

    // Leitura 2
    digitalWrite(AltaMassa, HIGH); // Alta para Massa
    digitalWrite(iniciarMega, HIGH);
    delay(200); // Tempo de espera de 1 minuto
    //i21 = analogRead(LeituraMega);
    delay(200); //Tempo de espera de 5 minuto
    //i25 = analogRead(LeituraMega);
    digitalWrite(iniciarMega, LOW);
    //delay(60000);
    digitalWrite(AltaMassa, LOW);

    // Leitura 3
    digitalWrite(BaixaMassa, HIGH); // Baixa para massa
    digitalWrite(iniciarMega, HIGH);
    delay(200); // Tempo de espera de 1 minuto
    //i31 = analogRead(LeituraMega);
    delay(200); //Tempo de espera de 5 minuto
    //i35 = analogRead(LeituraMega);
    digitalWrite(iniciarMega, LOW);
    //delay(60000);
    digitalWrite(BaixaMassa, LOW);
    digitalWrite(ativaMega, LOW); // Manda sinal para o módulo megômetro desativar

    // TESTE RELAÇÃO DAS ESPIRAS

    digitalWrite(ativaTTR, HIGH); // Manda sinal para o módulo TTR iniciar

    // Leitura 1
    digitalWrite(relacaoH1X1, HIGH); // Relação H1 - X1
    digitalWrite(iniciarTTR, HIGH); // manda o comando para iniciar os testes
    delay(200); // Tempo para realização do ensaio: 2 minutos
    //t1 = analogRead(LeituraTTR);

```

```

digitalWrite(iniciarTTR, LOW);
digitalWrite(relacaoH1X1, LOW);

// Leitura 2
digitalWrite(relacaoH2X2, HIGH); // Relação H2 - X2
digitalWrite(iniciarTTR, HIGH); // manda o comando para iniciar os testes
delay(200); // Tempo para realização do ensaio: 2 minutos
//t2 = analogRead(LeituraTTR);
digitalWrite(iniciarTTR, LOW);
digitalWrite(relacaoH2X2, LOW);

// Leitura 3
digitalWrite(relacaoH3X3, HIGH); // Relação H3 - X3
digitalWrite(iniciarTTR, HIGH); // manda o comando para iniciar os testes
delay(200); // Tempo para realização do ensaio: 2 minutos
//t3 = analogRead(LeituraTTR);
digitalWrite(iniciarTTR, LOW);
digitalWrite(relacaoH3X3, LOW);

digitalWrite(ativaTTR, LOW); // Manda sinal para o módulo TTR desativar

// TESTE RESISTENCIA DAS ESPIRAS

digitalWrite(ativaMicro, HIGH); // Manda sinal para o módulo Micromímetro iniciar

// Leitura 1
digitalWrite(bobinaX0, HIGH);
digitalWrite(bobinaX1, HIGH); // Medindo a resistência da bobina X1-X0
digitalWrite(iniciarMicro, HIGH); // manda o comando para iniciar os testes
delay(300); // Tempo para realização do ensaio: 1 minutos
//X1 = analogRead(LeituraMicro);
digitalWrite(iniciarMicro, LOW);
digitalWrite(bobinaX1, LOW);

// Leitura 2
digitalWrite(bobinaX2, HIGH); // Medindo a resistência da bobina X2-X0
digitalWrite(iniciarMicro, HIGH); // manda o comando para iniciar os testes
delay(300); // Tempo para realização do ensaio: 1 minutos
//X2 = analogRead(LeituraMicro);
digitalWrite(iniciarMicro, LOW);
digitalWrite(bobinaX1, LOW);

// Leitura 3
digitalWrite(bobinaX3, HIGH); // Medindo a resistência da bobina X3-X0
digitalWrite(iniciarMicro, HIGH); // manda o comando para iniciar os testes
delay(300); // Tempo para realização do ensaio: 1 minutos
//X3 = analogRead(LeituraMicro);
digitalWrite(iniciarMicro, LOW);
digitalWrite(bobinaX0, LOW);
digitalWrite(bobinaX3, LOW);

```

```

// Leitura 4
digitalWrite(bobinaH1, HIGH);
digitalWrite(bobinaH3, HIGH); // Medindo a resistência da bobina H1-H3
digitalWrite(iniciarMicro, HIGH); // manda o comando para iniciar os testes
delay(300); // Tempo para realização do ensaio: 1 minutos
//H1 = analogRead(LeituraMicro);
digitalWrite(iniciarMicro, LOW);
digitalWrite(bobinaH1, LOW);
digitalWrite(bobinaH3, LOW);

// Leitura 5
digitalWrite(bobinaH1, HIGH);
digitalWrite(bobinaH2, HIGH); // Medindo a resistência da bobina H2-H1
digitalWrite(iniciarMicro, HIGH); // manda o comando para iniciar os testes
delay(300); // Tempo para realização do ensaio: 1 minutos
//H2 = analogRead(LeituraMicro);
digitalWrite(iniciarMicro, LOW);
digitalWrite(bobinaH1, LOW);
digitalWrite(bobinaH2, LOW);

// Leitura 6
digitalWrite(bobinaH2, HIGH);
digitalWrite(bobinaH3, HIGH); // Medindo a resistência da bobina H3-H2
digitalWrite(iniciarMicro, HIGH); // manda o comando para iniciar os testes
delay(300); // Tempo para realização do ensaio: 1 minutos
//H3 = analogRead(LeituraMicro);
digitalWrite(iniciarMicro, LOW);
digitalWrite(bobinaH2, LOW);
digitalWrite(bobinaH3, LOW);

digitalWrite(ativaMicro, LOW); // Manda sinal para o módulo Micromímetro iniciar

i11 = 100;
i15 = 120;
i21 = 110;
i25 = 180;
i31 = 200;
i35 = 250;
r1 = 2.3;
r2 = 2.4;
r3 = 2.5;
r4 = 10.1;
r5 = 10.2;
r6 = 10.3;
t1 = 108.7;
t2 = 108.8;
t3 = 108.9;

int inteiro = int(i11);
int decimal = int((i11 - inteiro) * 100);
EEPROM.write(0,inteiro);
EEPROM.write(1,decimal);

```

```
inteiro = int(i15);  
decimal = int((i15 - inteiro) * 100);  
EEPROM.write(2,inteiro);  
EEPROM.write(3,decimal);
```

```
inteiro = int(i21);  
decimal = int((i21 - inteiro) * 100);  
EEPROM.write(4,inteiro);  
EEPROM.write(5,decimal);
```

```
inteiro = int(i25);  
decimal = int((i25 - inteiro) * 100);  
EEPROM.write(6,inteiro);  
EEPROM.write(7,decimal);
```

```
inteiro = int(i31);  
decimal = int((i31 - inteiro) * 100);  
EEPROM.write(8,inteiro);  
EEPROM.write(9,decimal);
```

```
inteiro = int(i35);  
decimal = int((i35 - inteiro) * 100);  
EEPROM.write(10,inteiro);  
EEPROM.write(11,decimal);
```

```
inteiro = int(r1);  
decimal = int((r1 - inteiro) * 100);  
EEPROM.write(12,inteiro);  
EEPROM.write(13,decimal);
```

```
inteiro = int(r2);  
decimal = int((r2 - inteiro) * 100);  
EEPROM.write(14,inteiro);  
EEPROM.write(15,decimal);
```

```
inteiro = int(r3);  
decimal = int((r3 - inteiro) * 100);  
EEPROM.write(16,inteiro);  
EEPROM.write(17,decimal);
```

```
inteiro = int(r4);  
decimal = int((r4 - inteiro) * 100);  
EEPROM.write(18,inteiro);  
EEPROM.write(19,decimal);
```

```
inteiro = int(r5);  
decimal = int((r5 - inteiro) * 100);  
EEPROM.write(20,inteiro);  
EEPROM.write(21,decimal);
```

```
inteiro = int(r6);  
decimal = int((r6 - inteiro) * 100);  
EEPROM.write(22,inteiro);  
EEPROM.write(23,decimal);
```

```
inteiro = int(t1);
decimal = int((t1 - inteiro) * 100);
EEPROM.write(24,inteiro);
EEPROM.write(25,decimal);

inteiro = int(t2);
decimal = int((t2 - inteiro) * 100);
EEPROM.write(26,inteiro);
EEPROM.write(27,decimal);

inteiro = int(t3);
decimal = int((t3 - inteiro) * 100);
EEPROM.write(28,inteiro);
EEPROM.write(29,decimal);

for(int i = 0; i < 6; i++){
    digitalWrite(LED_BUILTIN, LOW);
    delay(20);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(20);
}
Serial.println("f");
EEPROM.write(31, 0);
digitalWrite(abrirDJ, LOW);
}
break;
}
}
if (digitalRead(detector) == HIGH){
    EEPROM.write(32, 1);
}
if (digitalRead(detector) == LOW){
    EEPROM.write(32, 0);
}
}
```

## APÊNDICE B – CÓDIGO DA INTERFACE (SCREEN)

```

from tkinter import *
from tkinter import ttk
import serial
import main
Main = Tk()

class TWaitUp:
    def __init__(self, root):
        self.root = root
    def Show(self, message):
        self.Wait = Toplevel(self.root)
        x = int(self.root.winfo_screenwidth()/2.5)
        y = int(self.root.winfo_screenheight()/2.5)
        self.Wait.geometry('+{}+{}'.format(x, y))
        Label(self.Wait, text=message, font='Times 16 bold').pack()
        self.Wait.grab_set()
        self.Wait.update()

    def Close(self):
        self.Wait.destroy()

WaitUp = TWaitUp(Main)

def PopUp(message):
    Pop = Toplevel(Main)
    Label(Pop, text=message, font='Times 16 bold').grid(row = 1, column = 1, columnspan=3)
    Button(Pop, text='OK', command=Pop.destroy).grid(row=2, column=2)
    x = int(Main.winfo_screenwidth()/2.5)
    y = int(Main.winfo_screenheight()/2.5)
    Pop.geometry('+{}+{}'.format(x, y))
    Pop.grab_set()
    Pop.update()

Frame(Main, width = 150).grid(row=1, column=1, columnspan=3)
Label(Main, text='Interface de controle').grid(row=2, column=1, columnspan=3)
Label(Main, text="Portas COM:").grid(row=3, column=1)
cbCOM = StringVar(Main)
cbCOMs = ttk.Combobox(Main, textvar=cbCOM, width=10, state='readonly')
cbCOMs.grid(row=3, column=3)
Label(Main, text="Baudrate:").grid(row=4, column=1)

def callback(P):
    if str.isdigit(P) or P == "":
        return True
    else:
        return False
vcmd = (Main.register(callback))
enBaudrate = StringVar(Main)

```

```
Entry(Main, textvar=enBaudrate, width=10,validate='key', validatecommand=(vcmd,
'%P')).grid(row=4, column=3)
btOpen = Button(Main, text="OPEN", command=main.OpenCom)
btOpen.grid(row=5, column=2)
```

```
MAX_PORTAS = 32 # Número máximo de portas para verificação.
def getPortasEmUso () :
```

```
    """
```

```
    Método para buscar as portas em uso
```

```
    """
```

```
    # Faz um loop de 1 ao número maximo de portas para verificação e o concatena à string
'COM':
```

```
    for p in ['COM%s' % (i + 1) for i in range(MAX_PORTAS)] :
```

```
        try :
```

```
            # Tenta abrir a conexão:
```

```
            s = serial.Serial(p)
```

```
            # Tenta fechar a conexão
```

```
            s.close()
```

```
            # Retorna um por um
```

```
            yield s.name
```

```
        except (OSError, serial.SerialException) :
```

```
            # Caso der erro, a porta não está sendo utilizada. Ignoramos neste caso.
```

```
            pass
```

```
l = []
```

```
for i in getPortasEmUso():
```

```
    l.append(i)
```

```
cbCOMs.config(values=l)
```

```
if len(l) != 0:
```

```
    cbCOMs.current(0)
```

```
enBaudrate.set(9600)
```

```
btnLigar = Button(Main, text='Ligar', command=main.ligarCmd, state = 'disable')
```

```
btnLigar.grid(row=6, column=1)
```

```
btnAbrir = Button(Main, text='Abrir', command=main.abrirCmd, state = 'disable')
```

```
btnAbrir.grid(row=6, column=2)
```

```
btnIniciar = Button(Main, text='Iniciar', command=main.iniciarCmd, state = 'disable')
```

```
btnIniciar.grid(row=6, column=3)
```

```
edtStatus = StringVar(Main)
```

```
Entry(Main, textvar=edtStatus, state='readonly').grid(row=7, column=1, columnspan=3)
```

```
edtStatus.set('Não conectado')
```

```
Main.mainloop()
```



## APÊNDICE C – CÓDIGO DA INTERFACE (MAIN)

```

from tkinter import *
from tkinter import ttk
from tkinter import messagebox
from Screen import *
import Screen
import sys
import serial
import time

isOpen = False
s = serial.Serial

def OpenCom():
    global isOpen
    global s
    try:
        if (isOpen):
            s.close()
            isOpen = False
            Screen.btOpen.config(text='OPEN')
            Screen.edtStatus.set('Não conectado')
            Screen.btnAbrir['state'] = 'disable'
            Screen.btnLigar['state'] = 'disable'
            Screen.btnIniciar['state'] = 'disable'
        else :
            s = serial.Serial(port=Screen.cbCOM.get(),    baudrate=Screen.enBaudrate.get(),
            timeout=1000)
            time.sleep(2)
            isOpen = True
            Screen.btOpen.config(text='CLOSE')
            Screen.edtStatus.set('Conectado')
            Screen.btnLigar['state'] = 'normal'
    except:
        Screen.PopUp('Porta não pode ser aberta')# + str(sys.exc_info()[0]))
        isOpen = False
        raise

def ligarCmd():
    if isOpen:
        s.write(b'1')
        Screen.btnAbrir['state'] = 'normal'
        Screen.edtStatus.set('Sistema ligado')
    else:
        Screen.PopUp('Porta COM não aberta')

def abrirCmd():

```

```
if isOpen:
    s.write(b'a')
    Screen.btnIniciar['state'] = 'normal'
    Screen.edtStatus.set('Sistema aberto')
else:
    Screen.PopUp('Porta COM não aberta')

def iniciarCmd():
    if isOpen:
        s.write(b'i')
        Screen.edtStatus.set('Sistema iniciado')
        Screen.Main.update()
        waitResposta()
        Screen.edtStatus.set('Sistema finalizado')
    else:
        Screen.PopUp('Porta COM não aberta')

def waitResposta():
    global s

    Screen.WaitUp.Show('Fazendo Leituras')
    while(s.in_waiting == 0):
        Screen.Main.update()
    c = s.read()
    Screen.WaitUp.Close()
    Screen.PopUp('Leituras finalizadas')

OpenCom()
```

## APÊNDICE D – CÓDIGO FONTE PARA GERAÇÃO DO RELATÓRIO NO EXCEL

```
Sub Botão1_Clique()
```

```
    com$ = UCase(Cells(2, 17))
    baud$ = UCase(Val(Cells(3, 17)))
    ReturnValue = Shell("mode.com " + com$ + " baud=" + baud$ + " parity=n data=8 stop=2
to=off xon=off dtr=off rts=off")
    Application.Wait TimeSerial(Hour(Now()), Minute(Now()), Second(Now()) + 2)

    Open com$ For Binary Access Read Write As #1
    Sheets("Planilha1").Range("Q13") = "ON"
End Sub
```

```
Sub Botão2_Clique()
```

```
    Close #1
    Sheets("Planilha1").Range("Q13") = "OFF"
End Sub
```

```
Sub Botão3_Clique()
```

```
    Put #1, , "A"
End Sub
```

```
Public Sub bargraf()
```

```
    MyString$ = ""
    MyString$ = "p"
    Put #1, , MyString$
```

```
    Sheets("Planilha2").Range("B6") = Now()
' linha 1
    MyString$ = ""
```

```
    Do
        DoEvents
        char$ = Input(1, #1)
        If (char$ > Chr(31)) Then
            MyString$ = MyString$ + char$
        End If
```

```
    Loop Until (char$ = Chr(44))
    Sheets("Planilha2").Cells(10, 9) = Int(MyString) / 10
```

```
    MyString$ = ""
    Do
        DoEvents
        char$ = Input(1, #1)
        If (char$ > Chr(31)) Then
            MyString$ = MyString$ + char$
        End If
    Loop Until (char$ = Chr(44))
```

```
Sheets("Planilha2").Cells(10, 11) = Int(MyString) / 10
```

```
' linha 2
```

```
DoEvents
```

```
MyString$ = ""
```

```
Do
```

```
DoEvents
```

```
    char$ = Input(1, #1)
```

```
    If (char$ > Chr(31)) Then
```

```
        MyString$ = MyString$ + char$
```

```
    End If
```

```
Loop Until (char$ = Chr(44))
```

```
Sheets("Planilha2").Cells(11, 9) = Int(MyString) / 10
```

```
DoEvents
```

```
MyString$ = ""
```

```
Do
```

```
DoEvents
```

```
    char$ = Input(1, #1)
```

```
    If (char$ > Chr(31)) Then
```

```
        MyString$ = MyString$ + char$
```

```
    End If
```

```
Loop Until (char$ = Chr(44))
```

```
heets("Planilha2").Cells(11, 11) = Int(MyString) / 10
```

```
' linha 3
```

```
DoEvents
```

```
MyString$ = ""
```

```
Do
```

```
DoEvents
```

```
    char$ = Input(1, #1)
```

```
    If (char$ > Chr(31)) Then
```

```
        MyString$ = MyString$ + char$
```

```
    End If
```

```
Loop Until (char$ = Chr(44))
```

```
Sheets("Planilha2").Cells(12, 9) = Int(MyString) / 10
```

```
DoEvents
```

```
MyString$ = ""
```

```
Do
```

```
DoEvents
```

```
    char$ = Input(1, #1)
```

```
    If (char$ > Chr(31)) Then
```

```
        MyString$ = MyString$ + char$
```

```
    End If
```

```
Loop Until (char$ = Chr(44))
```

```
Sheets("Planilha2").Cells(12, 11) = Int(MyString) / 10
```

```
' linha 4
```

```
DoEvents
```

```

MyString$ = ""
Do
DoEvents
  char$ = Input(1, #1)
  If (char$ > Chr(31)) Then
    MyString$ = MyString$ + char$
  End If
Loop Until (char$ = Chr(44))
Sheets("Planilha2").Cells(10, 2) = Int(MyString) / 100

```

' linha 5

```

DoEvents
MyString$ = ""
Do
DoEvents
  char$ = Input(1, #1)
  If (char$ > Chr(31)) Then
    MyString$ = MyString$ + char$
  End If
Loop Until (char$ = Chr(44))
Sheets("Planilha2").Cells(11, 2) = Int(MyString) / 100

```

' linha 6

```

DoEvents
MyString$ = ""
Do
DoEvents
  char$ = Input(1, #1)
  If (char$ > Chr(31)) Then
    MyString$ = MyString$ + char$
  End If
Loop Until (char$ = Chr(44))
Sheets("Planilha2").Cells(12, 2) = Int(MyString) / 100

```

' linha 7

```

DoEvents
MyString$ = ""
Do
DoEvents
  char$ = Input(1, #1)
  If (char$ > Chr(31)) Then
    MyString$ = MyString$ + char$
  End If
Loop Until (char$ = Chr(44))
Sheets("Planilha2").Cells(10, 5) = Int(MyString) / 100

```

' linha 8

```

DoEvents
MyString$ = ""

```

```

Do
DoEvents
  char$ = Input(1, #1)
  If (char$ > Chr(31)) Then
    MyString$ = MyString$ + char$
  End If
Loop Until (char$ = Chr(44))
Sheets("Planilha2").Cells(11, 5) = Int(MyString) / 100

```

' linha 9

```

DoEvents
MyString$ = ""
Do
DoEvents
  char$ = Input(1, #1)
  If (char$ > Chr(31)) Then
    MyString$ = MyString$ + char$
  End If
Loop Until (char$ = Chr(44))
Sheets("Planilha2").Cells(12, 5) = Int(MyString) / 100

```

' linha 10

```

DoEvents
MyString$ = ""
Do
DoEvents
  char$ = Input(1, #1)
  If (char$ > Chr(31)) Then
    MyString$ = MyString$ + char$
  End If
Loop Until (char$ = Chr(44))
Sheets("Planilha2").Cells(19, 3) = Int(MyString) / 100

```

' linha 11

```

DoEvents
MyString$ = ""
Do
DoEvents
  char$ = Input(1, #1)
  If (char$ > Chr(31)) Then
    MyString$ = MyString$ + char$
  End If
Loop Until (char$ = Chr(44))

```

```

Sheets("Planilha2").Cells(20, 3) = Int(MyString) / 100

```

' linha 12

```

DoEvents

MyString$ = ""
Do

```

```
DoEvents
  char$ = Input(1, #1)
  If (char$ > Chr(31)) Then
    MyString$ = MyString$ + char$
  End If
Loop Until (char$ = Chr(13))

Sheets("Planilha2").Cells(21, 3) = Int(MyString) / 100
MyString$ = ""
MyString$ = "f" + Chr(13)
Put #1, , MyString$

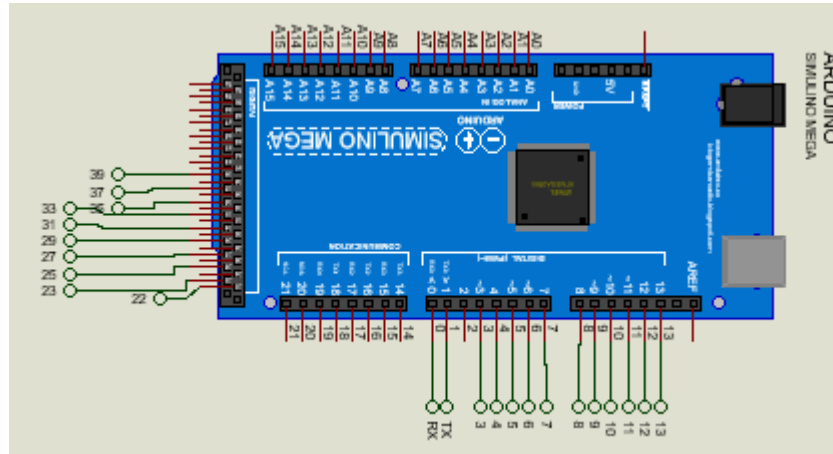
End Sub

Sub Botão4_Clique()
  Put #1, , "B"
End Sub

Sub salvar_planilha()
'salvar_planilha Macro
  Sheets("Planilha2").Select
  Sheets("Planilha2").Copy After:=Sheets(2)
'Sheets("Planilha2 (2)").Select
'Sheets("Planilha2 (2)").Name = ""
  ActiveWorkbook.Save
End Sub
```

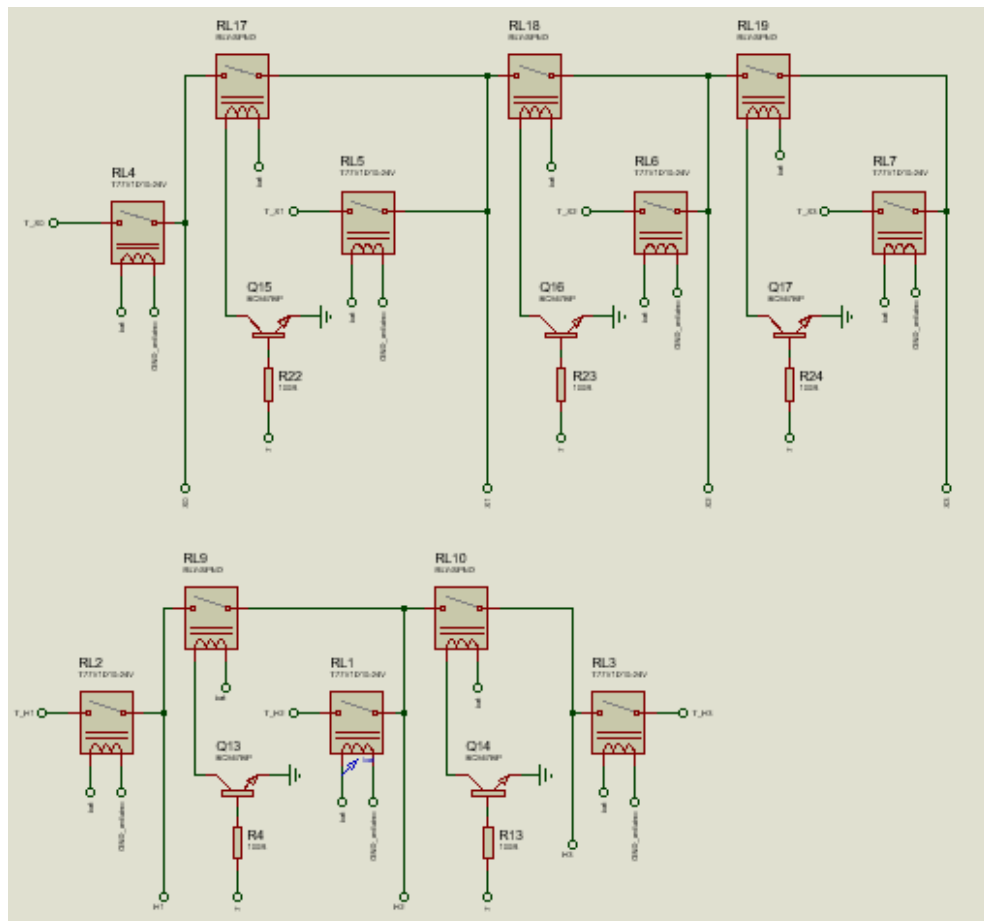
## APÊNDICE E – PROTÓTIPO MONTADO NO PROTEUS

Figura E-1: Controlador principal do sistema



Fonte: Própria

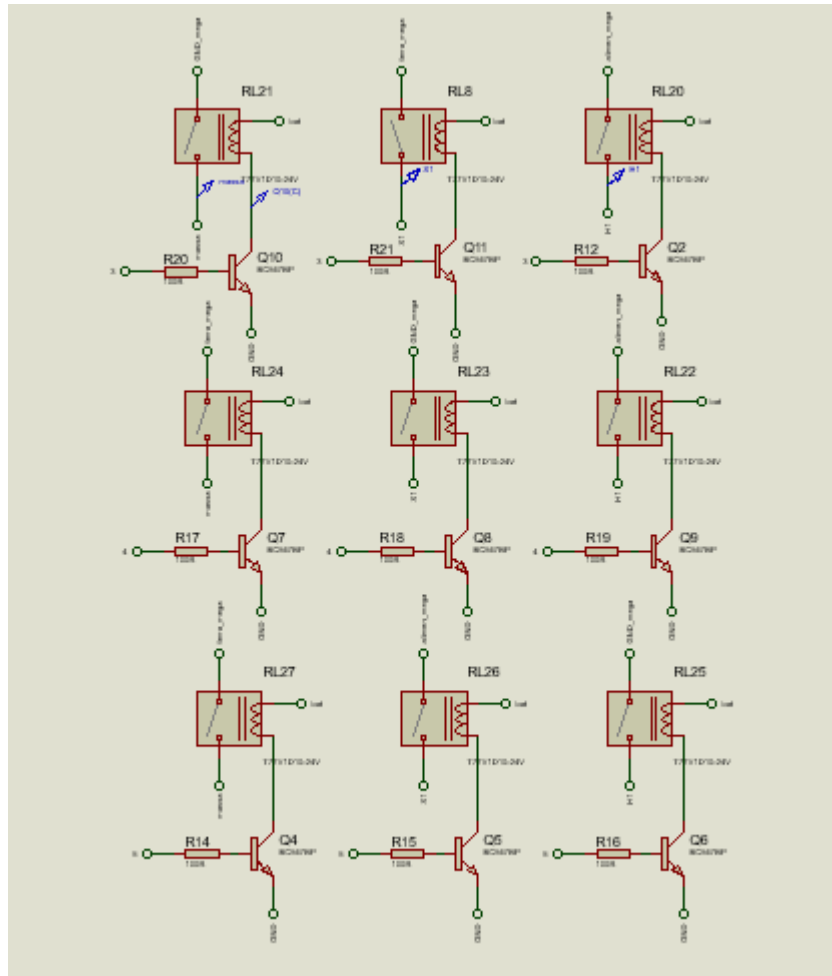
Figura E-2: Chaveamento dos barramentos do transformador



Fonte: Própria

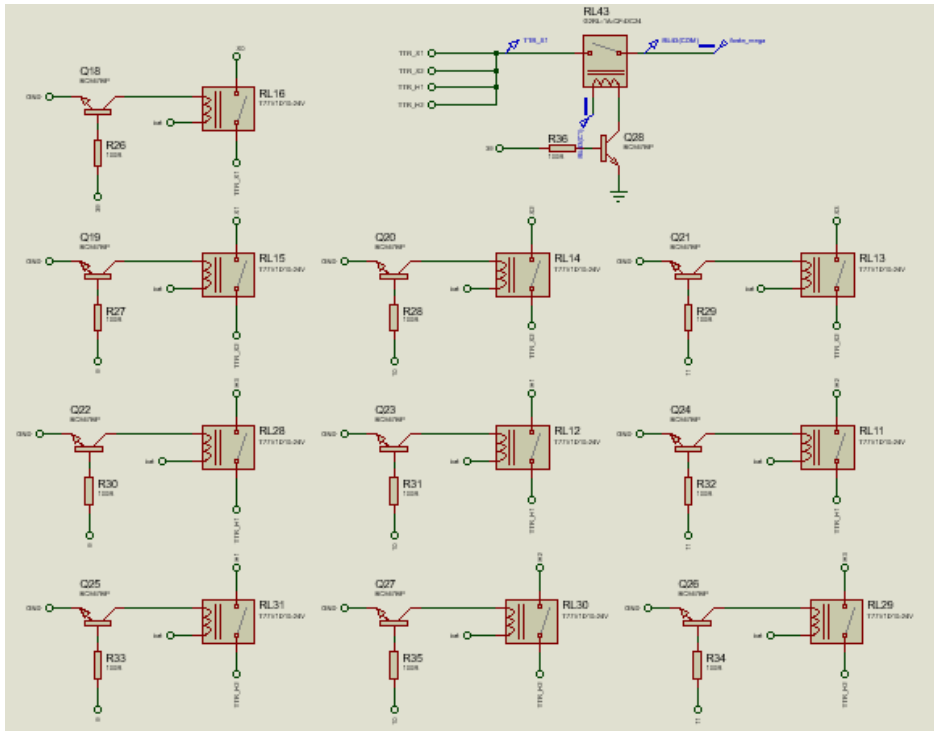


Figura E-3: Chaveamento do módulo de medição megômetro



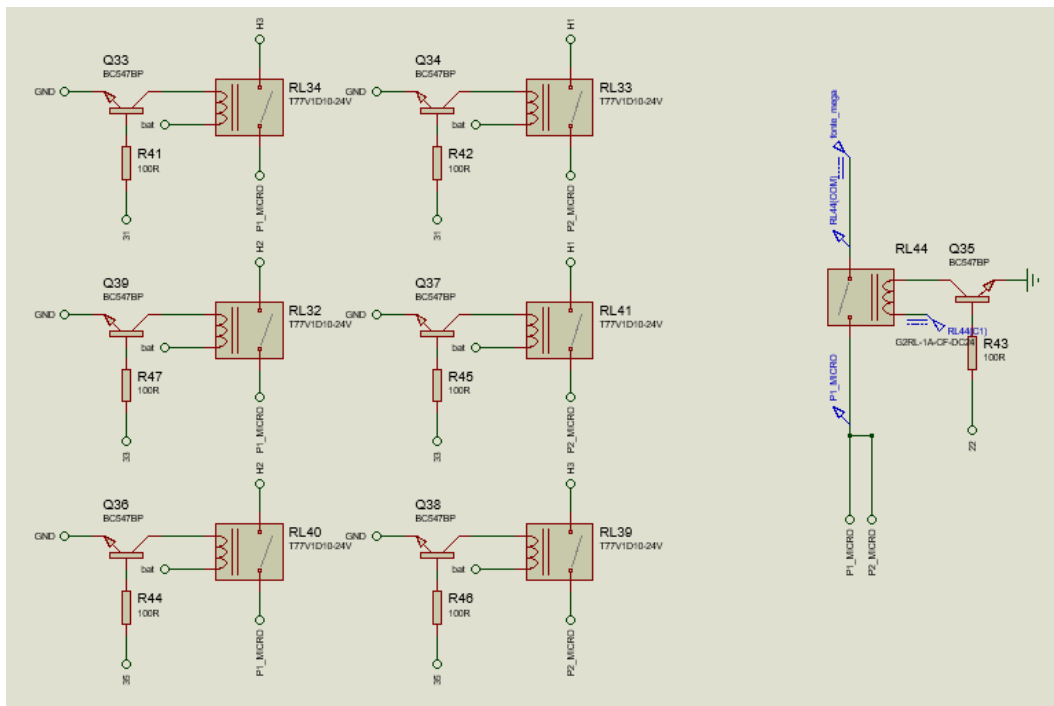
Fonte: Própria

Figura E-4: Chaveamento do módulo de medição TTR



Fonte: Própria

Figura E-5: Chaveamento do módulo de medição microhmímetro



Fonte: Própria