



**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA**

**DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E
AUTOMAÇÃO
CURSO DE ENGENHARIA MECATRÔNICA**

DANIEL HENRIQUE SILVA FERNANDES

RELATÓRIO ESTÁGIO FINAL

**NATAL
2016**

DANIEL HENRIQUE SILVA FERNANDES



EMPRESA: CTGAS-ER

ÁREA DE ATUAÇÃO: Engenharia Mecatrônica

PERÍODO DE REALIZAÇÃO: 01/04/2016 ATÉ 23/06/2016

CARGA HORÁRIA SEMANAL: 20h/semana

REMUNERAÇÃO/AJUDA: R\$ 40,00

SUPERVISOR: Hudson da Silva Resende

RELATÓRIO DE ESTÁGIO

Relatório Final de Estágio apresentado ao Curso de Engenharia Mecatrônica da Universidade Federal do Rio Grande do Norte, como requisito necessário para avaliação final na disciplina.

Coordenador de Estágio: Marcio Valério de Araújo

**NATAL
2016**

DANIEL HENRIQUE SILVA FERNANDES

RELATÓRIO ESTÁGIO FINAL

Este trabalho foi apresentado como Relatório de Estágio Final do Curso de Engenharia Mecatrônica da UFRN.

Natal, ____ de dezembro de 2015.

Professor Coordenador do Estágio

Supervisor do Estágio

RELATÓRIO ENTREGUE À COORDENAÇÃO DO CURSO EM

____/____/____.

Assinatura e carimbo do funcionário

RESUMO

Este relatório visa descrever a realização do estágio supervisionado do curso de Engenharia Mecatrônica da Universidade Federal do Rio Grande do Norte, o qual foi realizado no Centro de Tecnologias do Gás e Energias Renováveis.

O estágio teve como foco atender a duas demandas, a primeira foi a de auxiliar no desenvolvimento de um software de medição de qualidade de energia elétrica em aerogeradores. Já a segunda demanda se refere a uma necessidade de obter os dados de alguns sensores em um dessalinizador e de alguma forma salvar em algum software de planilhas para que se pudesse realizar comparativos como, por exemplo, o aumento da temperatura dos painéis coletores com o grau de incidência solar ao longo do dia.

SUMÁRIO

| | |
|------------------------------------------------------------------|----|
| 1 AMBIENTE DE ESTÁGIO | 6 |
| 2 MOTIVAÇÕES DO PROJETO | 6 |
| 3 ESPECIFICAÇÕES DO PROJETO | 7 |
| 3.1 DESENVOLVIMENTOS DE SOFTWARE PARA QUALIDADE DE ENERGIA | 7 |
| 3.2 AQUISIÇÕES DE DADOS DO DESSALINIZADOR SOLAR | 8 |
| 3.2.1 ACESSO AO FTP SERVER..... | 9 |
| 4 FASES DO PROJETO | 10 |
| 4.1 DESENVOLVIMENTO DE SOFTWARE PARA QUALIDADE DE ENERGIA..... | 10 |
| 4.2 AQUISIÇÕES DE DADOS DO DESSALINIZADOR SOLAR | 11 |
| 4.2.1 USO DO SOFTWARE WIRESHARK | 11 |
| 4.2.2 CRIAÇÃO DE PROGRAMA SOCKET | 11 |
| 4.2.3 JD-GUI..... | 12 |
| 5 FINALIZAÇÃO DE ATIVIDADES | 13 |
| 5.1 DESENVOLVIMENTO DE SOFTWARE PARA QUALIDADE DE ENERGIA..... | 13 |
| 5.2 AQUISIÇÕES DE DADOS DO DESSALINIZADOR SOLAR | 13 |
| 5.2.1 RESPOSTA AO NOVO PROGRAMA CLIENTE | 14 |
| 5.2 BIBLIOTECA OPENPXYL..... | 14 |
| 6 CONCLUSÕES..... | 15 |
| 7 REFERENCIAS | 15 |

1 AMBIENTE DE ESTÁGIO

O local onde foi realizado o estágio foi no Centro de Tecnologias do Gás e Energias Renováveis, localizado na Av. Capitão-mor Gouveia, 2770, Lagoa Nova, CEP 59064-164 – Natal/RN. Criado em 1999, a partir de uma parceria inédita e exitosa entre a Petrobras e o SENAI, o Centro de Tecnologias do Gás (CTGÁS) ampliou a sua área de atuação e passou a se chamar Centro de Tecnologias do Gás e Energias Renováveis (CTGAS-ER). Todas as atividades do estágio foram realizadas no Laboratório de Mapas e Dados de recursos naturais (LMD).

2 MOTIVAÇÕES DO PROJETO

Com relação ao desenvolvimento de software para software de medição de qualidade de energia elétrica em aerogeradores, a motivação principal se deu devido a recente expansão do setor eólico no estado aliado com a necessidade de um maior estudo no âmbito da anemometria em busca das melhores regiões geográficas que garantam um melhor retorno financeiro de investidores e uma capacidade de uma geração de energia elétrica que atenda a norma IEC 61400-21.

O projeto de armazenamento de dados do dessalinizador veio da necessidade de se entender melhor como o equipamento funcionava, visto que foi implementado em 2013 pela empresa alemã Agrosience, que em seu escopo inicial, não implementava um sistema de banco de dados no CLP do dessalinizador. Com isso, encontrar um modo de obter esses dados de uma maneira segura e de baixo custo, ou seja, sem atrapalhar o funcionamento atual do equipamento e de preferência sem custos em licenças de software, se tornou o maior objetivo do estágio.

3 ESPECIFICAÇÕES DO PROJETO

3.1 DESENVOLVIMENTOS DE SOFTWARE PARA QUALIDADE DE ENERGIA

Para a realização dos testes do software um anemômetro de copo junto ao uma biruta foi posto em uma pequena torre anemométrica e seus dados foram enviados através de uma antena e logo, poucos metros depois, outra antena recebe esses dados, envia a um Datalogger. Logo, a demanda a ser preenchida se resume ao um programa que fizesse a comunicação entre o computador e o Datalogger. Para isso a escolha para essa pratica foi a utilização do software LabView cujo CTGAS possuía uma licença para trabalho.

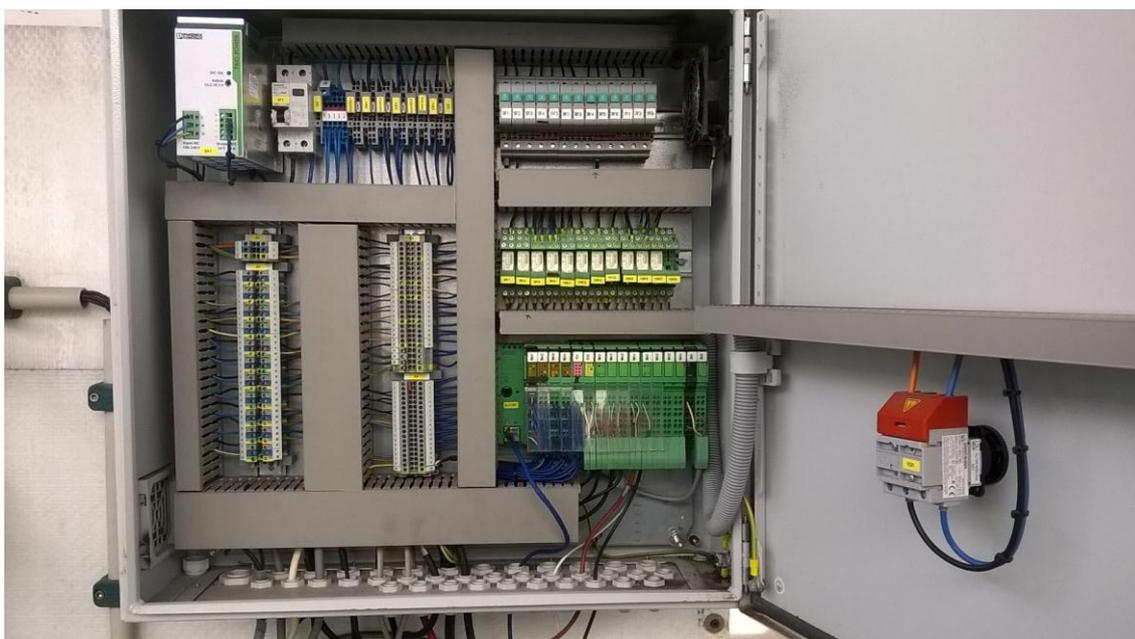


Com sua sintaxe de programação gráfica, que torna simples visualizar, criar e desenvolver códigos para sistemas de engenharia, o ambiente LabVIEW ajuda a reduzir os tempos dos testes e oferecer informações com base nos dados coletados. O LabVIEW oferece ainda integração sem precedentes com hardware de medição, programas de software anteriores e códigos existentes, capitalizando ainda as vantagens das tecnologias mais recentes de computação.

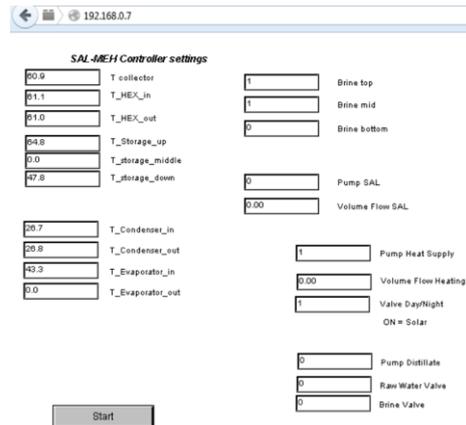
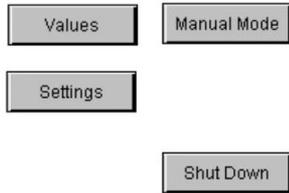
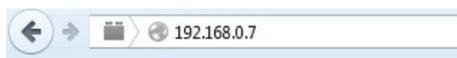
3.2 AQUISIÇÕES DE DADOS DO DESSALINIZADOR SOLAR

O dessalinizador solar do CTGAS surgiu da parceria com a AgroScience, instituto do governo alemão que atua na área de pesquisa, com quem a vem mantendo intercâmbio com vista ao desenvolvimento de estudos do interesse da indústria FIERN. Para se ter uma ideia da importância desse trabalho, para o semiárido nordestino, região com grande incidência solar e muita água subterrânea salinizada, o equipamento, ao final da pesquisa, utilizando água do mar, demonstrou capacidade para fornecer o líquido com percentuais bem abaixo dos exigidos pela legislação do Brasil e da Alemanha.

Com isso, o objetivo agora do projeto é de compreender mais o equipamento em vista de uma futura nacionalização do sistema, logo, fez-se necessário conhecer mais o comportamento do sistema ao longo do tempo e para isso criar um sistema que armazena os valores dos sensores fornecidos pelo equipamento se tornou uma prioridade.



Para acessar os valores dos sensores em tempo real, o ILC 150 do dessalinizador fornece um endereço o endereço http, basta conectar o cabo de rede do CLP a algum computador e acessar o site 192.168.0.7 como mostrado na imagem a seguir. Os valores que deveriam ser armazenados se encontram em 'Values', nele é possível observar a variação dos parâmetros com equipamento com o tempo, como mostrado na segunda imagem.



Todo o projeto do dessalinizador foi projetado pelo software PCWorx, da Phoenix Contact, empresa alemã líder de mercado e agente de inovação global da área eletrotécnica. A princípio, por mais que o controlador incluisse módulos para exportação dos dados no formato csv ou até mesmo protocolos de linguagens de banco de dados, como MySQL por exemplo, foi pensado que adquirir uma licença do software implicaria em comprometimento financeiro e algum possível dano no projeto do dessalinizador. Portanto, a ideia era de adquirir os dados de forma a manter o funcionamento do equipamento já instalado.

3.2.1 1 ACESSO AO FTP SERVER

Além do site http, o ILC 150 fornece um endereço do servidor FTP onde é possível observar algumas páginas e arquivos que são executadas pelo controlador, nele é possível observar, dentre outros arquivos o arquivo IMasterPhoenix6_10_00.jar, esse programa foi feito em java pelos desenvolvedores do controlador e tem como função básica servir de intermédio entre a programação feita no PcWorx e o cli com o propósito de aprimorar certas funções e facilitar a programação no controlador.

Conteúdo do diretório ftp://192.168.0.7/

Um diretório acima

- | Nome |
|------------------------------|
| SPIDERALRTRD |
| de.csv |
| en.csv |
| entry.html |
| es.csv |
| fr.csv |
| IMasterPhoenix6_10_00.jar |
| MsgBox.teq |
| Visio_SAL_CTGas-manu.teq |
| Visio_SAL_CTGas-settings.teq |
| Visio_SAL_CTGas-start.teq |
| Visio_SAL_CTGas-values.teq |
| Visio-SAL-CTGas.itq |
| Visio-SAL-CTGas.prj |
| Visio-SAL-CTGas.tcr |
| Visio-SAL-CTGas-shutdown.teq |

4.2 AQUISIÇÕES DE DADOS DO DESSALINIZADOR SOLAR

As principais etapas que permitiram a aquisição de dados com o tempo se resumem ao: uso do software WireShark para o melhor entendimento da transferência de pacotes entre cliente e servidor; criação de um programa com implementação de bibliotecas de socket e ‘descompilação’ do programa IMasterPhoenix6_10_00.jar utilizando a ferramenta JD-GUI.

4.2.1 USO DO SOFTWARE WIRESHARK

O WireShark é um analisador de pacotes de rede que visa capturar esses pacotes da melhor maneira mais detalhada e simples possível. O uso desse programa permitiu observar detalhadamente quais os tipos de requisição o usuário como o cliente precisa requisitar ao servidor para que o mesmo envie uma resposta. A seguir, é possível observar o comportamento dos pacotes ao cliente entrar no endereço do clp e clicar no botão ‘Values’.

4.2.2 CRIAÇÃO DE PROGRAMA SOCKET

Observando o tipo de solicitação de pacote que o cliente deveria fazer, restava agora criar um programa que enviasse a mesma solicitação e armazenava a resposta em um sistema de aquisição de dados. A linguagem escolhida para realizar esse procedimento foi Python por ser uma linguagem simples, intuitiva e com uma biblioteca simples chamada openpyxl que permitia criar arquivos no excel de maneira rápida e eficiente, pois a princípio os dados seriam salvos num ambiente de planilhas para a rápida observação do comportamento do programa.

Para o recebimento do pacote foi criado o seguinte programa

```

# Teste de programa Cliente
import socket

HOST = '192.168.0.7' # The remote host
PORT = 80 # The same port as used by the server
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
s.sendall('POST 192.168.0.7/cgi-bin/ILRRReadValues.exe HTTP1.1/r/n')
data = s.recv(1024)
s.close()
print 'Received', repr(data)

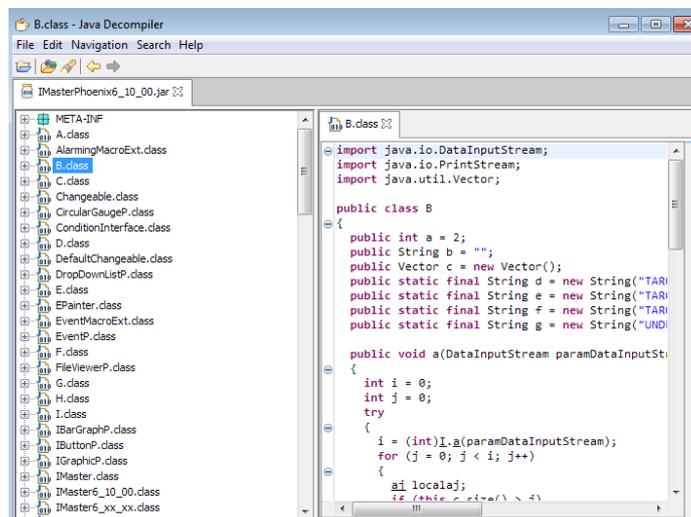
```

É possível notar que o comando enviado na linha 8 é o mesmo da imagem anterior observado pelo Wireshark. Entretanto a resposta a essa solicitação retornava o valor da variável data como sendo igual a 0, ou seja, a resposta estava longe de ser a esperada, restava tentar outras estratégias para a leitura de variáveis.

4.2.3 JD-GUI

Na busca de encontrar um método de enviar alguma solicitação para que o servidor retornasse o valor esperado, usamos o software JD-GUI para analisar o programa IMasterPhoenix6_10_00.jar em busca de encontrar um endereço de IP distinto que retornasse o valor das variáveis do sistema do dessalinizador.

O JD-GUI é um utilitário gráfico autônomo que exibe códigos-fonte em Java de arquivos do tipo '.class'. Você pode navegar o código fonte reconstruído com o JD-GUI para acesso instantâneo a métodos e campos. A seguir é mostrado uma imagem do programa descompilado pelo JD-GUI.



Muitas das classes do programa não possuem nenhuma indicação do que executa, logo procurar entender o programa de forma eficiente tomou dias até que na procura do comando certo, a classe 'z.class' mostrou uma pista indicativa do que poderia ser uma alternativa de enfileiramento por socket do programa:

```
protected String f(String paramString)
{
    String str = this.b + "/" + this.a + "/cgi-bin/readVal.exe?" + paramString;
    if (l.kq.bd) {
        str = this.b + "/" + this.a + "/cgi-bin/readVal.cgi?" + paramString;
    }
    if (l.kv.length() > 0)
    {
        str = this.b + "/" + l.kv + "/cgi-bin/readVal.exe?" + paramString;
        if (l.kq.bd) {
            str = this.b + "/" + l.kv + "/cgi-bin/readVal.cgi?" + paramString;
        }
    }
    return k(str);
}
```

Analisando essa função, é possível observar que o valor "this.b" se referia ao termo "http", enquanto que o "this.a" se referia ao endereço de IP da máquina e o valor "paramString" variava conforme o nome que o CLP quando foi programado deu as suas variáveis. Restava agora saber quais os valores de cada variável, que foi obtida novamente usando o software Wireshark.

5 FINALIZAÇÃO DE ATIVIDADES

5.1 Desenvolvimentos de Software para qualidade de energia

O software passou a responder da forma desejada e plotar adequadamente o valor da velocidade do vento do anemômetro de copo em tempo real, restando apenas a implementação a nível de programação dos cálculos de qualidade de energia.

5.2 AQUISIÇÕES DE DADOS DO DESSALINIZADOR SOLAR

Com todas as ferramentas em mãos, restava realizar os testes necessários para a aquisição dos dados e a exportação dos dados em planilha no formato tipo '.xlsx'.

5.2.1 RESPOSTA AO NOVO PROGRAMA CLIENTE

Para o teste do novo programa cliente a única mudança foi na linha 8 que é mostrada a seguir que no caso manda ao servidor uma solicitação para saber o valor da temperatura dos coletores.

```
# Teste de programa Cliente
import socket

HOST = '192.168.0.7'    # The remote host
PORT = 80              # The same port as used by the server
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
s.sendall('GET 192.168.0.7/cgi-bin/readVal.exe?@GV.rT_coll HTTP/1.1 \r\n\r\n')
data = s.recv(1024)
s.close()
print 'Received', repr(data)
```

Com o envio desse código foi possível obter um texto com as características do controlador ILC 150 da Phoenix Contact e ao final o mesmo valor de temperatura do painéis coletores.

5.2.2 BIBLIOTECA OPENPXYL

Openpyxl é uma biblioteca Python para ler e escrever Excel 2010 arquivos xlsx, xlsxm, xlsx e XLTM. Essa biblioteca foi integrada para poder obter e armazenar os dados no formato .xlsx para garantir a facilidade na geração de gráficos para uma melhor análise do equipamento.

6 CONCLUSÕES

Por motivos que excederam nosso limite, o projeto do software para a qualidade de energia elétrica sofreu algumas dificuldades administrativas que impossibilitaram algumas respostas das implementações mais avançadas no sistema, porém, após o sucesso em realizar a comunicação com o datalogger e visualizar a ação da planta em tempo real, os próximos passos seguirão mais firmes.

Agora com relação a aquisição de dados do dessalinizador, o programa em python mostrou-se eficiente em criar arquivos no formato .xlsx diariamente e com isso, dar continuidade ao projeto resumisse a implementar um sistema de banco de dados, o que muito provavelmente deve ser com MySQL ou SQLite.

7 REFERÊNCIAS

- TechTudo, “Como usar o Wireshark”, disponível em:
<<http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2012/09/como-usar-o-wireshark.html>> Acessado em 20 de Junho de 2016.
- JD Project, JD-GUI, disponível em:< <http://jd.benow.ca/>> Acessado em 20 de Junho de 2016.
- Openpxyl, Openpxyl, disponível em:
<<https://openpyxl.readthedocs.io/en/default/>> Acessado em 20 de Junho de 2016.