



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

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

**LUCAS ROLIM LARICCHIA**

**RELATÓRIO ESTÁGIO FINAL**

**NATAL  
2016**

## LUCAS ROLIM LARICCHIA



**EMPRESA:** CTGAS-ER

**ÁREA DE ATUAÇÃO:** Engenharia Mecatrônica

**PERÍODO DE REALIZAÇÃO:** 01/04/2016 ATÉ 30/06/2016

**CARGA HORÁRIA SEMANAL:** 20h/semana

**REMUNERAÇÃO/AJUDA:** R\$ 40,00/Mês

**SUPERVISOR:** Hudson da Silva Resende

## RELATÓRIO ESTÁGIO FINAL

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: Diogo Pinheiro Fernandes Pedrosa

**NATAL  
2016**

**LUCAS ROLIM LARICCHIA**

**RELATÓRIO ESTÁGIO FINAL**

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

Natal, \_\_\_\_ de Junho de 2016.

---

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 na empresa CTGÁS-ER.

O estágio teve como foco principal projetos relacionados a área de engenharia elétrica e automação, com foco também na parte de redes e aquisição de dados, onde foi desenvolvido um banco de dados para um sistema de dessalinização da água que existe no CTGÁS-ER.

## SUMÁRIO

1 Ambiente de estágio .....	6
2 Motivação do projeto.....	6
3 Especificações do projeto .....	7
4 Fases do projeto .....	8
5 Atividades desenvolvidas .....	9
5.1 Utilização do WireShark .....	9
5.2 Utilização do WebVisit .....	10
5.3 Utilização do WebVisit .....	11
6 Conclusões.....	10
Referencias.....	12

## 1. Ambiente de estágio

O local onde foi realizado o estágio, CTGAS-ER, é localizada na Avenida Capitão-Mor Gouveia, Nº 1480, Lagoa Nova, Natal/RN, CEP 59063-400.

O CTGAS-ER possui diversas salas de aula onde são ministrados cursos de nível técnico em diversas áreas, tanto da mecânica quanto da elétrica, sendo assim um local de alto nível de aprendizado, onde muitos profissionais que trabalham lá também ministram cursos.

O ambiente é conhecido por suas pesquisas na área de energias renováveis, tanto solar quanto eólica, onde são feitos análises e estudos climáticos e tecnológicos para sempre se buscar maior eficiência na produção de energia.

## 2. Motivações do projeto

Ao iniciar as atividades percebeu-se uma necessidade de implementação de um banco de dados ao sistema de dessalinização solar, pois é um sistema que está sendo utilizado há muito tempo para estudos de viabilidade sobre a tecnologia, o que torna imprescindível um registro das temperaturas dos coletores, acionamento de válvulas, e vazões para que seja feito estudos de eficiência.

Figura 1 – Dessalinizador Solar.



Tendo em vista que diversas pessoas já tentaram realizar a implementação do banco de dados no sistema de dessalinização, o projeto se mostra mais importante ainda, pois ficaria como uma grande ajuda para os engenheiros que estão trabalhando com ele.

### 3. Especificações do projeto

Para obter êxito na tarefa de implementação do banco de dados, primeiramente foi necessário um entendimento mais a fundo de como os dados trafegam pela rede, para isso diversas técnicas de análise foram utilizadas seguidas de vários testes realizados.

Figura 2 – CLP ILC150 da Phoenix Contact



Fonte: [www.phoenixcontact.com/](http://www.phoenixcontact.com/) (2016)

Este CLP possui uma porta Ethernet que possibilita o envio de dados do processo via rede, a empresa que instalou o sistema no CTGAS foi uma empresa Alemã, devido a dificuldades na compreensão dos manuais causadas por falta de conhecimento na língua alemã, toda a parte de transferência de dados estava praticamente ilegível para os trabalhadores do CTGAS.

O sistema já estava configurado para exibir online os dados obtidos no funcionamento da planta de dessalinização, ao conectar o computador a rede local da planta, e acessar o IP 192.168.0.7 podia-se visualizar a seguinte janela de exibição, mostrada na figura 3.

Figura 3 – Janela de visualização de dados

**SAL-AREH Controller settings**

<input type="text" value="20.9"/>	T_collector	<input type="text" value="1"/>	Brine top
<input type="text" value="21.1"/>	T_HEX_in	<input type="text" value="1"/>	Brine mid
<input type="text" value="21.0"/>	T_HEX_out	<input type="text" value="0"/>	Brine bottom
<input type="text" value="24.8"/>	T_storage_up	<input type="text" value="0"/>	Pump SAL
<input type="text" value="0.0"/>	T_storage_middle	<input type="text" value="0.00"/>	Volume Flow SAL
<input type="text" value="17.8"/>	T_storage_down	<input type="text" value="1"/>	Pump Heat Supply
<input type="text" value="28.7"/>	T_Condenser_in	<input type="text" value="0.00"/>	Volume Flow Heating
<input type="text" value="28.8"/>	T_Condenser_out	<input type="text" value="1"/>	Valve Day/Night DN = Solar
<input type="text" value="43.3"/>	T_Evaporator_in	<input type="text" value="0"/>	Pump Distillate
<input type="text" value="9.0"/>	T_Evaporator_out	<input type="text" value="0"/>	Raw Water Valve
		<input type="text" value="0"/>	Brine Valve

#### 4. Fases do projeto

Para a realização do projeto proposto se mostrava necessário à implementação de uma forma de coletar os dados da janela de visualização padrão do CLP e salva-los em alguma forma de banco de dados, para que os mesmos fossem analisados no futuro, porém não se tinha conhecimento sobre o caminho que os dados percorriam do CLP até a interface já existente.

Este projeto foi dividido em três partes, primeiramente foi utilizado um programa chamado WireShark com o intuito de ler os pacotes enviados pela rede, a fim de entender melhor como os dados saem do CLP e vão para a interface de exibição, depois foi utilizado um programa da própria empresa distribuidora do CLP, Phoenix Contact, para se conhecer melhor sobre a interface de visualização e suas variáveis, o programa utilizado chama-se WebVisit, e possibilitou a visualização do nome das variáveis utilizadas no CLP. Por fim foi desenvolvidos programas em C++ e em Python para a captação dos dados e seu armazenamento, criando o banco de dados final.

## 5. Atividades desenvolvidas

### 5.1 Utilização do WireShark

Inicialmente foi utilizado o programa de leitura de pacotes da rede, WireShark (ver figura 4), com o intuito de realizar uma leitura dos dados enviados pelo CLP.

Figura 4 – Saída do WireShark.

No.	Time	Source	Destination	Protocol	Leng	Info
1	0.000000	192.168.0.1	192.168.0.7	TCP	66	55642 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.003894	192.168.0.7	192.168.0.1	TCP	60	80 → 55642 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0 MSS=1460
3	0.003955	192.168.0.1	192.168.0.7	TCP	54	55642 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
4	0.005340	192.168.0.1	192.168.0.7	TCP	366	[TCP segment of a reassembled PDU]
5	0.005388	192.168.0.1	192.168.0.7	HTTP/XML	895	POST /cgi-bin/ILRReadValues.exe HTTP/1.1
6	0.008090	192.168.0.7	192.168.0.1	TCP	60	80 → 55642 [ACK] Seq=1 Ack=1154 Win=2920 Len=0
7	0.147393	192.168.0.7	192.168.0.1	HTTP/XML	13...	HTTP/1.0 200 OK
8	0.147854	192.168.0.1	192.168.0.7	TCP	54	55642 → 80 [FIN, ACK] Seq=1154 Ack=1276 Win=62965 Len=0
9	0.153408	192.168.0.7	192.168.0.1	TCP	60	80 → 55642 [FIN, ACK] Seq=1276 Ack=1155 Win=2920 Len=0
10	0.153472	192.168.0.1	192.168.0.7	TCP	54	55642 → 80 [ACK] Seq=1155 Ack=1277 Win=62965 Len=0
11	0.217349	192.168.0.1	192.168.0.255	NBNS	110	Registration NB <01><02>_MSBROWSE__<02><01>
12	0.513657	192.168.0.1	192.168.0.255	NBNS	92	Name query NB WWW.GOOGLE.COM<00>
13	0.864942	fe80::61a4:e929:c09...	ff02::1:3	LLMNR	84	Standard query 0x06f6 A wpad
14	0.865222	192.168.0.1	224.0.0.252	LLMNR	64	Standard query 0x06f6 A wpad
15	0.865322	fe80::61a4:e929:c09...	ff02::1:3	LLMNR	84	Standard query 0xcc2f A wpad
16	0.865535	192.168.0.1	224.0.0.252	LLMNR	64	Standard query 0xcc2f A wpad

Ao observar as saídas do programa, notou-se uma grande quantidade de pacotes utilizando o método “POST”, saindo do computador para o CLP, como forma de “solicitação” de dados, realizada pela interface ao exibir dados para o usuário, percebeu-se também a utilização da porta 80, que é padrão para uso em HTTP.

Com o WireShark foi possível a visualização do endereço exato para o acesso a variáveis, como pode-se notar na linha 5 da figura 4, “ POST /cgi-bin/ILRReadValues.exe HTTP/1.1”.

Porém, o método POST mostrou-se inviável para utilização com programas desenvolvidos em C++ e em Python, após algumas tentativas utilizando o mesmo programa, WireShark, descobriu-se o uso de outro método, o “GET”

Figura 5 – Saída com método “GET”.

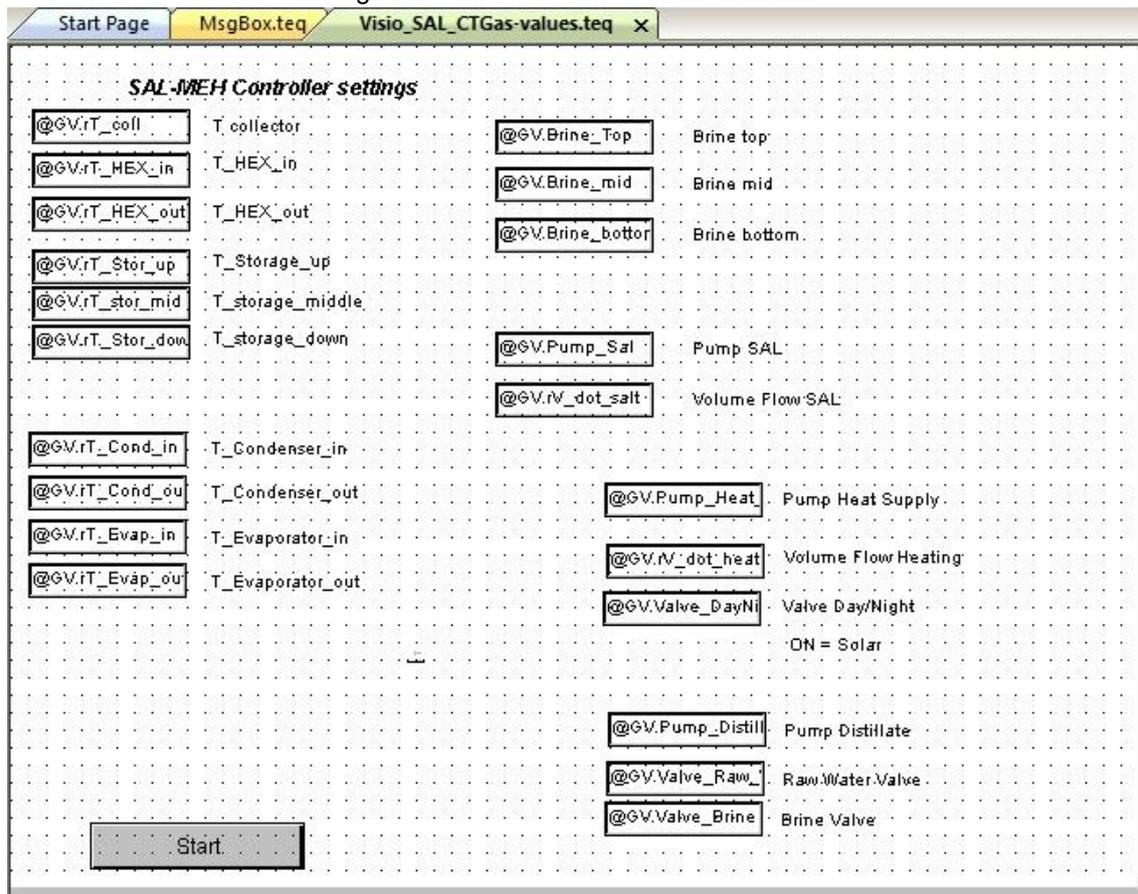
No.	Time	Source	Destination	Protocol	Leng	Info
1	0.000000	192.168.0.1	192.168.0.7	TCP	66	64414 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.013128	192.168.0.7	192.168.0.1	TCP	60	80 → 64414 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0 MSS=1460
3	0.013282	192.168.0.1	192.168.0.7	TCP	54	64414 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
4	0.013432	192.168.0.1	192.168.0.7	HTTP	389	GET /cgi-bin/readVal.exe?@GV.rT_coll HTTP/1.1
5	0.017347	192.168.0.7	192.168.0.1	TCP	60	80 → 64414 [ACK] Seq=1 Ack=336 Win=2585 Len=0
6	0.029348	192.168.0.7	192.168.0.1	HTTP	303	HTTP/1.0 200 OK (text/plain)
7	0.029421	192.168.0.1	192.168.0.7	TCP	54	64414 → 80 [ACK] Seq=336 Ack=250 Win=65535 Len=0
8	0.029514	192.168.0.1	192.168.0.7	TCP	54	64414 → 80 [FIN, ACK] Seq=336 Ack=250 Win=65535 Len=0
9	0.030572	192.168.0.7	192.168.0.1	TCP	60	80 → 64414 [FIN, ACK] Seq=250 Ack=336 Win=2920 Len=0
10	0.030624	192.168.0.1	192.168.0.7	TCP	54	64414 → 80 [ACK] Seq=337 Ack=251 Win=65535 Len=0
11	0.036275	192.168.0.7	192.168.0.1	TCP	60	[TCP Out-Of-Order] 80 → 64414 [FIN, ACK] Seq=250 Ack=337 Win=2920 Len=0
12	5.184719	192.168.0.1	192.168.0.7	TCP	66	64415 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1

Como foi possível observar na linha 4 da figura 5 o CLP também utilizava de um método GET que se mostrou mais específico para cada variável, com ele é possível receber cada variável individualmente, restando assim apenas saber o nome das variáveis, para poder utilizando este método obter os seus valores.

## 5.2. Utilização do WebVisit

Após os resultados obtidos com o WireShark, notou-se a necessidade de obtenção do nome de cada variável utilizada no processo, para acessa-las via socket, utilizando um programa desenvolvido inicialmente em C++ e depois um programa final desenvolvido em Python, para obter essas variáveis foi utilizado um programa da Phoenix Contact chamado de WebVisit, nele é possível desenvolver interfaces de comunicação entre o CLP e um dispositivo conectado a rede, o projeto em WebVisit do CLP foi obtido na memória interna do próprio CLP e lido, assim foi obtido o nome das variáveis como mostra a figura 6.

Figura 6 – Variáveis listadas no WebVisit

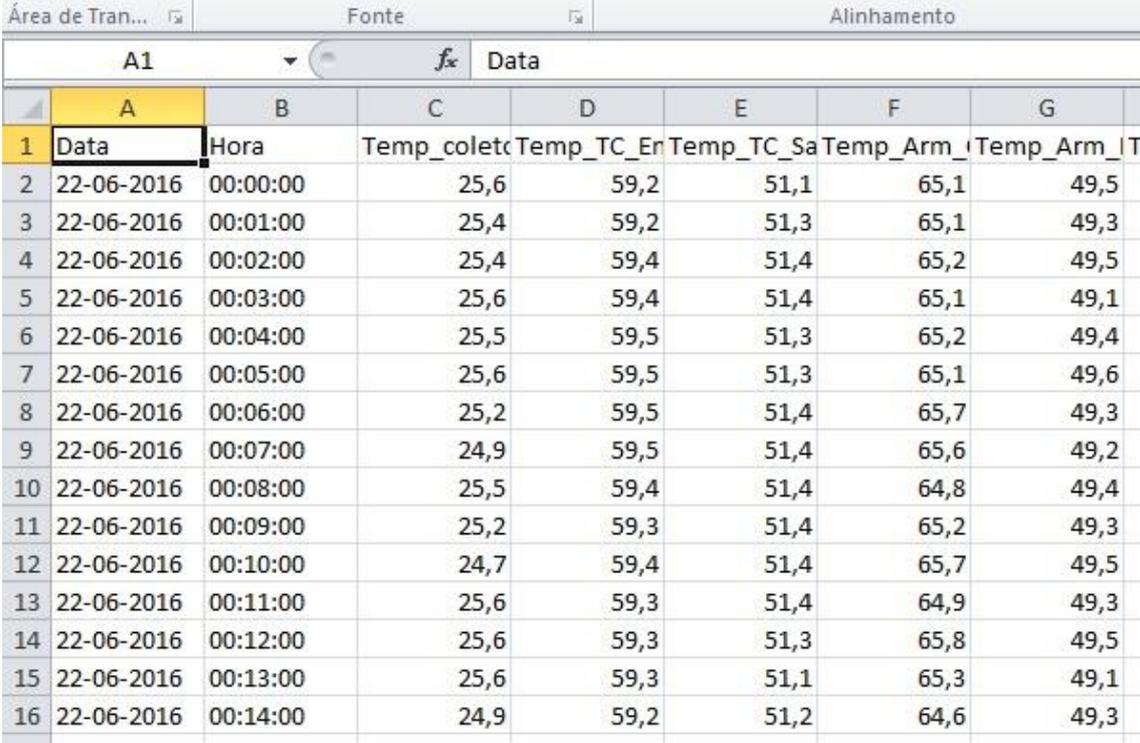


### 5.3. Desenvolvimento do programa em Python

Com o nome das variáveis, e os endereços obtidos, restava apenas o desenvolvimento de algum tipo de programa, para ler as variáveis uma a uma e salva-las numa espécie de banco de dados.

Para isto foi desenvolvido um programa em C++, que depois foi implementado em Python, que lê variável por variável, utilizando socket, e salva todas em um arquivo de Excel para ser utilizado futuramente para análises. O programa teve que sofrer pequenas modificações para atender os padrões do CTGAS, como medições de um em um minuto, e nomenclaturas específicas.

Figura 7 – Saída do programa



	A	B	C	D	E	F	G
1	Data	Hora	Temp_colet	Temp_TC_En	Temp_TC_Sa	Temp_Arm_I	Temp_Arm_IT
2	22-06-2016	00:00:00	25,6	59,2	51,1	65,1	49,5
3	22-06-2016	00:01:00	25,4	59,2	51,3	65,1	49,3
4	22-06-2016	00:02:00	25,4	59,4	51,4	65,2	49,5
5	22-06-2016	00:03:00	25,6	59,4	51,4	65,1	49,1
6	22-06-2016	00:04:00	25,5	59,5	51,3	65,2	49,4
7	22-06-2016	00:05:00	25,6	59,5	51,3	65,1	49,6
8	22-06-2016	00:06:00	25,2	59,5	51,4	65,7	49,3
9	22-06-2016	00:07:00	24,9	59,5	51,4	65,6	49,2
10	22-06-2016	00:08:00	25,5	59,4	51,4	64,8	49,4
11	22-06-2016	00:09:00	25,2	59,3	51,4	65,2	49,3
12	22-06-2016	00:10:00	24,7	59,4	51,4	65,7	49,5
13	22-06-2016	00:11:00	25,6	59,3	51,4	64,9	49,3
14	22-06-2016	00:12:00	25,6	59,3	51,3	65,8	49,5
15	22-06-2016	00:13:00	25,6	59,3	51,1	65,3	49,1
16	22-06-2016	00:14:00	24,9	59,2	51,2	64,6	49,3

Como pode ser observado na figura 7, a saída do programa possui todas as medições da planta de dessalinização, onde cada variável ocupa uma coluna. As temperaturas estão em graus célsius e ele realiza medições das 00 horas até as 23:59 do mesmo dia, ao completar um dia o programa cria um outro arquivo e assim por diante, o nome dos arquivos são iguais a data do mesmo dia.

## **6. Conclusões**

O sistema de banco de dados foi implementado com sucesso no CTGAS, hoje existe um computador dedicado a captação de dados do dessalinizador solar, ele fica conectado a rede local e a cada um minuto faz uma medição de todos os parâmetros.

Para quem possui um trabalho relacionado a planta de dessalinização o projeto se mostrou como um grande avanço nos estudos de eficiência da planta, pois agora pode-se comparar níveis de radiação solar com as temperaturas e acionamentos de válvulas agora registrados no sistema.

## 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.
- Canal com tutoriais para conexões de rede <<https://www.youtube.com/user/VoidRealms>> Acessado em 2 de abril de 2016.