

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO**

DHIONE MATHIAS DAS NEVES

**MONITORAMENTO *ONLINE* DE GRANDEZAS ELÉTRICAS
DO CENTRO TECNOLÓGICO XI**

VITÓRIA – ES
ABRIL/2017

DHIONE MATHIAS DAS NEVES

**MONITORAMENTO *ONLINE* DE GRANDEZAS ELÉTRICAS DO
CENTRO TECNOLÓGICO XI**

Parte manuscrita do Projeto de Graduação do aluno **Dhione Mathias das Neves**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Orientador: Prof. Dr. Evandro Ottoni Teatini Salles

VITÓRIA – ES
ABRIL/2017

DHIONE MATHIAS DAS NEVES

**MONITORAMENTO *ONLINE* DE GRANDEZAS ELÉTRICAS DO
CENTRO TECNOLÓGICO XI**

Parte manuscrita do Projeto de Graduação do aluno **Dhione Mathias das Neves**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Aprovada em 05 de abril de 2017.

COMISSÃO EXAMINADORA:

Prof. Dr. Evandro Ottoni Teatini Salles
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. André Ferreira
Universidade Federal do Espírito Santo
Examinador

Prof. Dr. Marcelo Eduardo Vieira Segatto
Universidade Federal do Espírito Santo
Examinador

Aos meus pais, Maria Delmi e João Luis.

AGRADECIMENTOS

Agradeço primeiramente a Deus, nosso bondoso e amoroso Pai, pelo dom da vida e por estar sempre ao meu lado durante todos os momentos.

Aos meus pais, meus maiores incentivadores e apoiadores. Sustentaram-me em todos os momentos das mais diversas formas, sempre dizendo palavras de muito carinho e de otimismo.

Aos amigos de curso, que sempre estiveram ao meu lado sempre contribuindo com meu aprendizado e ajudando nos momentos difíceis do curso.

Aos professores e servidores do Departamento de Engenharia Elétrica.

RESUMO

Neste trabalho é apresentado o desenvolvimento de um sistema de monitoramento *online* de grandezas elétricas do Centro Tecnológico XI (CT-XI) do Campus Goiabeiras da Universidade Federal do Espírito Santo – UFES. As informações elétricas são acessadas a partir da porta serial de um medidor eletrônico de energia elétrica, instalado no quadro geral de distribuição de energia do prédio citado, sendo exibidas em uma página *Web*. O medidor eletrônico lê os valores de corrente e tensão e gera, em tempo real, diversas grandezas elétricas sobre a qualidade da energia elétrica. Um computador embarcado, por sua vez, lê, por meio da interface serial RS485-USB, esses indicadores de energia e disponibiliza na página *Web*, além de armazená-los sob a forma de uma banco de dados. O computador disponibiliza na rede UFES os dados lidos por meio de uma conexão *WIFI* no prédio do CT-XI. Para isso, foi desenvolvido um conjunto de soluções em software, empregando-se linguagens de programação utilizadas para desenvolvimento da página *Web* e protocolos de comunicação de dados. Na página *Web*, o usuário pode observar as medidas tanto em forma de texto corrido como em forma de tabelas e gráficos. O documento também descreve as etapas do gerenciamento dos dados obtidos via software e, por fim, mostra os testes realizados no sistema desenvolvido. Os resultados obtidos mostraram um funcionamento satisfatório do sistema.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 – Equipamentos para o monitoramento..... | 13 |
| Figura 2 – Diagrama do sistema. | 13 |
| Figura 3 – Medidor eletromecânico de energia elétrica Nansen. | 15 |
| Figura 4 – Composição de um medidor eletromecânico..... | 16 |
| Figura 5 – Medidor eletrônico de energia elétrica Kron. | 17 |
| Figura 6 – Diagrama de blocos que representa o funcionamento do medidor eletrônico de energia elétrica..... | 17 |
| Figura 7 – Configuração de uma rede com padrão RS-485. | 20 |
| Figura 8 – Representação da energia medida em quadrantes..... | 22 |
| Figura 9 – Instalação do medidor..... | 22 |
| Figura 10 – Transformador de corrente HB 603 não articulado..... | 23 |
| Figura 11 – Esquemático da instalação final do medidor Kron..... | 24 |
| Figura 12 – Especificações do <i>Raspberry PI</i> | 27 |
| Figura 13 – Página de configuração do <i>phpMyAdmin</i> | 32 |
| Figura 14 – Fluxograma para apresentação via tabelas e gráficos. | 33 |
| Figura 15 – Diagrama de blocos para aquisição ou exibição de dados..... | 34 |
| Figura 16 – Teste de envio de mensagem ao medidor..... | 35 |
| Figura 17 – Validação da conversão dos dados lidos. | 35 |
| Figura 18 – Página inicial do site desenvolvido. | 36 |
| Figura 19 – Parte da página opção Leituras. | 37 |
| Figura 20 – Página de atualização de dados..... | 37 |
| Figura 21 – Página da opção tabelas..... | 38 |
| Figura 22 – Página da opção gráficos. | 39 |

LISTA DE QUADROS

| | |
|---|----|
| Quadro 1 – Grandezas elétricas medidas pelo medidor Kron. | 18 |
| Quadro 2 – Estrutura do quadro de mensagem via <i>MODBUS</i> | 21 |
| Quadro 3 – Estrutura da tabela do banco de dados. | 31 |
| Quadro 4 – Características dos bits do padrão IEEE-754 | 56 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------|---|
| A/D | Analógico/digital |
| DHT | Distorção harmônica total |
| CA | Corrente alternada |
| CC | Corrente contínua |
| RTU | <i>Remote Terminal Unit</i> (Unidade Terminal Remota) |
| CT | Centro Tecnológico |
| UFES | Universidade Federal do Espírito Santo |
| TC | Transformador de Corrente |
| ABB | Asea Brown Boveri |
| TP | Transformador de Potencial |
| TL | Tipo de Ligação |
| TI | Tempo de Integração |
| RAM | <i>Random Access Memory</i> (Memória de acesso randômico) |
| CPU | <i>Central Processing Unit</i> (Unidade central de processamento) |
| ARM | <i>Advanced RISC Machine</i> (Máquina RISC Avançada) |
| USB | <i>Universal Serial Bus</i> (Barramento Serial Universal) |
| GPIO | <i>General Purpose Input/Output</i> (Entrada/Saída de Uso Geral) |
| HDMI | <i>High Definition Multimedia Interface</i> (Interface Multimídia de Alta Definição) |
| LAN | <i>Local Area Network</i> (Rede Local) |
| GPU | <i>Graphics Processing Unit</i> (Unidade de Processamento de Gráficos) |
| SOC | <i>System on a chip</i> (Sistema em um chip) |
| WIFI | <i>Wireless Fidelity</i> (Conexão sem fio) |
| PHP | <i>PHP: Hypertext Processor</i> (Processador de Hipertexto PHP) |
| HTML | <i>HyperText Markup Language</i> (Linguagem de marcação de Hipertexto) |
| SQL | <i>Structured Query Language</i> (Linguagem Estruturada de Consulta) |
| HTTP | <i>Hypertext Transfer Protocol</i> (Protocolo de Transferência de Hipertexto) |
| OSI | <i>Open Systems Interconnection</i> (Interconexão de Sistemas Abertos) |
| IEEE | <i>Institute of Electrical and Electronic Engineers</i> (Instituto de Engenheiros Eletricistas e Eletrônicos) |
| IoT | <i>Internet of Things</i> (<i>Internet</i> das coisas) |
| LSB | <i>Least significant bit</i> (bit menos significativo) |
| MSB | <i>Most significant bit</i> (bit mais significativo) |

LISTA DE SÍMBOLOS

| | |
|------------------|--------------------------------------|
| V_{RMS} | Tensão RMS |
| I_{RMS} | Corrente RMS |
| P | Potência ativa |
| I_i | Amostra de corrente para cálculo RMS |
| V_i | Amostra de tensão para cálculo RMS |
| S | Potência Aparente |
| Q | Potência Reativa |
| FP | Fator de potência |
| V_{\emptyset} | Tensão trifásica |
| V_{1N} | Tensão de fase 1 |
| V_{2N} | Tensão de fase 2 |
| V_{3N} | Tensão de fase 3 |
| P_{\emptyset} | Potência ativa trifásica |
| Q_{\emptyset} | Potência reativa trifásica |
| P_{\emptyset} | Potência aparente trifásica |
| FP_{\emptyset} | Fator de potência trifásico |
| V_F | Tensão Fundamental |
| V_h | Tensão harmônica |

SUMÁRIO

| | |
|---|-----------|
| AGRADECIMENTOS..... | IV |
| 1 INTRODUÇÃO..... | 12 |
| 1.1 Objetivo | 14 |
| 1.2 Estrutura da Monografia..... | 14 |
| 2 MEDIDORES DE ENERGIA ELÉTRICA | 15 |
| 2.1 Medidores Eletromecânicos..... | 15 |
| 2.2 Medidores Eletrônicos..... | 16 |
| 2.2.1 Medidor eletrônico Mult-k..... | 17 |
| 2.2.1.1 Grandezas medidas..... | 18 |
| 2.2.1.2 Interface de comunicação | 19 |
| 2.2.1.3 Medição em quatro quadrantes | 21 |
| 2.2.1.4 Instalação do Medidor..... | 22 |
| 2.2.1.5 Configuração do medidor..... | 24 |
| 3 HARDWARE | 26 |
| 4 SOFTWARE | 29 |
| 4.1 Raspbian Linux | 29 |
| 4.2 PHP..... | 29 |
| 4.3 MySQL..... | 29 |
| 4.4 HTML..... | 29 |
| 4.5 Bootstrap..... | 30 |
| 4.6 Apache..... | 30 |
| 4.7 HTTP..... | 30 |
| 4.8 Desenvolvimento..... | 30 |
| 4.8.1 Aquisição dos dados | 30 |
| 4.8.2 Armazenamento dos dados | 31 |
| 4.8.3 Apresentação dos dados..... | 32 |
| 5 TESTES E RESULTADOS | 35 |
| 5.1 Aquisição de dados e conversão numérica | 35 |
| 5.2 Interface <i>Web</i> | 36 |
| 6 CONCLUSÕES E TRABALHOS FUTUROS..... | 41 |
| 7 REFERÊNCIAS BIBLIOGRÁFICAS | 43 |

| | |
|---|-----------|
| APÊNDICE A – EQUAÇÕES UTILIZADAS INTERNAMENTE PELO MEDIDOR PARA CÁLCULO DAS GRANDEZAS..... | 45 |
| APÊNDICE B – FUNÇÃO PRINCIPAL DA BIBLIOTECA PHPSERIAL | 49 |
| APÊNDICE C – QUADRO DE REGISTRADORES DO MEDIDOR MULT-K..... | 55 |
| APÊNDICE D – CÓDIGO PHP DA CONVERSÃO IEEE-754 HEXADECIMAL PARA DECIMAL | 57 |
| APÊNDICE E – CÓDIGO DA PÁGINA INICIAL..... | 58 |
| APÊNDICE F – CÓDIGO DA OPÇÃO LEITURAS | 60 |
| APÊNDICE G – CÓDIGO DA OPÇÃO TABELA | 72 |
| APÊNDICE H – CÓDIGO PARA GERAR GRÁFICOS..... | 89 |

1 INTRODUÇÃO

A energia elétrica é um produto fundamental nos mais diversos setores da sociedade. Por se tratar de um produto, há uma busca pela diminuição de gastos e uso mais adequado do mesmo. Grande parte de instituições e empresas, visando implementar medidas energéticas eficazes, otimizar investimentos, evitar custos adicionais na conta de energia e se beneficiar de tarifas menores, adquirir tecnologias que auxiliem na gestão da energia elétrica. Mas para que ocorra esse gerenciamento, deve ser conhecido o próprio consumo através de históricos de dados, monitoramento em tempo real do fluxo de energia. Ou seja, somente através desses dados, é que se torna possível o controle do consumo geral, evitando-se desperdícios e, pagamento de multas, dentre outros.

Algumas empresas como o Grupo Genergia oferecem uma solução comercial para gestão de energia elétrica, via plataforma *Web*, com dispositivos que são conectados ao medidor de energia elétrica e fornecem dados de consumo, controle de cargas da instalação, correção de fator de potência, alarmes via *email*, dentre outras funcionalidades (GRUPO GENERGIA, 2017).

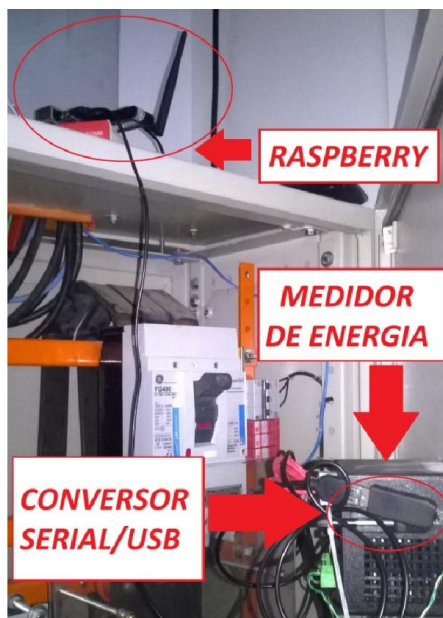
Outra solução comercial encontrada para monitoramento de energia elétrica é a oferecida pela empresa WM Energia, que é uma plataforma integradora de todos os dados de consumo energético, fornecendo e possibilitando a análise de dados de consumo energético em tempo real, disponibilizando ao usuário informação e conhecimento relevantes para a tomada de decisões com relação à gestão energética (WM ENERGIA, 2017).

Dada a importância do uso mais eficiente e racional da energia é que foi idealizado este projeto, que consiste em criar um sistema de monitoramento *online* de grandezas elétricas do Centro Tecnológico XI do Campus Goiabeiras da Universidade Federal do Espírito Santo (UFES). O projeto consiste no gerenciamento das grandezas mensuradas, desde a aquisição até a exibição através de uma página *Web*. As grandezas são obtidas através de um medidor eletrônico de energia instalado no prédio citado.

A Figura 1 mostra os equipamentos necessários ao monitoramento. Um microcomputador, *Raspberry*, está conectado ao medidor eletrônico via adaptador serial/USB e é responsável por todo o sistema de monitoramento. Nele ficam embarcados todos os *softwares*, serviços,

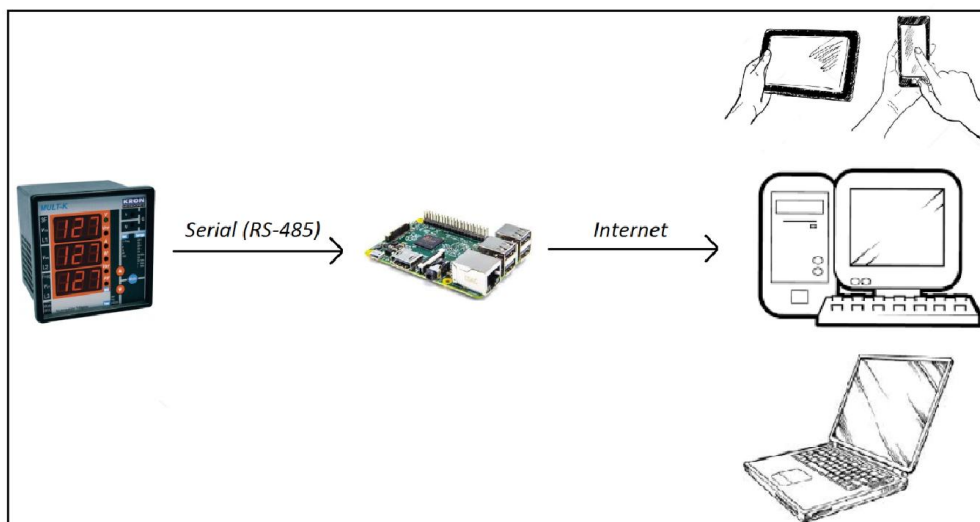
códigos, para o gerenciamento dos dados relativos às grandezas elétricas. A Figura 2 representa um diagrama com o fluxo de dados do sistema.

Figura 1 – Equipamentos para o monitoramento.



Fonte: Produção do próprio autor.

Figura 2 – Diagrama do sistema.



Fonte: Produção do próprio autor.

1.1 Objetivo

Implementar o monitoramento online de grandezas elétricas com um sistema compacto, de baixo custo, do Centro Tecnológico XI do Campus Goiabeiras da Universidade Federal do Espírito Santo.

1.2 Estrutura da Monografia

No capítulo 2 descreve-se as características e funcionamento dos tipos de medidores de energia elétrica existentes no mercado, bem como a instalação e configuração do medidor eletrônico utilizado. Já nos capítulos 3 e 4, serão apresentadas as características do *hardware*, *softwares* utilizados, respectivamente, bem como as funcionalidades e desenvolvimento do sistema. Em seguida no capítulo 5 serão apresentados testes e resultados do sistema desenvolvido para demonstrar o monitoramento proposto. Por fim, no Capítulo 6 apresentam-se as conclusões relativas ao projeto e futuras propostas de trabalho.

2 MEDIDORES DE ENERGIA ELÉTRICA

São dispositivos utilizados pelas concessionárias (entidade fornecedora de energia elétrica) para medir o consumo de energia elétrica, possibilitando com a apuração do consumo, emissão de faturas para cobrança do serviço prestado. Os dados de consumo não são importantes somente para as distribuidoras, são de suma importância para os clientes. Estes podem através do seu histórico de medições, fazer uma análise para que consumam energia elétrica de forma mais eficiente e reduzam suas faturas.

Atualmente são comercializados basicamente dois tipos de medidores de energia no Brasil, os eletromecânicos (analógicos) e os eletrônicos (digitais).

2.1 Medidores Eletromecânicos

Inicialmente eram utilizados somente medidores eletromecânicos (Figura 3), que possuem robustez e baixo custo e usam um sistema totalmente analógico. No Brasil, a maioria das unidades consumidoras, principalmente as residenciais e aquelas atendidas em baixa tensão, ainda é faturada tomando-se por base os valores de energia ativa registrados por medidores eletromecânicos (MÍNGUEZ, 2007).

Figura 3 – Medidor eletromecânico de energia elétrica Nansen.



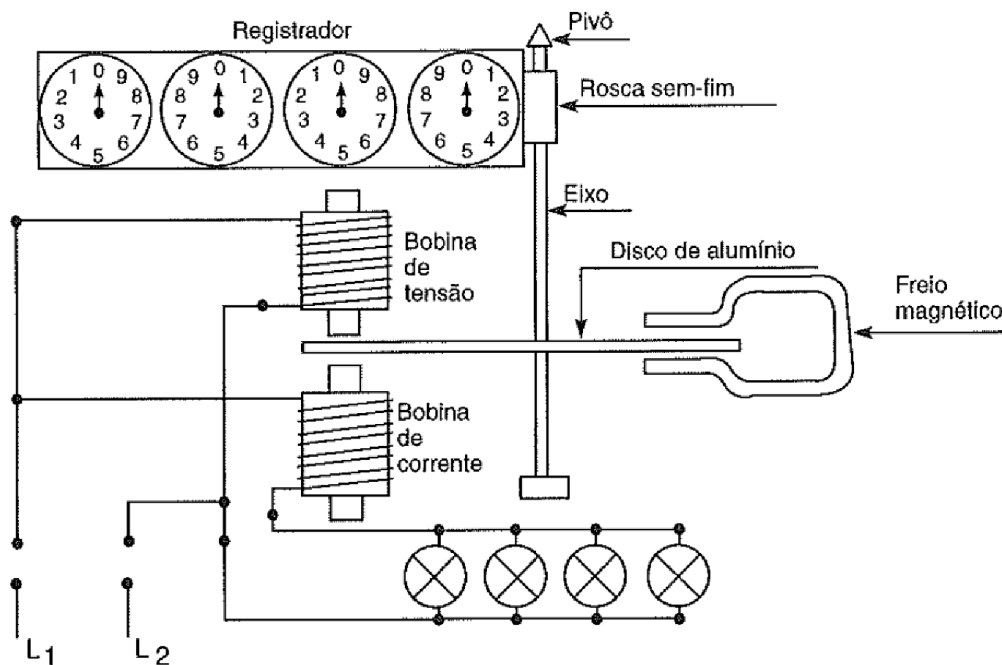
Fonte: NANSEN (2017).

O princípio de funcionamento do medidor eletromecânico é baseado no fenômeno da indução eletromagnética através de duas bobinas: bobina de tensão composta por muitas espiras e bobina de corrente, composta por poucas espiras. São gerados dois campos eletromagnéticos

que se combinam e fazem com que um disco metálico de alumínio gire. Solidário ao disco existe um eixo em conexão com uma rosca sem fim, que provoca a rotação dos registradores, realizando a leitura da energia ativa consumida. Um ímã permanente age como freio eletromagnético no disco, compensando variações bruscas de tensão ou corrente (CREDER, 2013).

As partes que compõem o medidor eletromecânico estão ilustradas na Figura 4.

Figura 4 – Composição de um medidor eletromecânico.



Fonte: CREDER (2013).

2.2 Medidores Eletrônicos

Os medidores eletrônicos (Figura 5) comercializados no Brasil são basicamente compostos por circuitos integrados dedicados. Realizam a leitura das variáveis através de amostragem em pequenos intervalos de tempo utilizando conversor analógico para digital (A/D). O processamento desses equipamentos é rápido o suficiente para proverem dados em tempo real, bem como transmissão via rede ou algum outro tipo de comunicação. O funcionamento básico desse dispositivo pode ser representado pelo diagrama de blocos da Figura 6. Os transdutores de tensão e corrente obtêm os sinais de entrada do ponto de medição e os mesmos são tratados e enviados à entrada de um módulo multiplicador, onde se obtém a potência instantânea.

Passando-se o sinal de potência calculado por um módulo integrador, que determina o valor da energia medida. Por fim, esses sinais são armazenados em um módulo registrador.

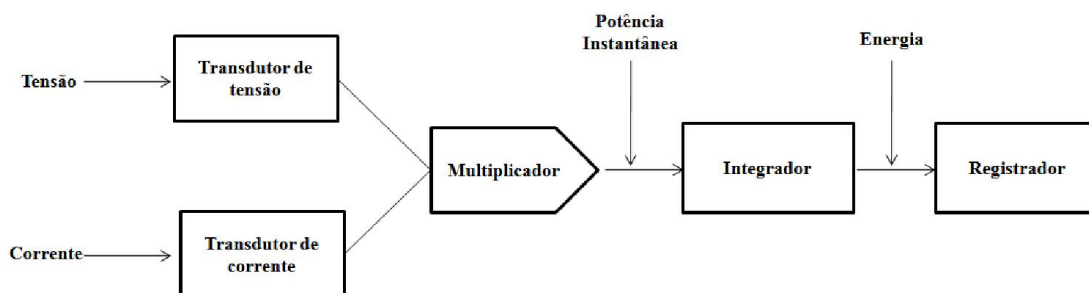
Com implementações, além da potência ativa, pode-se medir potências reativa e aparente, demanda máxima, fator de potência, distorção harmônica total (DHT), dentre outras grandezas. E ainda, tem-se a vantagem do uso de memória de massa para registro do consumo com informação de data e hora, o que permite a análise temporal dessas grandezas medidas.

Figura 5 – Medidor eletrônico de energia elétrica Kron.



Fonte: KRON MEDIDORES (2016).

Figura 6 – Diagrama de blocos que representa o funcionamento do medidor eletrônico de energia elétrica.



Fonte: MÍNGUEZ (2007).

2.2.1 Medidor eletrônico Mult-k

Medidor trifásico digital microprocessado (Figura 5) fabricado pela empresa Kron Instrumentos Elétricos Ltda, para instalação em porta de painel. Pode ser utilizado em instalações prediais, industriais, até subestações ou em qualquer aplicação que envolva medição de grandezas elétricas em sistemas com corrente alternada (CA).

Principais características técnicas do medidor Mult-k utilizado neste projeto:

- Alimentação externa: 12, 24 ou 48Vcc/120-220Vca;
- Consumo interno: menor que 0,5VA;
- Tensão de entrada (medição): 20 a 500Vca (fase-fase) (uso de transformador de tensão quando ultrapassar valor de 500V);
- Frequência (medição): 44 a 72Hz;
- Corrente de entrada (medição): 1 ou 5Aca (uso de transformador de corrente quando ultrapassar valor de 5A);
- Display de 7 segmentos;
- Interface serial: RS-485 com protocolo MODBUS-RTU até 57600bps.

2.2.1.1 Grandezas medidas

O medidor permite a medição de até 44 grandezas elétricas (Quadro 1). Suas medições são *TRUE RMS* (tipo de medição onde é levada em consideração a distorção presente em determinada forma de onda) feitas no sistema de quatro quadrantes, permitindo sua utilização em sistemas de cogeração de energia elétrica. As leituras das grandezas podem ser feitas localmente através do display de sete segmentos ou, remotamente pela interface serial RS-485.

Quadro 1 – Grandezas elétricas medidas pelo medidor Kron.

| | Grandeza | Unidade | Tipo de medição | D i s p l a y | R S - 4 8 5 |
|--------------|-------------------|---------|--|---------------------------------|----------------------------|
| Instantâneas | Tensão | V | Trifásica/bifásica, Fase-neutro (sistema estrela/monofásico) ou Fase-fase (sistema estrela ou delta) | x | x |
| | Corrente | A | Trifásica ou bifásica/Por fase | x | x |
| | Potência Ativa | W | Trifásica ou bifásica/Por fase | x | x |
| | Potência Reativa | Var | Trifásica ou bifásica/Por fase | x | x |
| | Potência Aparente | VA | Trifásica ou bifásica/Por fase | | x |

| | | | | | |
|--------------|---------------------------|-------|--|---|---|
| | Fator de Potência | - | Trifásico ou bifásico/Por fase | x | x |
| | Frequência | Hz | Fase R | x | x |
| | DHT | % | Por fase | x | x |
| Acumulativas | Energia Ativa Positiva | kWh | Trifásica, bifásica ou monofásica, dependendo do circuito que está sendo medido. | x | x |
| | Energia Ativa Negativa | kWh | | x | x |
| | Energia Reativa Positiva | kVarh | | x | x |
| | Energia Reativa Negativa | kW | | x | x |
| | Demanda Média Ativa | kVA | | x | x |
| | Demanda Média Aparente | kW | | x | x |
| | Demanda Máxima Ativa | kW | | x | x |
| | Demanda Máxima Aparente | kVA | | x | x |
| | Máxima Tensão Trifásica | V | | x | x |
| | Máxima Corrente Trifásica | A | | x | x |

Fonte: KRON MEDIDORES (2016) (Adaptado).

Internamente, para o cálculo das grandezas elétricas, o medidor utiliza as equações apresentadas no Apêndice A. Para cálculos relacionados à energia segue o procedimento descrito na Figura 6.

Informações sobre demais características desse medidor, podem ser acessadas no manual fornecido pelo fabricante no site <http://www.kronweb.com.br/br/produto-6-mult-k-multimedidor-sem-memoria-de-massa>. Vale também ressaltar que informações detalhadas sobre o circuito eletrônico interno do medidor, para saber como são feitos os cálculos das grandezas, não foram cedidas pelo suporte técnico do fabricante. Dessa forma, não é possível determinar se o equipamento não faz isolamento galvânica entre a medida e a leitura dos dados, geralmente feito por opto-acopladores, o que daria ao equipamento maior segurança.

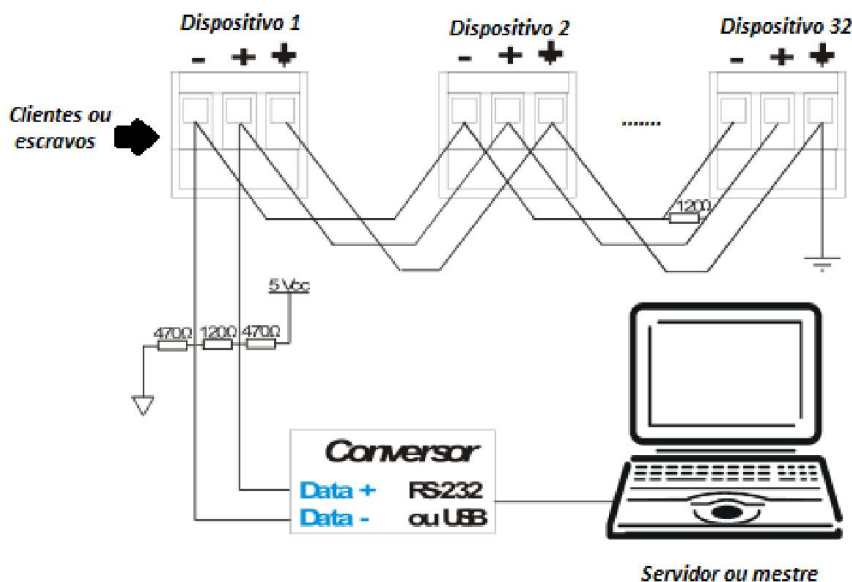
2.2.1.2 Interface de comunicação

A conexão RS-485 é uma interface de comunicação que opera em linhas diferenciais, capaz de se comunicar com até 32 dispositivos sem necessidade de amplificadores de sinal. Opera a dois fios (par trançado) e com comunicação half-duplex, para leitura e parametrização remota do instrumento. Seu funcionamento pressupõe a presença de um servidor ou mestre que se comunicará com seus clientes ou escravos, como mostrado na Figura 7.

Essa interface serial permite conexões de dados até uma distância máxima de 1200 metros e uma velocidade de 10 Mbps, no entanto, o comprimento máximo e a taxa de *bits* máxima não

podem ser alcançados ao mesmo tempo. Para par de cabos trançados de diâmetro 24 AWG, a taxa máxima é de aproximadamente 90 kbps. Para conexão a 10 Mbps a distância é de 6 metros. (MACKAY, 2004).

Figura 7 – Configuração de uma rede com padrão RS-485.



Fonte: KRON MEDIDORES (2016) (Adaptado).

Uma rede configurada com o padrão RS-485 possui custos bem menores quando comparado ao *Ethernet*, *Fieldbus*, outros. Opera corretamente na presença de tensões diferenciais de terra e em ambientes eletricamente ruidosos. Possui uma configuração flexível, permitindo que o usuário implemente seu próprio protocolo de comunicação ou use protocolos abertos já definidos.

O protocolo de comunicação utilizado nessa interface do medidor é o *MODBUS* no modo *Remote Terminal Unit (RTU)*. Esse protocolo foi desenvolvido para sistemas de controle de processos para comunicação entre dispositivos do tipo mestre-escravo ou cliente-servidor, permitindo a conexão de até 247 dispositivos em uma mesma rede. Sua estrutura é simples, flexível e de código aberto. A estrutura típica (Quadro 2) de uma mensagem possui os campos:

- Endereço: byte de informação, contém o endereço do dispositivo que a mensagem está sendo dirigida (mensagem de solicitação); o endereço do dispositivo que está enviando a resposta (mensagem de resposta);

- Função: byte de informação, contém a função que o escravo deve executar; como resposta, um eco ao pedido dizendo se pode ou não executar a tarefa;
- Dados: neste campo estão presentes os dados que foram solicitados pelo campo função, variando de acordo com solicitado;
- Checagem de Erro: contém bytes para verificação de erro durante a transmissão.

O modo *RTU* do *MODBUS* é uma mensagem transmitida de 8 bits (no mínimo), pelos campos, que conterà dois caracteres hexadecimais de 4 bits.

Quadro 2 – Estrutura do quadro de mensagem via *MODBUS*.

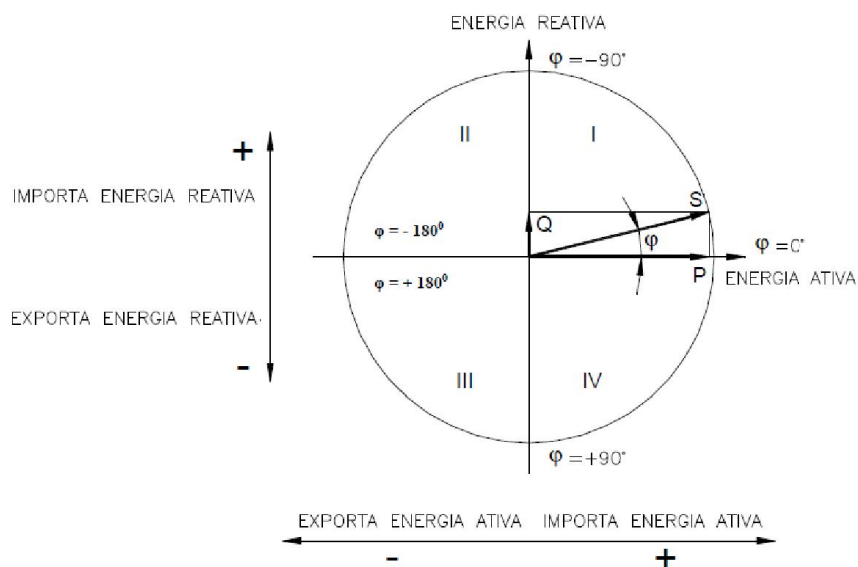
| Endereço | Função | Dados | Checagem de erro |
|-----------------|---------------|--------------|-------------------------|
| 8 bits | 8 bits | N x 8 bits | 16 bits |

Fonte: MACKAY (2004) (Adaptado).

2.2.1.3 Medição em quatro quadrantes

Para a medição da energia num sistema de quadrantes, a Figura 8 a seguir ilustra como o fluxo de energia medido é representado nesse sistema. Quando as leituras de energia reativa são positivas (quadrantes 1 e 2) há recebimento da mesma. Leituras negativas (quadrantes 3 e 4) há envio da mesma. Para energia ativa, leituras positivas (quadrantes 1 e 4) indicam recebimento da mesma e, leituras negativas (quadrantes 2 e 3), indicam envio da mesma. Essas leituras são importantes para os casos em que o medidor for instalado em um sistema que há recebimento e geração de energia.

Figura 8 – Representação da energia medida em quadrantes.



Fonte: INMETRO (2017) (Adaptado).

2.2.1.4 Instalação do Medidor

O medidor encontra-se instalado na porta do painel de um quadro de distribuição principal (quadro de entrada) de energia, com alimentação trifásica em estrela. O quadro está localizado no Centro Tecnológico XI (CT XI), Campus Goiabeiras da Universidade Federal do Espírito Santo (UFES).

Figura 9 – Instalação do medidor



Fonte: Produção do próprio autor.

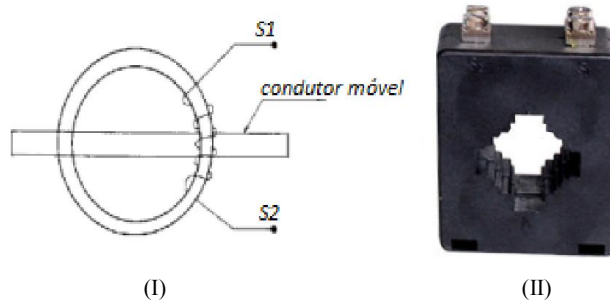
Na instalação do medidor foi necessária a utilização de transformadores de corrente (TC), dispositivos que reproduzem no seu circuito secundário uma amostra, geralmente 1 ou 5A, da corrente nominal que circula no primário. Isto foi necessário, pois a corrente consumida pela instalação do prédio ultrapassa o limite máximo (5A) de entrada do medidor. O TC (Figura 10) utilizado, modelo HB 603, é fabricado pela empresa ABB e possui relação de transformação, entre circuitos primário e secundário, de 400:5A. Esse TC se caracteriza por ser do tipo janela, mas não de núcleo dividido.

A instalação dos TC's não foi feita durante a execução deste projeto, já haviam sido previamente instalados. Na execução desse projeto de fim de curso, constatou-se, entretanto, os seguintes problemas na instalação:

- o TC da fase R foi instalado com polaridade invertida, o que acarreta leitura negativa da potência ativa;
- apesar do TC da fase T encontrar-se instalado, o medidor não consegue fazer a leitura do sinal de corrente e não é possível concluir se o problema encontra-se no TC ou no medidor Kron.

Por uma questão de segurança, não foi realizada nenhuma manobra nos TC's para corrigir os erros detectados, uma vez que seria necessário “desenergizar” todo o prédio do CT XI, assim como o ramal de entrada do mesmo, evitando-se os problemas de geração de arco voltaico ao “abrir” a conexão de um TC.

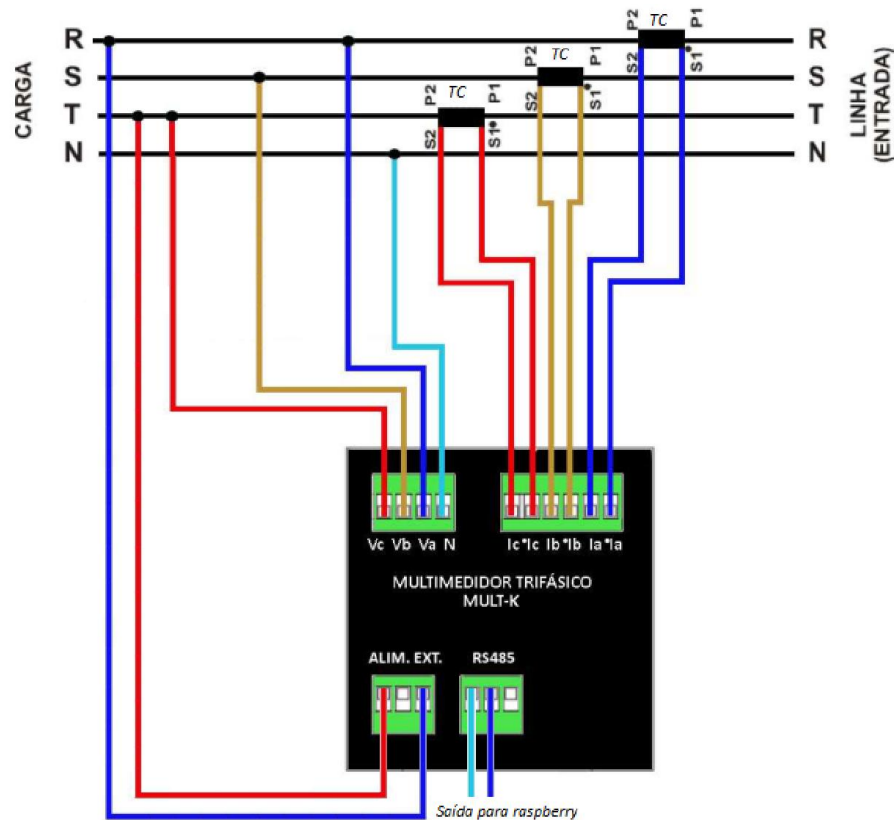
Figura 10 – Transformador de corrente HB 603 não articulado.



Fontes: (I) UTFPR (2017) Adaptado; (II) ABB (2017).

Durante a execução deste projeto foi finalizada a instalação do medidor, que consistiu em conectar cada uma das fases (R, S, T), o neutro (N) e a alimentação externa (fases R e T) ao medidor. Adicionalmente, fez-se a ligação da interface serial de saída do medidor ao *Raspberry Pi*. O esquemático simbólico da instalação pode ser visto na Figura 11 a seguir.

Figura 11 – Esquemático da instalação final do medidor Kron.



Fonte: KRON MEDIDORES (2016) (Adaptado).

2.2.1.5 Configuração do medidor

Para que a leitura dos dados feita pelo medidor estivesse correta, foi feita a parametrização, através do display de 7 segmentos, de alguns elementos essenciais para o funcionamento do mesmo. São os seguintes:

- Relação do TP (transformador de potencial);
- Relação do TC;
- Constante TL (tipo de ligação);
- Constante TI (tempo de integração);
- Dados do RS-485.

Na parametrização da relação de TP não foi usado nenhum valor, pois como visto na instalação final do medidor não foi feito uso desse tipo de transformador. Já para a parametrização do TC foi inserido o valor de 400:5A (dito antes). Foi constatado, durante a

execução desse projeto, que essa constante havia sido carregada previamente de forma equivocada no medidor, o que fornecia medidas errôneas de corrente e demais grandezas. Para a constante TL, o usuário precisa informar o tipo de instalação elétrica (monofásica, bifásica e trifásica) em que o medidor está sendo instalado. Para este projeto, como dito antes, temos uma instalação trifásica em estrela, logo o valor de TL foi inserido para esse tipo de instalação. A constante TI refere-se ao tempo de integração usado para cálculo de demanda. O padrão usado por norma pelas distribuidoras de energia é de, no mínimo 15 minutos e máximo de 60 minutos. O valor inserido foi 15 minutos. Finalmente os dados para interface serial, que foram inseridos com valor de 9600bps para a velocidade de transmissão e, 8 bits com bit de parada no valor 1 e sem paridade, para as mensagens de dados.

3 HARDWARE

O *Raspberry Pi* foi originalmente desenvolvido para inspirar jovens programadores a aprimorar seus talentos e a ganharem um lugar no curso de computação da Universidade de *Cambridge*. Mas quando saiu a notícia sobre um minúsculo computador por menos de 30 libras, todos queriam ter um em suas mãos. (COMPUTER SHOPPER, 2015)

Por se tratar de um microcomputador de baixo custo, fácil acessibilidade, passou a ser utilizado para muitas outras finalidades, como monitoramento, controle de cabines de informação, implementação de servidores dedicados de pequeno porte, dentre outros.

O primeiro dispositivo comercializado possuía 256 MB de memória RAM, processador com único núcleo com clock de 700 MHz, sendo que atualmente, já estão sendo comercializados modelos com 1 GB de memória RAM e processador com quatro núcleos com clock de 1,2 GHz.

As especificações do *Raspberry Pi 1 Model B+* utilizado neste projeto são:

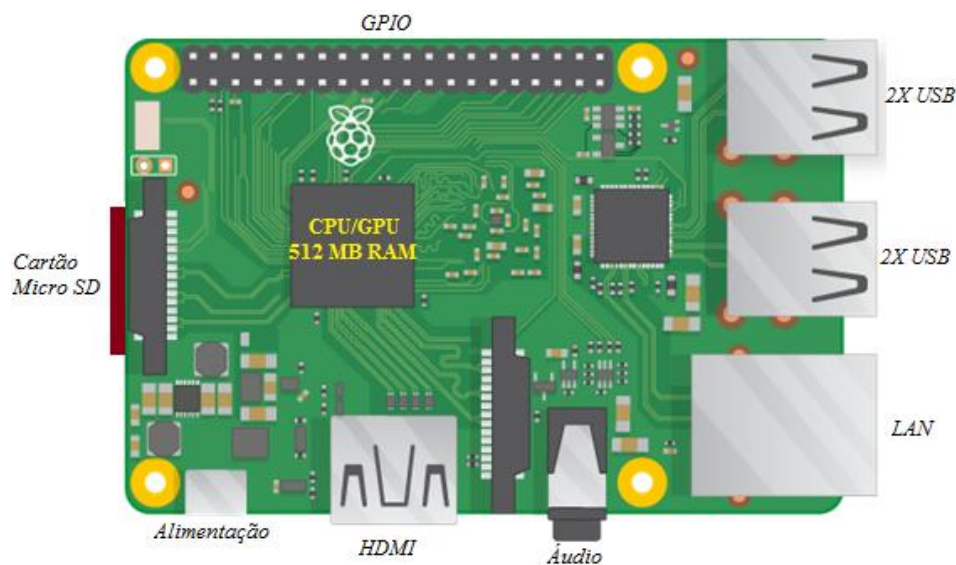
- CPU ARM1176JZF-S core de 700 MHz;
- Co-processador multimídia *Dual Core Videocore IV*;
- 512 MB de memória RAM;
- *Ethernet* 10/100;
- 4 portas USB;
- *GPIO*;
- Saída de vídeo HDMI;
- Alimentação 5Vcc 1200mA;
- Encapsulamento em uma placa de dimensões 85,60mm x 56mm x 21mm.

A escolha desse dispositivo para o projeto foi devido às características, citadas acima, de fácil acessibilidade, baixo custo (em torno de 90 reais) e dimensões do mesmo. Um dispositivo que poderia ter sido utilizado seria o *Arduíno*, pois apresenta essas mesmas características, porém possui *hardware* bem limitado. Possui nos modelos mais avançados processador com 84 MHz de *clock* e memória RAM de 96 Kb, o que seria inviável para as necessidades deste projeto. Como o site desenvolvido explora a geração de gráficos, a adição de mais gráficos e aumento da base de dados iria esbarrar na limitação, isso sem contar no poder de processamento, capacidade de embarcar um sistema robusto como o *Linux*.

Poderia ter utilizado ainda dispositivos que possuem processamento parecido ou até melhor que o *Raspberry Pi*, como por exemplo os microcomputadores *Cubieboard*, *Beagle Bone*, *Udoo*. Estes atenderiam as necessidades deste projeto, mas não foram utilizados devido aos custos maiores apresentados pelos mesmos em relação ao *Raspberry*. O preço médio desses dispositivos giram em torno de 200 reais ou mais.

Os periféricos do *Raspberry Pi* estão representados na Figura 12, que mostra também a localização de cada um deles na placa.

Figura 12 – Especificações do *Raspberry Pi*.



Fonte: RASPBERRY PI (2017) (Adaptado).

O processamento do *Raspberry* está em um único chip (sistema *SOC – System on a chip*), este contém a memória RAM, unidades de processamento central e gráfico. Usa arquitetura *ARM*, a mesma encontrada em muitos dispositivos do cotidiano como *tablets*, *smartphones*, televisores, *eReaders*, dentre outros.

Dentre as características dessa arquitetura pode-se destacar:

- Processador de 16, 32 e 64 bits (apenas alguns processadores);
- 16 registradores de uso geral;
- Conjunto de instruções extensível com uso de co-processadores;
- Instruções básicas similares ao 6502;
- Instruções de 3 endereços;

- Capacidade de executar instruções de 16 bits usando a arquitetura *Thumb* (instruções codificadas para aumentar o desempenho do processador em certas aplicações);
- Baixo consumo de energia;
- Tamanho do núcleo reduzido;
- Até 16 co-processadores lógicos.

O *Raspberry* não possui um disco rígido, logo os arquivos do sistema operacional e arquivos de usuário ficaram armazenados em um cartão micro SD de 8 GB. Sua comunicação com o medidor de energia elétrica neste projeto foi feita pela porta USB, por meio de um conversor USB/Serial, que foi conectado à interface serial do medidor. Para conexão com a *Internet*, foi conectada ao dispositivo um adaptador *WIFI* via porta USB.

4 SOFTWARE

Neste capítulo serão descritos todos os *softwares*, linguagens e protocolo utilizados para o desenvolvimento do sistema, bem como a descrição do funcionamento do mesmo.

4.1 Raspbian Linux

Para gerenciar o hardware do *Raspberry*, foi utilizado o sistema operacional *Raspbian Linux*, que é uma versão do *Debian Linux* para processadores que usam arquitetura *ARM*, disponível para download de forma gratuita no site do fabricante.

4.2 PHP

Para aquisição e tratamento dos dados fornecidos pelo medidor, foi utilizada a linguagem de programação *PHP* (*Hypertext Processor*), que é uma linguagem adequada para o desenvolvimento *web* e de criação de *scripts* inseridos em *HTML*, permitindo a rápida criação de páginas dinâmicas de internet. É uma linguagem de programação aberta que possui diversos materiais disponíveis em vários formatos, o que facilita sua aprendizagem. Além de possuir diversas bibliotecas disponíveis, o que facilita o desenvolvimento de sistemas.

4.3 MySQL

No armazenamento dos dados (banco de dados) foi utilizado o *MySQL*, que é um sistema de gerenciamento de banco de dados de código aberto, que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface. Funciona nos mais diversos sistemas operacionais, sendo reconhecido por sua robustez e desempenho utilizando hardware bem restrito.

4.4 HTML

Na apresentação dos dados, feita por uma página *Web*, foi utilizado *PHP* e *HTML* (do inglês *HyperText Markup Language*). Esta é uma linguagem de marcação usada para desenvolvimento de páginas da *internet*. Os elementos da linguagem são incluídos por *tags*, e o conteúdo da página (texto, tabelas, imagens, dentre outros.) é inserido entre elas.

4.5 Bootstrap

Para auxiliar no desenvolvimento do layout da página utilizou-se dessa ferramenta. É uma coleção de vários elementos e funções personalizáveis para otimizar projetos *Web*, usados para desenvolver temas, menus, fontes e outros da página. Possui um sistema responsivo, o que permite a adaptação da página a vários dispositivos como computadores, *smartphones*, *tablets*.

4.6 Apache

Para implementar o servidor *Web* ou *HTTP* foi utilizado o *Apache*, que é um aplicativo livre responsável por disponibilizar páginas *Web* e todos os recursos que podem ser acessados pela *internet*. É compatível com o protocolo *HTTP* e a comunicação é, por padrão, feita pela porta 80.

4.7 HTTP

Para a comunicação com o *Raspberry* é usado o protocolo *HTTP* (do inglês *Hypertext Transfer Protocol*). Este é um protocolo da camada de aplicação do modelo OSI que utiliza o protocolo TCP/IP e tem como finalidade a transferência de páginas de internet. Utiliza um modelo cliente-servidor, onde o cliente envia uma mensagem ao servidor com a requisição de um recurso e este responde a solicitação do cliente.

4.8 Desenvolvimento

Após instalação de sistema operacional, softwares, bibliotecas no *Raspberry* e programação com as linguagens citadas, o desenvolvimento do sistema foi dividido em três etapas: aquisição, armazenamento e apresentação dos dados.

4.8.1 Aquisição dos dados

A aquisição de dados do medidor eletrônico feita via software, é feita pela biblioteca serial *MODBUS* em PHP, que pode ser vista no Apêndice B. Esse código faz todo o trâmite para que haja comunicação entre o medidor eletrônico e o *Raspberry*. Como variáveis de entrada, é necessário explicitar dados relativos à velocidade de comunicação, paridade e bits de início, dados e parada. Esses devem estar de acordo com os valores configurados no medidor previamente. Outro dado importante, é indicar em qual porta do dispositivo será feita a

comunicação. Com esses dados, a porta serial (adaptador USB) do *Raspberry* é “aberta” e configurada.

Com a porta inicializada, deve-se preencher o quadro de mensagens do protocolo para informar ao cliente (medidor) o que está sendo requisitado. O quadro é preenchido e enviado com o número do cliente, função leitura e registrador a ser lido. A lista com descrição dos registradores do medidor pode ser vista no Apêndice C. Em seguida o cliente responde ao servidor (*Raspberry*) com um quadro preenchido com os dados requisitados. Estes dados são exibidos em formato hexadecimal pelo código. Para exibi-los em formato decimal, foi utilizado um programa em *PHP* (Apêndice D), pois o medidor usa como representação numérica o formato IEEE-754. Este define algumas regras de normalização a serem seguidas nas operações e representação binária em ponto flutuante, foi desenvolvido pelo IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos).

4.8.2 Armazenamento dos dados

Para que fosse criado um histórico dos dados relacionados às leituras feitas pelo medidor foi criado um banco de dados, pois esse não armazena as leituras realizadas. O banco de dados do sistema foi implementado por um servidor *MySQL*, como dito anteriormente, que fica hospedado no próprio *Raspberry*. As leituras são feitas de 15 em 15 minutos automaticamente pelo *Raspberry* como se fosse uma rotina do sistema operacional.

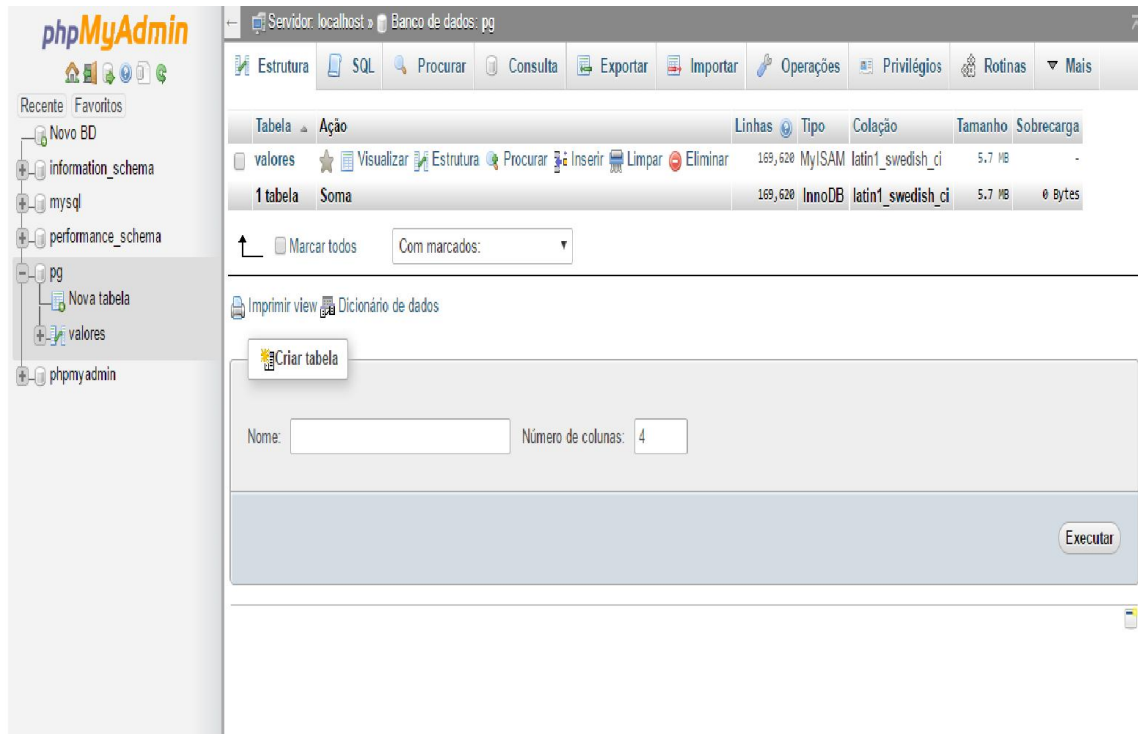
Para gerenciamento do banco de dados foi criado um usuário *root* que possui todas as permissões para configurar o banco como, inserção, exclusão, leitura de dados, dentre outros. O banco é estruturado por apenas uma tabela, como é mostrado no Quadro 3.

Quadro 3 – Estrutura da tabela do banco de dados.

| <i>root</i> | |
|------------------|---------------------------|
| id | Inteiro auto-incrementado |
| reg_id | varchar (20) |
| val | varchar (20) |
| timestmap | timestamp |

Fonte: Produção do próprio autor.

Para o gerenciamento do banco de dados foi utilizado o aplicativo *phpMyAdmin* (Figura 13), que é uma ferramenta PHP de gerenciamento de banco de dados que funciona também pela rede, permitindo assim que o banco de dados fosse gerenciado remotamente.

Figura 13 – Página de configuração do *phpMyAdmin*.

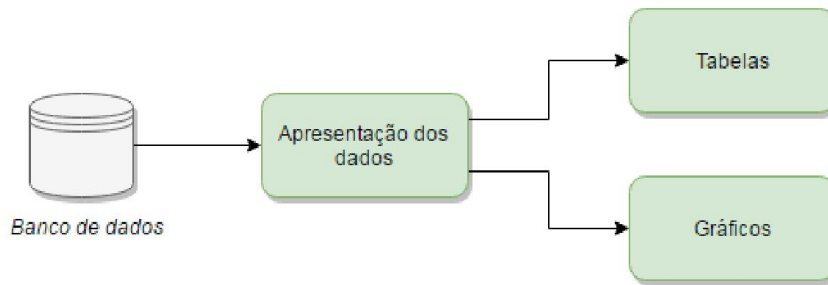
Fonte: Produção do próprio autor.

4.8.3 Apresentação dos dados

Feita através de uma página *Web*, foi utilizada a linguagem HTML com auxílio do bootstrap, como dito, com scripts PHP para fornecer todas as funcionalidades necessárias para criação de uma interface. O site fica “hospedado” no *Raspberry*, que pode ser acessado por qualquer usuário que esteja utilizando a rede interna de computadores da UFES pelo endereço 172.20.25.162. Para a exibição da página *Web* o usuário se conecta ao servidor *Web* via protocolo *HTTP*, como dito antes. Para que haja essa conexão, no *Raspberry* foi instalado um adaptador USB *WIFI* (como já dito), que se conecta a um roteador *WIFI* do prédio, dando assim acesso a qualquer usuário da rede UFES.

Para apresentação via opção tabelas ou gráficos, os dados são pesquisados no banco de dados e apresentados na forma escolhida pelo usuário, como mostra o fluxograma na Figura 14.

Figura 14 – Fluxograma para apresentação via tabelas e gráficos.

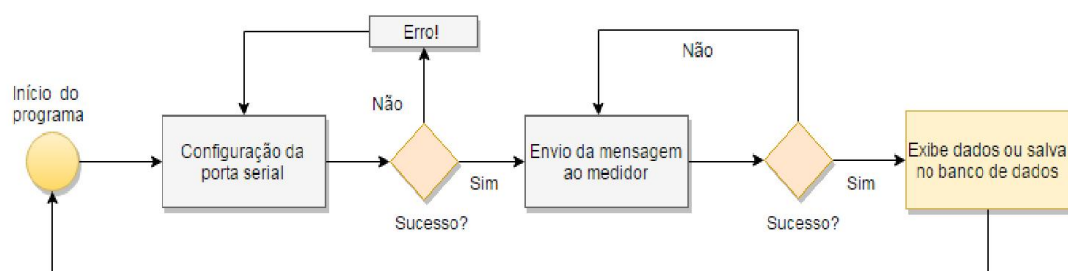


Fonte: Produção do próprio autor.

Já para apresentação dos dados via opção leituras, são feitas aquisições de dados em tempo real por software. Este tem seu diagrama de blocos exibido na Figura 15. O programa é iniciado com configurações da porta serial que, se estiver operando corretamente a mesma é “aberta” para comunicação e prossegue para envio de mensagem ao medidor ou, caso contrário, é reiniciado o processo de configuração da mesma. Em seguida é enviada mensagem ao medidor eletrônico onde, estando a mensagem recebida livre de erros, os dados são exibidos para o usuário ou salvos, se não, é analisado o que está errado com a resposta recebida e enviada nova mensagem. Os erros relacionados às configurações de porta serial e envio de mensagens foram tratados somente no início dos testes, depois foram desconsiderados devido à confiabilidade do sistema. Os erros apresentados pela porta serial foram de reconhecimento do *Raspiberry Pi* para com o conversor serial-*USB* como um dispositivo de entrada e saída de dados. Com instalação de bibliotecas adicionais para comunicação serial, esses erros foram resolvidos. Já os erros das mensagens foram relacionados à quantidade de bits de dados solicitados, a conversão dos dados recebidos e aos endereços dos registradores das grandezas elétricas no medidor. Com solicitação via *email*, ao fabricante do medidor dessas características dos dados de mensagens, foram resolvidos os problemas.

No último bloco do diagrama há duas opções para os dados, exibição ou armazenamento no banco de dados. Isso ocorre porque o software é usado tanto para leituras em tempo real quanto para salvar os dados que são lidos de 15 em 15 minutos de forma automática.

Figura 15 – Diagrama de blocos para aquisição ou exibição de dados.



Fonte: Produção do próprio autor.

5.2 Interface Web

A página inicial do site, código visto no Apêndice E, traz um texto descritivo informando ao usuário as características do projeto. Na parte superior há um menu de navegação, com as opções *Início*, *Leituras*, *Tabelas* e *Gráficos* (Figura 18).

Figura 18 – Página inicial do site desenvolvido.

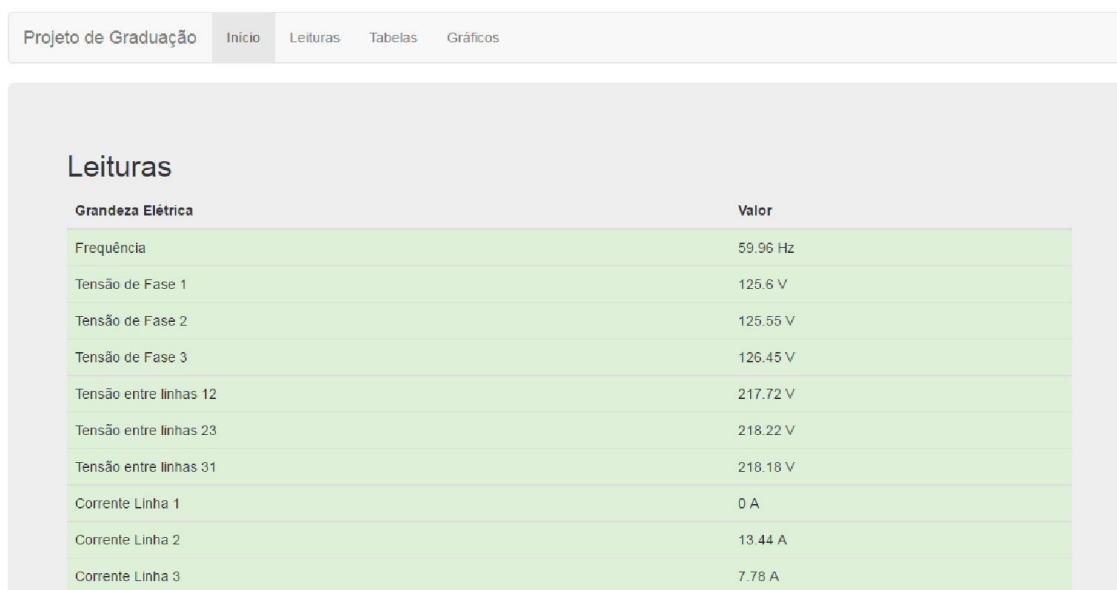


Fonte: Produção do próprio autor.

Na opção *Leituras*, código no Apêndice F, são apresentadas as medições (Figura 19), de todas as grandezas elétricas, feitas pelo medidor naquele instante. Para que essas medições sejam apresentadas em tempo real são utilizados os códigos de aquisição de dados e o de conversão numérica. A página faz uma atualização automática de 30 em 30 segundos e sua exibição é

interrompida todas as vezes que estão sendo salvas leituras (15 em 15 minutos) no banco de dados. É exibida uma mensagem ao usuário (Figura 20) e depois de registrados os dados no banco, as leituras das grandezas são apresentadas novamente.

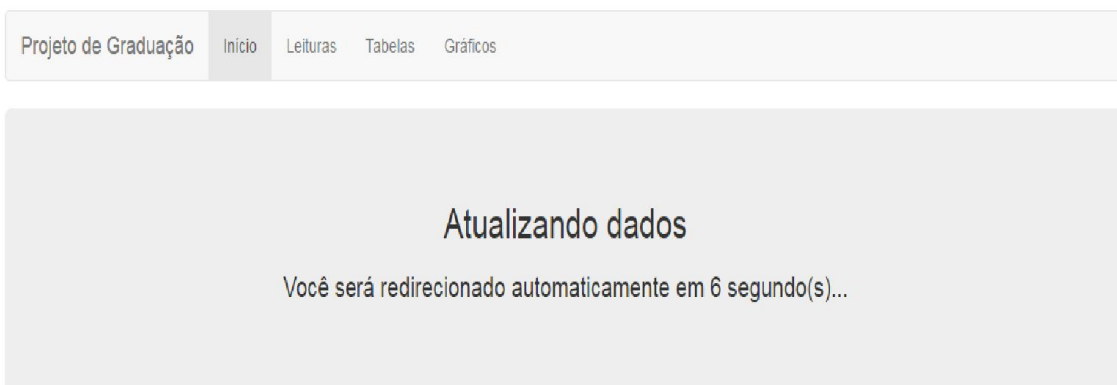
Figura 19 – Parte da página opção Leituras.



| Grandeza Elétrica | Valor |
|------------------------|----------|
| Frequência | 59.96 Hz |
| Tensão de Fase 1 | 125.6 V |
| Tensão de Fase 2 | 125.55 V |
| Tensão de Fase 3 | 126.45 V |
| Tensão entre linhas 12 | 217.72 V |
| Tensão entre linhas 23 | 218.22 V |
| Tensão entre linhas 31 | 218.18 V |
| Corrente Linha 1 | 0 A |
| Corrente Linha 2 | 13.44 A |
| Corrente Linha 3 | 7.78 A |

Fonte: Produção do próprio autor.

Figura 20 – Página de atualização de dados.



Fonte: Produção do próprio autor.

Na opção *Tabelas* (Apêndice G) é aberta uma página em que o usuário seleciona a grandeza elétrica a ser analisada, com datas e horários de interesse, para que sejam listados os valores da mesma no período selecionado (Figura 21). Para exibição desses dados, uma consulta é aberta ao banco de dados, para que sejam coletados os registros da grandeza selecionada e exibidos em seguida na página.

Figura 21 – Página da opção tabelas.

Tabela

Tipo: Corrente Linha 2 (A)

Data Inicial: 07-03-2017

Data Final: 08-03-2017

Hora Inicial: 00:00:00

Hora Final: 23:59:59

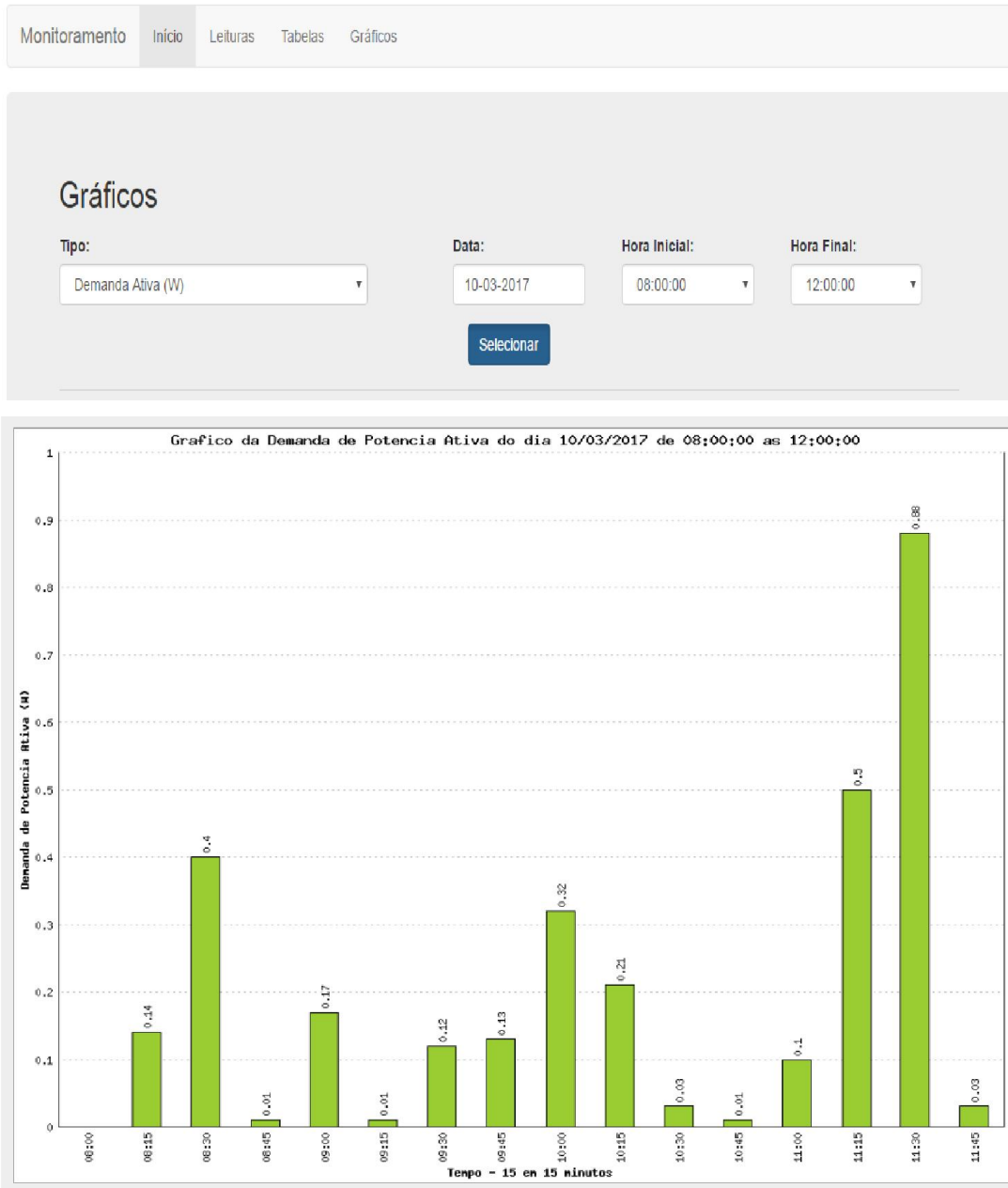
Listando: Corrente Linha 2 (A) de 07-03-2017 00:00 às 07-03-2017 23:59:59

| Data | Valor |
|------------------|-------|
| 07-03-2017 00:00 | 8.28 |
| 07-03-2017 00:15 | 8.33 |
| 07-03-2017 00:30 | 8.28 |
| 07-03-2017 00:45 | 8.26 |
| 07-03-2017 01:00 | 8.29 |
| 07-03-2017 01:15 | 8.26 |
| 07-03-2017 01:30 | 8.31 |
| 07-03-2017 01:45 | 8.3 |
| 07-03-2017 02:00 | 8.29 |
| 07-03-2017 02:15 | 8.43 |
| 07-03-2017 02:30 | 8.27 |

Fonte: Produção do próprio autor.

Por último, a opção *Gráficos* (Apêndice H), que exibe gráficos (Figura 22) com escolha idem opção anterior. A diferença na exibição dos dados é que foi utilizada a biblioteca *phplot*, uma biblioteca *PHP* de código aberto que oferece a geração de variados gráficos de forma dinâmica e rápida, diminuindo assim o tamanho do código feito.

Figura 22 – Página da opção gráficos.



Fonte: Produção do próprio autor.

Os dados que são apresentados na página *Web*, em formas de tabela e gráfico, começaram a ser armazenados no banco de dados a partir do dia 01 de fevereiro de 2017. E a partir dessa data, já com vários serviços implementados, o *Raspberry* teve sua temperatura monitorada diariamente para garantir um bom funcionamento do dispositivo. Em dias de temperatura mais elevadas, a temperatura do mesmo não passava de 47 graus centígrados. Sendo que, em

um ambiente com ar condicionado antes da instalação para coleta dos dados, o mesmo atingia 42 graus centígrados.

Vale destacar que grande parte das medições exibidas pelo site não condizem com a realidade. Isso se deve à instalação do TC da fase T ter sido feita com polaridade invertida, como dito antes. Dessa forma o medidor interpreta que a instalação está fornecendo (potência ativa negativa) corrente para a rede elétrica. Outro fato é o não recebimento do sinal do TC da fase R pelo medidor, como dito. Esses dois problemas afetam diretamente todos os valores de grandezas relacionadas às correntes medidas dessas fases, pois corrente e tensão de cada fase são as bases para cálculos das outras grandezas. Isso pode ser visto nas equações do Apêndice A, pois as equações são dependentes umas das outras e, no diagrama de blocos da Figura 6 apresentado sobre o funcionamento básico de um medidor eletrônico.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste projeto foi desenvolvido o monitoramento *online* de grandezas elétricas de uma instalação elétrica através de uma página *Web*. Esse se torna cada vez mais necessário nos mais diversos tipos de instalações, pois através do mesmo, dados podem ser analisados em tempo real para tomada de decisões rápidas ou mesmo análise do histórico de consumo da instalação para um uso mais eficiente da energia elétrica.

Com os testes realizados foi verificado que as leituras dos dados feitas através da porta serial estavam de acordo com as fornecidas pelo medidor eletrônico. A aquisição, armazenamento e apresentação dos mesmos, tudo feito em um microcomputador de baixo custo foi satisfatório. O acesso à apresentação dos dados via página *Web* pode ser feito em qualquer lugar em tempo real, desde que o usuário tenha acesso à rede de computadores da UFES. Com ajuda de tabelas e gráficos, a análise e entendimento desses dados se tornam mais fáceis e rápidos.

Para projetos futuros, podem ser analisados os dados das leituras no banco de dados para serem tomadas decisões relativas a um consumo mais eficiente da energia nesta instalação, desde que as devidas reparações dos TC's sejam feitas. Para criação de um sistema mais robusto, pode ser trocado o *hardware* utilizado por um dispositivo de maior velocidade e capacidade de processamento. Ou ainda, fazer do *Raspberry* apenas um módulo de tratamento que envia dados para um servidor externo, o qual ficará responsável por armazenamento e apresentação dos dados. Na rede mundial de computadores a segurança de um site é fundamental, no processo para tornar o sistema mais robusto, deve ser considerada a implantação de um sistema de autenticação de usuário para a liberação do acesso às informações somente a quem for de interesse, tornando o sistema mais seguro e confidencial quanto ao funcionamento.

A *internet* vive uma nova fase, onde milhares de “coisas” do nosso dia a dia processam informações e se conectam no mundo virtual para tornar o mundo real mais eficiente seguro e sustentável através da tecnologia. A essa nova fase dá-se o nome de internet das coisas ou *IoT* (do inglês: *Internet of Things*). A partir dessa idéia, pode ficar como proposta a criação de um monitoramento mais específico, que consiste em observar o consumo de cada equipamento da instalação e saber o que cada um consome em tempo real, exibindo em aplicativo o valor

gasto na conta de energia. Pode-se também ligar ou desligar os equipamentos em horários programados para economia de energia.

7 REFERÊNCIAS BIBLIOGRÁFICAS

ABB. **Ficha técnica – N0036**. Disponível em:

<<http://www.eletronicalucense.it/Product/Datasheet/ABB/N003650500H1.pdf>>. Acesso em: 20 fev. 2017.

CEFET-RJ. **Subestações: Tipos, Equipamentos e Proteção**. Disponível em:

<<http://www.uff.br/lev/downloads/apostilas/SE.pdf>>. Acesso em: 20 mar. 2017.

COMPUTER SHOPPER. **Ultimate Guide to Raspiberry PI**. Disponível em:

<<http://micklord.com/foru/Raspiberry%20Pi%20Pages%20from%20Computer%20Shopper%202015-02.pdf>>. Acesso em: 10 fev. 2017.

CREDER, H. **Instalações Elétricas**. 15. ed. Rio de Janeiro: LTC. 2013.

GRIOTTI, P. L. H. **Mostrador Digital**. 2008. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica), Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008.

GRUPO GENERGIA. Disponível em: <<http://www.genergia.com.br/monitoramento-de-energia-eletrica-e-utilidade-via-internet>>. Acesso em: 25 mar. 2017.

INMETRO. **Proposta de Regulamento Técnico Metrológico que estabelece os requisitos técnicos que devem ser atendidos pelos medidores eletrônicos de energia elétrica**.

Disponível em: <<http://www.inmetro.gov.br/legislacao/rtac/pdf/RTAC001738.pdf>>. Acesso em: 09 dez. 2016.

JÚNIOR, N. de N.; **Sistema de Monitoramento Online do Consumo de Energia Elétrica do Campus de Goiabeiras da Ufes**. 2014. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Colegiado do Curso de Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, 2014.

KRON MEDIDORES. Disponível em: <<http://www.kronweb.com.br/br/produto-6-mult-k-multimedidor-sem-memoria-de-massa>>. Acesso em: 05 dez. 2016.

MACKAY, S.; WRIGHT, E.; REYNDERS, D.; PARK, J. **Practical Industrial Data Networks: Design, Installation and Troubleshooting**. 1. ed. Oxford: Elsevier. 2004.

MÍNGUEZ, A.; **Medidores de Energia Ativa: Funcionamento, Práticas Usuais, Principais Ensaios e Análise das Fraudes Mais Comuns**. 2007. Trabalho de Conclusão de Curso

(Graduação em Engenharia Elétrica) - Escola Politécnica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007.

NANSEN Medidores. Disponível em: < <http://www.nansen.com.br/polifasicos,1,13>>. Acesso em: 26 fev. 2017.

RASPBERRY PI. Disponível em: < <https://www.raspberrypi.org/learning/hardware-guide/components/raspberry-pi/>>. Acesso em: 29 jan. 2017.

UTFPR; **Transformadores de medição e Proteção**. Disponível em: < <https://pt.slideshare.net/rodrigoprado54/transformadores-para-medio-e-proteo>>. Acesso em: 20 mar. 2017.

WM ENERGIA. Disponível em: < <http://www.wmenergia.com.br/wp-content/uploads/2016/06/E-book- Porque-devemos-monitorar-energia.pdf>>. Acesso em: 25 mar 2017.

APÊNDICE A – EQUAÇÕES UTILIZADAS INTERNAMENTE PELO MEDIDOR PARA CÁLCULO DAS GRANDEZAS

- Tensão RMS por fase:

$$V_{rms} = \sqrt{\sum_{1}^{n} (V_i)^2 / n} \quad (V) \quad (A.1)$$

Onde i é a ordem da amostra no cálculo do valor RMS; n , quantidade de amostras; V_{rms} a tensão em RMS.

- Corrente RMS:

$$I_{rms} = \sqrt{\sum_{1}^{n} (I_i)^2 / n} \quad (A) \quad (A.2)$$

Onde i é a ordem da amostra no cálculo do valor RMS; n , quantidade de amostras; I_{rms} a corrente em RMS.

- Potência ativa por fase:

$$P = \sum_{1}^{n} (V_i * I_i) / n \quad (W) \quad (A.3)$$

Onde i é a ordem da amostra no cálculo do valor RMS; n , quantidade de amostras; P a potência ativa por fase.

- Potência Aparente por fase:

$$S = V_{rms} * I_{rms} \quad (VA) \quad (A.4)$$

Onde V_{rms} é a tensão em RMS, I_{rms} é a corrente em RMS e S a potência aparente por fase.

- Potência Reativa por fase:

$$Q = \sqrt{S^2 - P^2} \text{ (Var)} \quad (\text{A.5})$$

Q é a potência reativa por fase, S a potência aparente por fase e P a potência ativa por fase.

- Fator de Potência por fase:

$$FP = P/S \quad (\text{A.6})$$

FP é o fator de potência por fase, S a potência aparente por fase e P a potência ativa por fase.

- Tensão Trifásica (instalação em configuração Delta):

$$V_{\emptyset} = (V_{12} + V_{23} + V_{31})/3 \text{ (V)} \quad (\text{A.7})$$

Onde V_{\emptyset} é a tensão trifásica; V_{12} é a tensão entre as fases 1 e 2; V_{23} é a tensão entre as fases 2 e 3; V_{31} é a tensão entre as fases 3 e 1.

- Tensão Trifásica (instalação em configuração Estrela):

$$V_{\emptyset} = \frac{V_{1N} + V_{2N} + V_{3N}}{3} * \sqrt{3} \text{ (V)} \quad (\text{A.8})$$

Onde V_{\emptyset} é a tensão trifásica; V_{1N} a tensão entre a fase 1 e o neutro; V_{2N} a tensão entre a fase 2 e o neutro; V_{3N} a tensão entre a fase 3 e o neutro.

- Potência Ativa Trifásica:

$$P_{\emptyset} = P_1 + P_2 + P_3 \text{ (W)} \quad (\text{A.9})$$

P_ϕ é potência ativa trifásica; P_1 é potência ativa na fase 1; P_2 é a potência ativa na fase 2; P_3 é a potência ativa na fase 3.

- Potência Reativa Trifásica:

$$Q_\phi = Q_1 + Q_2 + Q_3 \text{ (Var)} \quad (\text{A.10})$$

Q_ϕ é a potência reativa trifásica; Q_1 é a potência reativa na fase 1; Q_2 é a potência reativa na fase 2; Q_3 é a potência reativa na fase 3.

- Potência Aparente Trifásica:

$$S_\phi = \sqrt{P_\phi^2 + Q_\phi^2} \text{ (VA)} \quad (\text{A.11})$$

S_ϕ é a potência aparente trifásica; P_ϕ é potência ativa trifásica; Q_ϕ é a potência reativa trifásica.

- Corrente Trifásica:

$$I_\phi = \frac{S_\phi}{V_\phi * \sqrt{3}} \text{ (A)} \quad (\text{A.12})$$

I_ϕ é a corrente trifásica; S_ϕ é a potência aparente trifásica; V_ϕ é a tensão trifásica.

- Fator de Potência Trifásico:

$$FP_\phi = P_\phi / S_\phi \quad (\text{A.13})$$

FP_ϕ é o fator de potência trifásico; S_ϕ é a potência aparente trifásica; P_ϕ é potência ativa trifásica.

- Distorção harmônica total:

$$DHT = 100 * \left(\sqrt{\sum_{i=2}^{31} V_h^2} \right) / V_F (\%) \quad (\text{A.14})$$

Onde V_F é a Magnitude da Fundamental; V_h é Magnitude da harmônica de ordem i ; DHT é a distorção harmônica total.

APÊNDICE B – FUNÇÃO PRINCIPAL DA BIBLIOTECA PHPSERIAL

```
<?php
require_once 'PhpSerial.php';
class PhpSerialModbus
{
    // Enable for debugging
    public $debug = false;

    public function __construct()
    {
        $this->serial = new PhpSerial;
    }
    // Initialize serial port with specified parameters
    public function deviceInit($port='/dev/ttyUSB0', $baud=9600, $parity='none',
    $char=8, $sbits=1, $flow='none')
    {
        $this->serial->deviceSet($port);
        $this->serial->confBaudRate($baud);
        $this->serial->confParity($parity);
        $this->serial->confCharacterLength($char);
        $this->serial->confStopBits($sbits);
        $this->serial->confFlowControl($flow);
        exec('stty -F '.$port.' -brkint -icrnl -imaxbel -opost -isig -icanon -echo -echoe');
        return $this->serial->_dState;
    }
    // Open serial port
    public function deviceOpen()
    {
        $this->serial->deviceOpen();
        return $this->serial->_dState;
    }
}
```

```

// Close serial port
public function deviceClose()
{
    $this->serial->deviceClose();
    return $this->serial->_dState;
}
// Calculate CRC16 (ModBus)
public function crc16($data)
{
    $crc = 0xFFFF;
    for ($i = 0; $i < strlen($data); $i++)
    {
        $crc ^=ord($data[$i]);
        for ($j = 8; $j !=0; $j--)
        {
            if (($crc & 0x0001) !=0)
            {
                $crc >>= 1;
                $crc ^= 0xA001;
            }
            else
                $crc >>= 1;
        }
    }
    $highCrc=floor($crc/256);
    $lowCrc=($crc-$highCrc*256);
    return chr($lowCrc).chr($highCrc);
}

// Convert bin string to readable hex
private function bin2hexString ($binString)
{
    $hexString=bin2hex($binString);
    $hexString=chunk_split($hexString,2,"\\x");
}

```

```

    $hexString= "\\x" . substr($hexString,0,-2);
    return $hexString;
}

public function sendRawQuery ($string, $response = true) {
    $this->serial->sendMessage($string);
    if ($this->debug) print "DEBUG [query sent]: ".$this-
>bin2hexString($string)."\n";
    if ($response) return $this->getResponse(); else return 1;
}

// Send Modbus query to slave
public function sendQuery ($slaveId, $functionCode, $registerAddress,
$regCountOrData, $response = true)
{
    if ( ($functionCode > 6 ) || ($functionCode < 1) ) {
        if ($this->debug) print "DEBUG [invalid function code]\n";
        return 0;
    }

    if ($functionCode < 5) $regCountOrData = dechex($regCountOrData);

    $regHighByte=hexdec(substr($registerAddress,0,2));
    $regLowByte=hexdec(substr($registerAddress,2));

    $regCountOrData = str_pad($regCountOrData,4,"0",STR_PAD_LEFT);

    $rcdHighByte=hexdec(substr($regCountOrData,0,2));
    $rcdLowByte=hexdec(substr($regCountOrData,2));

    // Create query and convert to a binary string
    $query=array($slaveId, hexdec($functionCode), $regHighByte, $regLowByte,
$rcdHighByte, $rcdLowByte);

```

```

$queryString=implode(array_map("chr",$query));
// Calculate CRC
$queryString.=$this->crc16($queryString);

if ($this->debug) print "DEBUG [query sent]: ".$this-
>bin2hexString($queryString)."\n";

// Send over serial port
$this->serial->sendMessage($queryString);

if ($response) return $this->getResponse(); else return 1;
}
// Read response from slave
public function getResponse ($raw=false,$offsetl=0,$offsetr=0)
{
// Time started (for timing)
$startTime = microtime(true);

$responseString = "";

// Allow until one second for the response
while(((microtime(true)-$startTime)<1) && ($responseString=="") )
{
// Read serial port buffer (with three seconds timeout)
while( ($byte = $this->serial->ReadPort()) && ((microtime(true)-
$startTime)<3.0)) {
$responseString=$responseString.$byte;
usleep(50);
}
}
if ($this->debug) print "DEBUG [response received]: ".$this-
>bin2hexString($responseString)."\n";

```

```

if ($raw) return $responseString;

$stringLength=strlen($responseString);

// If we have at least 5 bytes...
if ($stringLength>=5)
{
    // ...but no more than 5
    if ($stringLength==5) {
        if ($this->debug) print "DEBUG [no valid data]\n";
        return 0;
    }
    // CRC Check
    $checkArray=(str_split($responseString,strlen($responseString)-2));

    $crc=$this->crc16($checkArray[0]);
    if ($crc!=$checkArray[1]) {
        if ($this->debug) print "DEBUG [crc check failed]: (expected
".$this->bin2hexString($crc)." received ".$this->bin2hexString($checkArray[1]).")\n";
        return 0;
    }
    // Convert string in array of bytes
    $responseArray = str_split($responseString);
    // This is the byte containing the number of data bytes
    $bytesNum = hexdec(bin2hex($responseArray[2]));

    // Create a new array with data without headers and CRC
    $responseData = array_slice($responseArray,3+$offsetl,$bytesNum-
$offsetr);

    if ($this->debug) print "DEBUG [data]: ".$this-
>bin2hexString(implode($responseData))."\n";

    // Return an array with hex data bytes
    return array_map("bin2hex",$responseData);

```

```
    } else
    {
        if ($this->debug) print "DEBUG [no response]\n";
        return 0;
    }
    return 0;
}
}
```

?>

APÊNDICE C – QUADRO DE REGISTRADORES DO MEDIDOR MULT-K

| REGISTRADOR (HEXADECIMAL) | DESCRIÇÃO | TIPO |
|--------------------------------------|----------------------------------|---------------------------------|
| 0x00 | Número de Série | Unsigned int 32-bit (MSB, LSB) |
| 0x02 | Tensão Trifásica (V) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x04 | Corrente Trifásica (A) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x06 | Fator de Potência Trifásico | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x08 | Potência Aparente Trifásica (VA) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x0A | Potência Reativa Trifásica (Var) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x0C | Potência Ativa Trifásica (W) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x0E | Frequência (Hz) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x10 | Tensão de Fase 1 (V) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x12 | Tensão de Fase 2 (V) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x14 | Tensão de Fase 3 (V) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x16 | Corrente Linha 1 (A) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x18 | Corrente Linha 2 (A) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x1A | Corrente Linha 3 (A) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x1C | Potência Ativa Linha 1 (W) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x1E | Potência Ativa Linha 2 (W) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x20 | Potência Ativa Linha 3 (W) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x22 | Potência Reativa Linha 1 (Var) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x24 | Potência Reativa Linha 2 (Var) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x26 | Potência Reativa Linha 3 (Var) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x28 | Potência Aparente Linha 1 (VA) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x2A | Potência Aparente Linha 2 (VA) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x2C | Potência Aparente Linha 3 (VA) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x2E | Fator de Potência Linha 1 | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x30 | Fator de Potência Linha 2 | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x32 | Fator de Potência Linha 3 | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x34 | Energia Ativa Positiva (kWh) | IEEE 32-bit fp(F2, F1, F0, EXP) |

| | | |
|------|----------------------------------|---------------------------------|
| 0x36 | Energia Reativa Positiva (kVarh) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x38 | Energia Ativa Negativa (kWh) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x3A | Energia Reativa Negativa (kVarh) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x3C | Máxima Demanda Ativa (W) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x3E | Demanda Ativa (W) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x40 | Máxima Demanda Aparente (VA) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x42 | Demanda Aparente (VA) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x54 | Tensão de Linha 12 (V) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x56 | Tensão de Linha 23 (V) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x58 | Tensão de Linha 31 (V) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x5A | Máxima Tensão Trifásica (V) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0x5C | Máxima Corrente Trifásica (A) | IEEE 32-bit fp(F2, F1, F0, EXP) |
| 0xC8 | DHT – Tensão Linha 1 | Inteiro 16-bit (MSB, LSB) |
| 0xC9 | DHT – Tensão Linha 2 | Inteiro 16-bit (MSB, LSB) |
| 0XCA | DHT – Tensão Linha 3 | Inteiro 16-bit (MSB, LSB) |
| 0XCB | DHT – Corrente Linha 1 | Inteiro 16-bit (MSB, LSB) |
| 0XCC | DHT – Corrente Linha 2 | Inteiro 16-bit (MSB, LSB) |
| 0XCD | DHT – Corrente Linha 3 | Inteiro 16-bit (MSB, LSB) |
| 0xDD | Código de Erro | Inteiro 8-bit (MSB=0, LSB) |

Fonte: KRON MEDIDORES (2016) Adaptado.

Quadro 4 – Características dos bits do padrão IEEE-754

| EXP | F0 | F1 | F2 |
|-----------------|-----------------|-----------------|-----------------|
| 0 1 1 1 0 0 1 0 | 0 1 1 1 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 |

Fonte: KRON MEDIDORES (2016) Adaptado.

APÊNDICE D – CÓDIGO PHP DA CONVERSÃO IEEE-754 HEXADECIMAL PARA DECIMAL

```
function hexTo32Float($strHex) {
    $bin = str_pad(base_convert($strHex, 16, 2), 32, "0", STR_PAD_LEFT);
    $sign = $bin[0];
    $exp = bindec(substr($bin, 1, 8)) - 127;
    $man = (2 << 22) + bindec(substr($bin, 9, 23));
    $dec = $man * pow(2, $exp - 23) * ($sign ? -1 : 1);
    return $dec;
}
```

O código inicia transformando o valor recebido (*\$strHex*) em hexadecimal para binário. Adiciona zeros à esquerda desse novo binário até completar a quantidade de 32 bits, caso a transformação retorne uma quantidade menor que 32 bits. Esses passos são salvos em *\$bin*. Em seguida, o valor da posição zero do binário é salvo em *\$sign*. Em *\$exp*, é salva a subtração entre, o valor decimal dos bits de 1 a 8 e, 127. Em *\$man*, é salvo um deslocamento de 22 à esquerda (multiplica por 44), somado com o valor decimal dos bits de 9 a 23. Por fim, para encontrar o número decimal é resolvida a equação D.1.

$$decimal = \$man * 2^{exp-23} * sinal \quad (D.1)$$

Onde sinal será igual a 1, caso o valor salvo em *\$sign* seja 0 ou, igual a -1 caso *\$sign* possua valor 1.

APÊNDICE E – CÓDIGO DA PÁGINA INICIAL

```
<?php $nome_pagina = "Home"; include 'cabecalho.php'; ?>
```

```
<div class="jumbotron">
```

```
<center></center>
```

```
<p>&nbsp;</p>
```

```
<p align="center">
```

```
<strong>Este site foi desenvolvido para monitoramento de grandezas
el&eacute;tricas no pr&eacute;dio CT XI, Campus Goiabeiras, </strong><strong>da
Universidade Federal do Esp&eacute;rito Santo</strong></p>
```

```
<p align="center">
```

```
<p>
```

```
Para a medi&ccedil;&atilde;o das grandezas el&eacute;tricas, encontra-
se instalado no quadro de distribui&ccedil;&atilde;o geral de energia el&eacute;trica do
pr&eacute;dio, um medidor eletr&ocirc;nico, modelo Mult-k, da empresa Kron
medidores.</p>
```

```
<p>
```

```
&nbsp;</p>
```

```
<center></center>
```

```
<p>&nbsp;</p>
```

```
<p>
```

Os dados das leituras desse medidor são enviados para um microcomputador, o Raspberry Pi, através da interface serial. Nesse, os dados são tratados e exibidos neste site para o usuário.

```
<center></center>
```

```
<p>&nbsp;</p>
```

```
</div>
```

```
<?php
```

```
include 'rodape.php';
```

```
?>
```

APÊNDICE F – CÓDIGO DA OPÇÃO LEITURAS

```
<?php
$nome_pagina = "Leituras";
include 'cabecalho.php';
$InicioCarga=date("H-i-s");
$critico = 0;

// Redirecionamento Automático, atualização de dados
$Stempo = explode('-', $InicioCarga);

if($Stempo[1] == 00 || $Stempo[1] == 15 || $Stempo[1] == 30 || $Stempo[1] == 45) {
    if($Stempo[2] < 15) {
        $critico = 1;
        header("Location: http://172.20.25.162/atualizando.php");
    }
} else if($Stempo[1] == 59 || $Stempo[1] == 14 || $Stempo[1] == 29 || $Stempo[1] == 44) {
    if($Stempo[2] > 56) {
        $critico = 1;
        header("Location: http://172.20.25.162/atualizando.php");
    }
}
?>

<div class="jumbotron">
  <h2>Leituras</h2>
  <table class="table">
    <thead>
      <tr>
        <th>Grandeza Elétrica</th>
        <th>Valor</th>
      </tr>
    </thead>
    <tbody>
```

```

<?php
//Pegando URL e Atualizando a pagina a cada 30 segundos
$url=$_SERVER['REQUEST_URI'];
header("Refresh: 30; URL=$url");

if($critico == 0) {
require 'mod_bus/PhpSerialModbus.php';

function hexTo32Float($strHex) {
    $bin = str_pad(base_convert($strHex, 16, 2), 32, "0", STR_PAD_LEFT);
    $sign = $bin[0];
    $exp = bindec(substr($bin, 1, 8)) - 127;
    $man = (2 << 22) + bindec(substr($bin, 9, 23));
    $dec = $man * pow(2, $exp - 23) * ($sign ? -1 : 1);
    return $dec;
}
$modbus = new PhpSerialModbus;
// Inicializa porta
$modbus->deviceInit('/dev/ttyUSB0',9600,'none',8,1,'none');
$modbus->deviceOpen();
}
// Frequencia
$result=$modbus->sendQuery(1,4,"000e",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Frequência</td>
    <td>{$final} Hz</td>
</tr>";
// Tensao de fase 1
$result=$modbus->sendQuery(1,4,"0010",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Tensão de Fase 1</td>

```

```

        <td>{$final} V</td>
    </tr>";
// Tensao de fase 2
$result=$modbus->sendQuery(1,4,"0012",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Tensão de Fase 2</td>
    <td>{$final} V</td>
</tr>";
// Tensao de fase 3
$result=$modbus->sendQuery(1,4,"0014",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Tensão de Fase 3</td>
    <td>{$final} V</td>
</tr>";
// Tensao entre linhas 12
$result=$modbus->sendQuery(1,4,"0054",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Tensão entre linhas 12</td>
    <td>{$final} V</td>
</tr>";
// Tensao entre linhas 23
$result=$modbus->sendQuery(1,4,"0056",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Tensão entre linhas 23</td>
    <td>{$final} V</td>
</tr>";
// Tensao entre linhas 31
$result=$modbus->sendQuery(1,4,"0058",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);

```

```

echo "<tr class='success'>
    <td>Tensão entre linhas 31</td>
    <td>{$final} V</td>
</tr>";

// Corrente linha 1
$result=$modbus->sendQuery(1,4,"0016",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Corrente Linha 1</td>
    <td>{$final} A</td>
</tr>";

// Corrente linha 2
$result=$modbus->sendQuery(1,4,"0018",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Corrente Linha 2</td>
    <td>{$final} A</td>
</tr>";

// Corrente linha 3
$result=$modbus->sendQuery(1,4,"001a",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Corrente Linha 3</td>
    <td>{$final} A</td>
</tr>";

// Potencia ativa linha 1
$result=$modbus->sendQuery(1,4,"001c",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Potência Ativa Linha 1</td>
    <td>{$final} W</td>
</tr>";

// Potencia ativa linha 2

```



```

$result=$modbus->sendQuery(1,4,"001e",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Potência Ativa Linha 2</td>
    <td>{$final} W</td>
</tr>";

// Potencia ativa linha 3
$result=$modbus->sendQuery(1,4,"0020",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Potência Ativa Linha 3</td>
    <td>{$final} W</td>
</tr>";

// Potencia reativa linha 1
$result=$modbus->sendQuery(1,4,"0022",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Potência Reativa Linha 1</td>
    <td>{$final} Var</td>
</tr>";

// Potencia reativa linha 2
$result=$modbus->sendQuery(1,4,"0024",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Potência Reativa Linha 2</td>
    <td>{$final} Var</td>
</tr>";

// Potencia reativa linha 3
$result=$modbus->sendQuery(1,4,"0026",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Potência Reativa Linha 3</td>
    <td>{$final} Var</td>

```

```

    </tr>";
// Potencia aparente linha 1
$result=$modbus->sendQuery(1,4,"0028",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Potência Aparente Linha 1</td>
    <td>{$final} VA</td>
</tr>";
// Potencia aparente linha 2
$result=$modbus->sendQuery(1,4,"002a",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Potência Aparente Linha 2</td>
    <td>{$final} VA</td>
</tr>";
// Potencia aparente linha 3
$result=$modbus->sendQuery(1,4,"002c",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Potência Aparente Linha 3</td>
    <td>{$final} VA</td>
</tr>";
// Fator de potencia linha 1
$result=$modbus->sendQuery(1,4,"002e",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Fator de Potência Linha 1</td>
    <td>{$final} </td>
</tr>";
// Fator de potencia linha 2
$result=$modbus->sendQuery(1,4,"0030",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>

```

```

        <td>Fator de Potência Linha 2</td>
        <td>{$final} </td>
    </tr>";

// Fator de potencia linha 3
$result=$modbus->sendQuery(1,4,"0032",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Fator de Potência Linha 3</td>
    <td>{$final} </td>
</tr>";

// Tensao Trifasica
$result=$modbus->sendQuery(1,4,"0002",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Tensão Trifásica</td>
    <td>{$final} V</td>
</tr>";

// Maxima Tensao Trifasica
$result=$modbus->sendQuery(1,4,"005a",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Máxima Tensão Trifásica</td>
    <td>{$final} V</td>
</tr>";

// Corrente Trifasica
$result=$modbus->sendQuery(1,4,"0004",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Corrente Trifásica</td>
    <td>{$final} A</td>
</tr>";

// Maxima Corrente Trifasica
$result=$modbus->sendQuery(1,4,"005c",2);

```

```

$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Máxima Corrente Trifásica</td>
    <td>{$final} A</td>
</tr>";

// fator de potencia trifasico
$result=$modbus->sendQuery(1,4,"0006",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Fator de Potência Trifásico</td>
    <td>{$final} </td>
</tr>";

// Potencia aparente trifasica
$result=$modbus->sendQuery(1,4,"0008",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Potência Aparente Trifásica</td>
    <td>{$final} VA</td>
</tr>";

// Potencia reativa trifasica
$result=$modbus->sendQuery(1,4,"000a",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Potência Reativa Trifásica</td>
    <td>{$final} Var</td>
</tr>";

// Potencia ativa trifasica
$result=$modbus->sendQuery(1,4,"000c",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Potência Ativa Trifásica</td>
    <td>{$final} W</td>

```

```

    </tr>";
// Energia ativa positiva
$result=$modbus->sendQuery(1,4,"0034",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Energia Ativa Positiva</td>
    <td>{$final} kWh</td>
</tr>";
// Energia ativa negativa
$result=$modbus->sendQuery(1,4,"0038",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Energia Ativa Negativa</td>
    <td>{$final} kWh</td>
</tr>";
// Energia reativa positiva
$result=$modbus->sendQuery(1,4,"0036",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Energia Reativa Positiva</td>
    <td>{$final} kVarh</td>
</tr>";
// Energia reativa negativa
$result=$modbus->sendQuery(1,4,"003a",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Energia Reativa Negativa</td>
    <td>{$final} kVarh</td>
</tr>";
// Demanda ativa
$result=$modbus->sendQuery(1,4,"003e",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),4);
echo "<tr class='success'>

```

```

        <td>Demanda Ativa </td>
        <td>{$final} W</td>
    </tr>";

// Maxima Demanda ativa
$result=$modbus->sendQuery(1,4,"003c",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),4);
echo "<tr class='success'>
    <td>Máxima Demanda Ativa </td>
    <td>{$final} W</td>
</tr>";

// Demanda aparente
$result=$modbus->sendQuery(1,4,"0042",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Demanda Aparente </td>
    <td>{$final} VA</td>
</tr>";

// Maxima Demanda aparente
$result=$modbus->sendQuery(1,4,"004",2);
$final = round(hexTo32Float($result[3].$result[2].$result[1].$result[0]),2);
echo "<tr class='success'>
    <td>Máxima Demanda Aparente </td>
    <td>{$final} VA</td>
</tr>";

// THD – Tensão Linha 1
$result=$modbus->sendQuery(1,4,"00c8",1);
$valor = $result[0].$result[1];
$final = (hexdec($valor[0].$valor[3])*0.1);
echo "<tr class='success'>
    <td>THD – Tensão Linha 1</td>
    <td>{$final}</td>
</tr>";

// THD – Tensão Linha 2

```

```
$result=$modbus->sendQuery(1,4,"00c9",1);
$valor = $result[0].$result[1];
$final = (hexdec($valor[0].$valor[3])*0.1);
echo "<tr class='success'>
    <td>THD – Tensão Linha 2</td>
    <td>{$final}</td>
</tr>";
// THD – Tensão Linha 3
$result=$modbus->sendQuery(1,4,"00ca",1);
$valor = $result[0].$result[1];
$final = (hexdec($valor[0].$valor[3])*0.1);
echo "<tr class='success'>
    <td>THD – Tensão Linha 3</td>
    <td>{$final}</td>
</tr>";
// THD – Corrente Linha 1
$result=$modbus->sendQuery(1,4,"00cb",1);
$valor = $result[0].$result[1];
$final = (hexdec($valor[0].$valor[3])*0.1);
echo "<tr class='success'>
    <td>THD – Corrente Linha 1</td>
    <td>{$final}</td>
</tr>";
// THD – Corrente Linha 2
$result=$modbus->sendQuery(1,4,"00cc",1);
$valor = $result[0].$result[1];
$final = (hexdec($valor[0].$valor[3])*0.1);
echo "<tr class='success'>
    <td>THD – Corrente Linha 2</td>
    <td>{$final}</td>
</tr>";
// THD – Corrente Linha 3
$result=$modbus->sendQuery(1,4,"00cd",1);
```

```
$valor = $result[0].$result[1];  
$final = (hexdec($valor[0].$valor[3])*0.1);  
echo "<tr class='success'>  
    <td>THD – Corrente Linha 3</td>  
    <td>{$final}</td>  
</tr>";  
// fecha porta  
$modbus->deviceClose();  
>  
    </tbody>  
</table>  
</div>  
</div>  
<?php  
include 'rodape.php';  
>
```


APÊNDICE G – CÓDIGO DA OPÇÃO TABELA

```
<?php
$nome_pagina = "Tabela";
include 'cabecalho.php';
if (empty($_GET["data1"])) {
    $data1 = date('Y-m-d');
} else {
    $data1 = implode("-",array_reverse(explode("-",$_GET['data1'])));
}

if (empty($_GET["data2"])) {
    $data2 = date('Y-m-d');
} else {
    $data2 = implode("-",array_reverse(explode("-",$_GET['data2'])));
}

if (empty($_GET["hora1"])) {
    $hora1 = '00:00:00';
} else {
    $hora1 = $_GET["hora1"];
}

if (empty($_GET["hora2"])) {
    $hora2 = '23:59:59';
} else {
    $hora2 = $_GET["hora2"];
}

if (empty($_GET["tipo"])) {
    $tipo = "";
} else {
    $tipo = $_GET["tipo"];
}
```

```
}  
switch($tipo) {  
  
    case "0002":  
        $nome = "Tensão Trifásica (V)";  
        break;  
    case "0004":  
        $nome = "Corrente Trifásica (A)";  
        break;  
    case "0006":  
        $nome = "Fator de Potência Trifásico";  
        break;  
    case "0008":  
        $nome = "Potência Aparente Trifásica (VA)";  
        break;  
    case "000a":  
        $nome = "Potência Reativa Trifásica (Var)";  
        break;  
    case "000c":  
        $nome = "Potência Ativa Trifásica (W)";  
        break;  
    case "000e":  
        $nome = "Frequência (Hz)";  
        break;  
    case "0016":  
        $nome = "Corrente Linha 1 (A)";  
        break;  
    case "0018":  
        $nome = "Corrente Linha 2 (A)";  
        break;  
    case "001a":  
        $nome = "Corrente Linha 3 (A)";  
        break;
```

```
case "001c":
    $nome = "Potência Ativa Linha 1 (W)";
    break;
case "001e":
    $nome = "Potência Ativa Linha 2 (W)";
    break;
case "0020":
    $nome = "Potência Ativa Linha 3 (W)";
    break;
case "0022":
    $nome = "Potência Reativa Linha 1 (Var)";
    break;
case "0024":
    $nome = "Potência Reativa Linha 2 (Var)";
    break;
case "0026":
    $nome = "Potência Reativa Linha 3 (Var)";
    break;
case "0028":
    $nome = "Potência Aparente Linha 1 (VA)";
    break;
case "002a":
    $nome = "Potência Aparente Linha 2 (VA)";
    break;
case "002c":
    $nome = "Potência Aparente Linha 3 (VA)";
    break;
case "002e":
    $nome = "Fator de Potência Linha 1";
    break;
case "0030":
    $nome = "Fator de Potência Linha 2";
    break;
```

```
case "0032":
    $nome = "Fator de Potência Linha 3";
    break;
case "0034":
    $nome = "Energia Ativa Positiva (kWh)";
    break;
case "0036":
    $nome = "Energia Reativa Positiva (kVarh)";
    break;
case "0038":
    $nome = "Energia Ativa Negativa (kWh)";
    break;
case "003a":
    $nome = "Energia Reativa Negativa (kVarh)";
    break;
case "003c":
    $nome = "Máxima Demanda Ativa (W)";
    break;
case "003e":
    $nome = "Demanda Ativa (W)";
    break;
case "0040":
    $nome = "Máxima Demanda Aparente (VA)";
    break;
case "0042":
    $nome = "Demanda Aparente (VA)";
    break;
case "0054":
    $nome = "Tensão entre Linhas 12 (V)";
    break;
case "0056":
    $nome = "Tensão entre Linhas 23 (V)";
    break;
```

```
case "0058":
    $nome = "Tensão entre Linhas 31 (V)";
    break;
case "005a":
    $nome = "Máxima Tensão Trifásica (V)";
    break;
case "005c":
    $nome = "Máxima Corrente Trifásica (A)";
    break;
case "00c8":
    $nome = "THD - Tensão Linha 1";
    break;
case "00c9":
    $nome = "THD - Tensão Linha 2";
    break;
case "00ca":
    $nome = "THD - Tensão Linha 3";
    break;
case "00cb":
    $nome = "THD - Corrente Linha 1";
    break;
case "00cc":
    $nome = "THD - Corrente Linha 2";
    break;
case "00cd":
    $nome = "THD - Corrente Linha 3";
    break;
default:
    $nome = "Nenhum Tipo";
}
if (!empty($_GET["tipo"])) {
    $servername = "localhost";
    $username = "root";
```

```

$password = "123456";
$dbname = "pg";
// cria conexao
$conn = new mysqli($servername, $username, $password, $dbname);
// checa conexao
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT * FROM valores WHERE reg_id='$tipo' AND timestamp
BETWEEN '$data1 ' . "$hora1' AND '$data2 ' . "$hora2'";
$result = $conn->query($sql);
}
?>
<div class="jumbotron">
<h2>Tabela</h2>
<p></p>
<form action="" method="get">

<table style="width:100%">
  <tr>
  <td>
      <label for="sel1">Tipo:</label>
      <select name="tipo" class="form-control" id="sel1" style="width: 350px;" >
          <option <?php if ($tipo==" " or $tipo=="0016") echo
'selected="selected"' ?> value="0016">Corrente Linha 1 (A)</option>
          <option <?php if ($tipo=="0018") echo 'selected="selected"' ?>
value="0018">Corrente Linha 2 (A)</option>
          <option <?php if ($tipo=="001a") echo 'selected="selected"' ?>
value="001a">Corrente Linha 3 (A)</option>
          <option <?php if ($tipo=="0004") echo 'selected="selected"' ?>
value="0004">Corrente Trifásica (A)</option>
          <option <?php if ($tipo=="0042") echo 'selected="selected"' ?>
value="0042">Demanda Aparente (VA)</option>

```

```

        <option <?php if ($tipo=="003e") echo 'selected="selected"'; ?>
value="003e">Demanda Ativa (W)</option>
        <option <?php if ($tipo=="0034") echo 'selected="selected"'; ?>
value="0034">Energia Ativa Positiva (kWh)</option>
        <option <?php if ($tipo=="0038") echo 'selected="selected"'; ?>
value="0038">Energia Ativa Negativa (kWh)</option>
        <option <?php if ($tipo=="0036") echo 'selected="selected"'; ?>
value="0036">Energia Reativa Positiva (kVarh)</option>
        <option <?php if ($tipo=="003a") echo 'selected="selected"'; ?>
value="003a">Energia Reativa Negativa (kVarh)</option>
        <option <?php if ($tipo=="002e") echo 'selected="selected"'; ?>
value="002e">Fator de Potência Linha 1</option>
        <option <?php if ($tipo=="0030") echo 'selected="selected"'; ?>
value="0030">Fator de Potência Linha 2</option>
        <option <?php if ($tipo=="0032") echo 'selected="selected"'; ?>
value="0032">Fator de Potência Linha 3</option>
        <option <?php if ($tipo=="0006") echo 'selected="selected"'; ?>
value="0006">Fator de Potência Trifásico</option>
        <option <?php if ($tipo=="000e") echo 'selected="selected"'; ?>
value="000e">Frequência (Hz)</option>
        <option <?php if ($tipo=="005c") echo 'selected="selected"'; ?>
value="005c">Máxima Corrente Trifásica (A)</option>
        <option <?php if ($tipo=="0040") echo 'selected="selected"'; ?>
value="0040">Máxima Demanda Aparente (VA)</option>
        <option <?php if ($tipo=="003c") echo 'selected="selected"'; ?>
value="003c">Máxima Demanda Ativa (W)</option>
        <option <?php if ($tipo=="005a") echo 'selected="selected"'; ?>
value="005a">Máxima Tensão Trifásica (V)</option>
        <option <?php if ($tipo=="0028") echo 'selected="selected"'; ?>
value="0028">Potência Aparente Linha 1 (VA)</option>
        <option <?php if ($tipo=="002a") echo 'selected="selected"'; ?>
value="002a">Potência Aparente Linha 2 (VA)</option>

```

```

        <option <?php if ($tipo=="002c") echo 'selected="selected"'; ?>
value="002c">Potência Aparente Linha 3 (VA)</option>
        <option <?php if ($tipo=="0008") echo 'selected="selected"'; ?>
value="0008">Potência Aparente Trifásica (VA)</option>
        <option <?php if ($tipo=="001c") echo 'selected="selected"'; ?>
value="001c">Potência Ativa Linha 1 (W)</option>
        <option <?php if ($tipo=="001e") echo 'selected="selected"'; ?>
value="001e">Potência Ativa Linha 2 (W)</option>
        <option <?php if ($tipo=="0020") echo 'selected="selected"'; ?>
value="0020">Potência Ativa Linha 3 (W)</option>
        <option <?php if ($tipo=="000c") echo 'selected="selected"'; ?>
value="000c">Potência Ativa Trifásica (W)</option>
        <option <?php if ($tipo=="0022") echo 'selected="selected"'; ?>
value="0022">Potência Reativa Linha 1 (Var)</option>
        <option <?php if ($tipo=="0024") echo 'selected="selected"'; ?>
value="0024">Potência Reativa Linha 2 (Var)</option>
        <option <?php if ($tipo=="0026") echo 'selected="selected"'; ?>
value="0026">Potência Reativa Linha 3 (Var)</option>
        <option <?php if ($tipo=="000a") echo 'selected="selected"'; ?>
value="000a">Potência Reativa Trifásica (Var)</option>
        <option <?php if ($tipo=="0054") echo 'selected="selected"'; ?>
value="0054">Tensão entre Linhas 12 (V)</option>
        <option <?php if ($tipo=="0056") echo 'selected="selected"'; ?>
value="0056">Tensão entre Linhas 23 (V)</option>
        <option <?php if ($tipo=="0058") echo 'selected="selected"'; ?>
value="0058">Tensão entre Linhas 31 (V)</option>
        <option <?php if ($tipo=="0002") echo 'selected="selected"'; ?>
value="0002">Tensão Trifásica (V)</option>
        <option <?php if ($tipo=="00cb") echo 'selected="selected"'; ?>
value="00cb">THD - Corrente Linha 1</option>
        <option <?php if ($tipo=="00cc") echo 'selected="selected"'; ?>
value="00cc">THD - Corrente Linha 2</option>

```



```

        <option <?php if ($tipo=="00cd") echo 'selected="selected"'; ?>
value="00cd">THD - Corrente Linha 3</option>
        <option <?php if ($tipo=="00c8") echo 'selected="selected"'; ?>
value="00c8">THD - Tensão Linha 1</option>
        <option <?php if ($tipo=="00c9") echo 'selected="selected"'; ?>
value="00c9">THD - Tensão Linha 2</option>
        <option <?php if ($tipo=="00ca") echo 'selected="selected"'; ?>
value="00ca">THD - Tensão Linha 3</option>

```

```

    </select>

```

```

    </td>

```

```

    <td>

```

```

        <label for="usr">Data Inicial:</label>

```

```

        <input type="text" class="form-control" name ="data1" id="calendario1"
style="width: 150px;" placeholder="Dia-Mes-Ano" value="<?php echo implode("-"
,array_reverse(explode("-", $data1)));?>">

```

```

    </td>

```

```

    <td>

```

```

        <label for="usr">Data Final:</label>

```

```

        <input type="text" class="form-control" name ="data2" id="calendario2"
style="width: 150px;" placeholder="Dia-Mes-Ano" value="<?php echo implode("-"
,array_reverse(explode("-", $data2)));?>">

```

```

    </td>

```

```

    <td>

```

```

        <label for="sel1">Hora Inicial:</label>

```

```

        <select name="hora1" class="form-control" id="sel1" style="width: 150px;"

```

```

>

```

```

        <option selected="selected"><?php echo $hora1; ?></option>

```

```

        <option>00:00:00</option>

```

```

        <option>00:15:00</option>

```

```

        <option>00:30:00</option>

```

```

        <option>00:45:00</option>

```

```

        <option>01:00:00</option>

```

<option>01:15:00</option>
<option>01:30:00</option>
<option>01:45:00</option>
<option>02:00:00</option>
<option>02:15:00</option>
<option>02:30:00</option>
<option>02:45:00</option>
<option>03:00:00</option>
<option>04:15:00</option>
<option>04:30:00</option>
<option>04:45:00</option>
<option>05:00:00</option>
<option>05:15:00</option>
<option>05:30:00</option>
<option>05:45:00</option>
<option>06:00:00</option>
<option>06:15:00</option>
<option>06:30:00</option>
<option>06:45:00</option>
<option>07:00:00</option>
<option>07:15:00</option>
<option>07:30:00</option>
<option>07:45:00</option>
<option>08:00:00</option>
<option>08:15:00</option>
<option>08:30:00</option>
<option>08:45:00</option>
<option>09:00:00</option>
<option>09:15:00</option>
<option>09:30:00</option>
<option>09:45:00</option>
<option>10:00:00</option>
<option>10:15:00</option>

<option>10:30:00</option>
<option>10:45:00</option>
<option>11:00:00</option>
<option>11:15:00</option>
<option>11:30:00</option>
<option>11:45:00</option>
<option>12:00:00</option>
<option>12:15:00</option>
<option>12:30:00</option>
<option>12:45:00</option>
<option>13:00:00</option>
<option>13:15:00</option>
<option>13:30:00</option>
<option>13:45:00</option>
<option>14:00:00</option>
<option>14:15:00</option>
<option>14:30:00</option>
<option>14:45:00</option>
<option>15:00:00</option>
<option>15:15:00</option>
<option>15:30:00</option>
<option>15:45:00</option>
<option>16:00:00</option>
<option>16:15:00</option>
<option>16:30:00</option>
<option>16:45:00</option>
<option>17:00:00</option>
<option>17:15:00</option>
<option>17:30:00</option>
<option>17:45:00</option>
<option>18:00:00</option>
<option>18:15:00</option>
<option>18:30:00</option>

```
<option>18:45:00</option>
<option>19:00:00</option>
<option>19:15:00</option>
<option>19:30:00</option>
<option>19:45:00</option>
<option>20:00:00</option>
<option>20:15:00</option>
<option>20:30:00</option>
<option>20:45:00</option>
<option>21:00:00</option>
<option>21:15:00</option>
<option>21:30:00</option>
<option>21:45:00</option>
<option>22:00:00</option>
<option>22:15:00</option>
<option>22:30:00</option>
<option>22:45:00</option>
<option>23:00:00</option>
<option>23:15:00</option>
<option>23:30:00</option>
<option>23:45:00</option>
<option>23:59:59</option>
</select>
</td>
<td>
<label for="sel1">Hora Final:</label>
<select name="hora2" class="form-control" id="sel1" style="width: 150px;"
>
<option>00:15:00</option>
<option>00:30:00</option>
<option>00:45:00</option>
<option>01:00:00</option>
<option>01:15:00</option>
```

<option>01:30:00</option>
<option>01:45:00</option>
<option>02:00:00</option>
<option>02:15:00</option>
<option>02:30:00</option>
<option>02:45:00</option>
<option>03:00:00</option>
<option>04:15:00</option>
<option>04:30:00</option>
<option>04:45:00</option>
<option>05:00:00</option>
<option>05:15:00</option>
<option>05:30:00</option>
<option>05:45:00</option>
<option>06:00:00</option>
<option>06:15:00</option>
<option>06:30:00</option>
<option>06:45:00</option>
<option>07:00:00</option>
<option>07:15:00</option>
<option>07:30:00</option>
<option>07:45:00</option>
<option>08:00:00</option>
<option>08:15:00</option>
<option>08:30:00</option>
<option>08:45:00</option>
<option>09:00:00</option>
<option>09:15:00</option>
<option>09:30:00</option>
<option>09:45:00</option>
<option>10:00:00</option>
<option>10:15:00</option>
<option>10:30:00</option>

<option>10:45:00</option>
<option>11:00:00</option>
<option>11:15:00</option>
<option>11:30:00</option>
<option>11:45:00</option>
<option>12:00:00</option>
<option>12:15:00</option>
<option>12:30:00</option>
<option>12:45:00</option>
<option>13:00:00</option>
<option>13:15:00</option>
<option>13:30:00</option>
<option>13:45:00</option>
<option>14:00:00</option>
<option>14:15:00</option>
<option>14:30:00</option>
<option>14:45:00</option>
<option>15:00:00</option>
<option>15:15:00</option>
<option>15:30:00</option>
<option>15:45:00</option>
<option>16:00:00</option>
<option>16:15:00</option>
<option>16:30:00</option>
<option>16:45:00</option>
<option>17:00:00</option>
<option>17:15:00</option>
<option>17:30:00</option>
<option>17:45:00</option>
<option>18:00:00</option>
<option>18:15:00</option>
<option>18:30:00</option>
<option>18:45:00</option>

```

        <option>19:00:00</option>
        <option>19:15:00</option>
        <option>19:30:00</option>
        <option>19:45:00</option>
        <option>20:00:00</option>
        <option>20:15:00</option>
        <option>20:30:00</option>
        <option>20:45:00</option>
        <option>21:00:00</option>
        <option>21:15:00</option>
        <option>21:30:00</option>
        <option>21:45:00</option>
        <option>22:00:00</option>
        <option>22:15:00</option>
        <option>22:30:00</option>
        <option>22:45:00</option>
        <option>23:00:00</option>
        <option>23:15:00</option>
        <option>23:30:00</option>
        <option>23:45:00</option>
        <option>23:59:59</option>
        <option selected="selected"><?php echo $hora2; ?></option>
    </select>
</td>
</tr>
</table>
<p><div align="center"><button type="submit" class="btn btn-primary
active">Seleccionar</button></div></p>
</form>

<hr>
<table class="table">
    <thead>

```

```

<tr>
  <th>Data</th>
  <th>Valor</th>
</tr>
</thead>
  <tbody>
<?php
if (!empty($_GET["tipo"])) {
    $data1_br = implode("-",array_reverse(explode("-", $data1)));
    $data2_br = implode("-",array_reverse(explode("-", $data2)));

    echo "<b><font size='4'>Listando: {$nome} de {$data1_br} as {$hora1} até
{$data2_br} as {$hora2}</font></b>";
    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            $data = date('d-m-Y H:i', strtotime($row['timestamp']));
            echo "<tr class='success'>
                <td>{$data}</td>
                <td>{$row['val']}</td>
            </tr>";
        }
    } else {
        echo "<tr class='success'>
            <td>Sem Resultados.</td>
            <td></td>
        </tr>";
    }
    $conn->close();
}
?>

</tbody>

```



```
</table>
</div>
<script>
$(function() {
    $("#calendario1").datepicker({dateFormat: 'dd-mm-yy'});
    $("#calendario2").datepicker({dateFormat: 'dd-mm-yy'});
});
</script>
<?php
include 'rodape.php';
?>
```

APÊNDICE H – CÓDIGO PARA GERAR GRÁFICOS

```

<?php
require 'phplot.php';
function mysql_prepare ($query, $phs = array()) {
    foreach ($phs as $ph) {
        $ph = "" . mysql_real_escape_string($ph) . "";
        $query = substr_replace(
            $query, $ph, strpos($query, '?'), 1
        );
    }
    return mysql_query($query);
}

if (empty($_GET["tipo"])) {
    $tipo = "003e";
} else {
    $tipo = $_GET["tipo"];
}

if (empty($_GET["data"])) {
    $data = date('Y-m-d');
} else {
    $data=implode("-",array_reverse(explode("-",$_GET['data'])));
}

if (empty($_GET["hora1"])) {
    $hora1 = '00:00:00';
} else {
    $hora1 = $_GET["hora1"];
}

if (empty($_GET["hora2"])) {

```

```

        $hora2 = '23:59:59';
    } else {
        $hora2 = $_GET["hora2"];
    }

    $data_br = implode('/', array_reverse(explode('-', $data)));

    // Cria conexão com mysql
    $servername = "localhost";
    $username = "root";
    $password = "123456";
    $dbname = "pg";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "SELECT * FROM valores WHERE timestamp BETWEEN '$data ' . "$hora1' AND
    '$data ' . "$hora2' AND reg_id='$tipo'";

    $result = $conn->query($sql);

    $dados=array();
    $indexcc=0;

    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            $ccarga[$indexcc]['timestamp'] = date('H:i', strtotime($row['timestamp']));
            //echo "<p>Time {$indexcc} {$ccarga[$indexcc]['timestamp']}";
            $ccarga[$indexcc]['valor'] = $row['val'];
        }
    }

```

```

        if ($ccarga[$indexcc]['valor'] == '0') {
            $ccarga[$indexcc]['valor'] = "";
        }
        //echo " Valor: {$ccarga[$indexcc]['valor']}</p>";
        $indexcc+=1;
    }
    $erro = false;
} else {
    echo 'Erro';
    $erro = true;
}
for($i=0;$i<$indexcc;$i++){
    array_push($dados,array($ccarga[$i]['timestamp'],$ccarga[$i]['valor']));
}
if($tipo=="0042") {
//Demanda aparente
$plot = new PHPlot(1024, 768);
$plot->SetImageBorderType('plain');
$plot->SetPlotType('bars');
$plot->SetShading(0);
$plot->SetDataType('text-data');
$plot->SetDataValues($dados);
$plot->SetDataColors('YellowGreen');
$plot->SetFont('x_label', 2, [10]);
$plot->SetFont('y_label', 2, [8]);
$plot->SetTitle("Grafico da Demanda de Potencia Aparente do dia $data_br de $hora1 as
$hora2");
$plot->SetXTitle("Tempo - 15 em 15 minutos");
$plot->SetYTitle("Demanda de Potencia Aparente (VA)");
$plot->SetYDataLabelPos('plotin');
$plot->SetYDataLabelAngle(90);

$plot->SetDrawXDataLabelLines(True);

```

```
$plot->SetYTickLabelPos('plotleft');
$plot->SetYTickPos('none');
$plot->SetDrawYGrid(True);

$plot->SetXTickPos('none');
$plot->SetXTickLabelPos('none');
$plot->SetXLabelAngle(90);

        if(!$erro) {
                $plot->DrawGraph();
        }
}

else if($tipo=="003e") {
//Demanda ativa
$plot = new PHPlot(1024, 768);
$plot->SetImageBorderType('plain');
$plot->SetPlotType('bars');
$plot->SetShading(0);
$plot->SetDataType('text-data');
$plot->SetDataValues($dados);
$plot->SetDataColors('YellowGreen');
$plot->SetFont('x_label', 2, [10]);
$plot->SetFont('y_label', 2, [8]);
$plot->SetTitle("Grafico da Demanda de Potencia Ativa do dia $data_br de $hora1 as
$hora2");
$plot->SetXTitle("Tempo - 15 em 15 minutos");
$plot->SetYTitle("Demanda de Potencia Ativa (W)");

$plot->SetYDataLabelPos('plotin');
$plot->SetYDataLabelAngle(90);
```

```

$plot->SetDrawXDataLabelLines(True);

$plot->SetYTickLabelPos('plotleft');
$plot->SetYTickPos('none');
$plot->SetDrawYGrid(True);

$plot->SetXTickPos('none');
$plot->SetXTickLabelPos('none');
$plot->SetXLabelAngle(90);
    if(!$erro) {
        $plot->DrawGraph();
    }
}
else if($tipo=="002e") {
//Fator de Potencia Linha 1
$plot = new PHPlot(1024, 768);
$plot->SetImageBorderType('plain');
$plot->SetPlotType('lines');
$plot->SetShading(0);
$plot->SetDataType('text-data');
$plot->SetDataValues($dados);
$plot->SetDataColors('YellowGreen');
$plot->SetFont('x_label', 2, [10]);
$plot->SetFont('y_label', 2, [8]);
$plot->SetTitle("Gráfico da Demanda do Fator de Potencia da Linha 1 do dia $data_br de
$hora1 as $hora2");
$plot->SetXTitle("Tempo - 15 em 15 minutos");
$plot->SetYTitle("Fator de Potencia da Linha 1");

$plot->SetYDataLabelPos('plotin');
$plot->SetYDataLabelAngle(90);
$plot->SetDrawXDataLabelLines(True);

```

```
$plot->SetYTickLabelPos('plotleft');
$plot->SetYTickPos('none');
$plot->SetDrawYGrid(True);

$plot->SetXTickPos('none');
$plot->SetXTickLabelPos('none');
$plot->SetXLabelAngle(90);
    if(!$erro) {
        $plot->DrawGraph();
    }
}
else if($tipo=="0030") {
//Fator de Potencia Linha 2
$plot = new PHPlot(1024, 768);
$plot->SetImageBorderType('plain');
$plot->SetPlotType('lines');
$plot->SetShading(0);
$plot->SetDataType('text-data');
$plot->SetDataValues($dados);
$plot->SetDataColors('YellowGreen');
$plot->SetFont('x_label', 2, [10]);
$plot->SetFont('y_label', 2, [8]);
$plot->SetTitle("Gráfico do Fator de Potencia da Linha 2 do dia $data_br de $hora1 as
$hora2");
$plot->SetXTitle("Tempo - 15 em 15 minutos");
$plot->SetYTitle("Fator de Potencia da Linha 2");

$plot->SetYDataLabelPos('plotin');
$plot->SetYDataLabelAngle(90);
$plot->SetDrawXDataLabelLines(True);

$plot->SetYTickLabelPos('plotleft');
$plot->SetYTickPos('none');
```

```

$plot->SetDrawYGrid(True);

$plot->SetXTickPos('none');
$plot->SetXTickLabelPos('none');
$plot->SetXLabelAngle(90);
    if(!$erro) {
        $plot->DrawGraph();
    }
}
else if($tipo=="0032") {
//Fator de Potencia Linha 3
$plot = new PHPlot(1024, 768);
$plot->SetImageBorderType('plain');
$plot->SetPlotType('lines');
$plot->SetShading(0);
$plot->SetDataType('text-data');
$plot->SetDataValues($dados);
$plot->SetDataColors('YellowGreen');
$plot->SetFont('x_label', 2, [10]);
$plot->SetFont('y_label', 2, [8]);
$plot->SetTitle("Grafico do Fator de Potencia da Linha 3 do dia $data_br de $hora1 as
$hora2");
$plot->SetXTitle("Tempo - 15 em 15 minutos");
$plot->SetYTitle("Fator de Potencia da Linha 3");

$plot->SetYDataLabelPos('plotin');
$plot->SetYDataLabelAngle(90);
$plot->SetDrawXDataLabelLines(True);

$plot->SetYTickLabelPos('plotleft');
$plot->SetYTickPos('none');
$plot->SetDrawYGrid(True);

```



```

$plot->SetXTickPos('none');
$plot->SetXTickLabelPos('none');
$plot->SetXLabelAngle(90);
    if(!$erro) {
        $plot->DrawGraph();
    }
}
else if($tipo=="0006") {
//Fator de Potencia Trifasico
$plot = new PHPlot(1024, 768);
$plot->SetImageBorderType('plain');
$plot->SetPlotType('lines');
$plot->SetShading(0);
$plot->SetDataType('text-data');
$plot->SetDataValues($dados);
$plot->SetDataColors('YellowGreen');
$plot->SetFont('x_label', 2, [10]);
$plot->SetFont('y_label', 2, [8]);
$plot->SetTitle("Gráfico do Fator de Potencia Trifasico do dia $data_br de $hora1 as
$hora2");
$plot->SetXTitle("Tempo - 15 em 15 minutos");
$plot->SetYTitle("Fator de Potencia Trifasico");

$plot->SetYDataLabelPos('plotin');
$plot->SetYDataLabelAngle(90);
$plot->SetDrawXDataLabelLines(True);

$plot->SetYTickLabelPos('plotleft');
$plot->SetYTickPos('none');
$plot->SetDrawYGrid(True);

$plot->SetXTickPos('none');
$plot->SetXTickLabelPos('none');

```

```
$plot->SetXLabelAngle(90);  
  if(!$erro) {  
    $plot->DrawGraph();  
  }  
}  
$conn->close();  
?>
```