

**ALISSON RODRIGUES ROCIO**

**MEDIÇÃO DE ENERGIA DA UNIVERSIDADE FEDERAL DE LAVRAS  
ATRAVÉS DO SOFTWARE LIVRE MRTG**

Monografia de graduação apresentada ao Departamento de  
Ciência da Computação da Universidade Federal de Lavras  
como parte das exigências do curso de Ciência da  
Computação para obtenção do título de Bacharel em Ciência  
da Computação.

LAVRAS  
MINAS GERAIS – BRASIL  
2005

**ALISSON RODRIGUES ROCIO**

**MEDIÇÃO DE ENERGIA DA UNIVERSIDADE FEDERAL DE LAVRAS  
ATRAVÉS DO SOFTWARE LIVRE MRTG**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Orientador:

Prof. Rêmulo Maia Alves

Co-orientador:

Prof. Anderson Bernardo dos Santos

LAVRAS  
MINAS GERAIS – BRASIL  
2005

**ALISSON RODRIGUES ROCIO**

**MEDIÇÃO DE ENERGIA DA UNIVERSIDADE FEDERAL DE LAVRAS  
ATRAVÉS DO SOFTWARE LIVRE MRTG**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 30 de Junho de 2005.

---

Joaquim Paulo da Silva

---

Luciano Mendes dos Santos

---

Anderson Bernardo dos Santos (Co-orientador)

---

Rêmulo Maia Alves (Orientador)

LAVRAS  
MINAS GERAIS – BRASIL  
2005

## **AGRADECIMENTOS**

Agradeço a Deus

Aos meus pais e familiares

A todos aqueles que colaboraram para a conclusão desse trabalho

E como dizia mestre Yoda

“Do, or do not .There is no try”

## **MEDIÇÃO DE ENERGIA DA UNIVERSIDADE FEDERAL DE LAVRAS ATRAVÉS DO SOFTWARE LIVRE MRTG**

**Resumo.** O objetivo desse trabalho consiste em fazer a adaptação do software livre MRTG para fazer a geração de gráficos de como está sendo o consumo de energia elétrica dos departamentos da Universidade Federal de Lavras. Para a aquisição dos dados da rede elétrica foi usado um medidor da empresa Yokogawa que tem a função de ler a potência elétrica da rede, e disponibilizar esses dados na forma analógica. De posse desses dados analógicos foi construído um conversor A/D (Analógico/Digital) que tem a função de transformar aquele sinal analógico em sinais digitais, que serão usados pelo MRTG para a geração de gráficos do consumo elétrico do local.

**Palavras chaves.** MRTG, medição, energia, conversor, ADC0804, analógico, digital.

## **MEANSUREMENT OF ENERGY OF THE FEDERAL UNIVERSITY OF LAVRAS THROUGH FREE SOFTWARE MRTG**

**Abstract.** The objective of this work consists of making the adaptation of free software MRTG to make the generation of graphs of as it is being the consumption of electric energy of the departments of the Federal University of Lavras. For the acquisition of the data of the electric net a measurer of the Yokogawa company was used who has the function to read the electric power of the net, and to make available these data in the analogical form. Of ownership of these analogical data was constructed a converter A/D (Analogical/Digital) that it has the function to transform that analogical signal into digital signals, that will be used for the MRTG for the generation of graphs of the electric consumption of the place.

**Keyword.** MRTG , measurement , power , converter , ADC0804 , analogical , digital.

# Sumário

<b>CAPÍTULO 1</b>	<b>4</b>
Introdução	4
1.1 - Considerações Iniciais	4
1.2 - Objetivos e Justificativas	4
1.3 - Escopo do Trabalho	5
<b>CAPÍTULO 2</b>	<b>7</b>
Referencial Teórico	8
2.1 -MRTG	9
2.1.1 - Servidor Web	9
2.1.2 - Apache	10
2.1.3 - TCP/IP e o Apache	10
2.1.4 - HTTP	13
2.2 - Conversor Analógico - Digital	16
2.2.1- Especificações de um conversor A/D	17
2.3 - Circuito integrado ADC0804 oito bits	18
2.3.1- Resolução	19
2.4 - Porta Paralela	20
2.4.1 - Introdução	20
2.4.2 - Modelos de porta paralela	20
2.4.3 - Descrição da porta paralela	21
2.5 - Medidores	22
2.6 - Linhas de Alimentação	24
2.7 - Consumo elétrico	25
2.7.1 - Potência elétrica	25
2.7.2 - Potência em Sistemas de Energia Elétrica	26
<b>CAPÍTULO 3</b>	<b>29</b>
Metodologia	29
3.1 – Tipo de Pesquisa	29
3.2 - Quanto aos procedimentos técnicos	29
3.3 - Implementação e Construção	29
<b>CAPÍTULO 4</b>	<b>31</b>
Resultados e Discussões	31
4.1 - Conversor	31
4.1.1 - Processo de aquisição de dados	32
4.2 - Implementação	33
4.2.1 - Linux e a porta paralela	33
4.3 - Configuração do MRTG e do Apache	34
4.3.1 - Configurando o Apache	35
4.3.2 - Configurando o MRTG	35
4.4 - Monitor multi-função YOKOGAWA	38
4.5 - Integração dos componentes	40
4.5.1 - Preparação	40
4.6 - Resultados	40

CAPÍTULO 5	47
Conclusão	47
CAPÍTULO 6	48
Propostas Futuras	48

## Lista de Figuras

Ilustração 1 - Gráfico gerado pelo MRTG	7
Ilustração 2 - Classes de ips	11
Ilustração 3 - Funcionamento do servidor web	13
Ilustração 4 - Conversor Analógico Digital	16
Ilustração 5 - ADC0804	19
Ilustração 6 - Porta Paralela	21
Ilustração 7 - Medidor de ponteiros	23
Ilustração 8 - Medidor ciclométrico	23
Ilustração 9 - Relação entre energia e potência	26
Ilustração 10 - Circuito elétrico de corrente alternada	27
Ilustração 11 - Gráfico de corrente alternada	27
Ilustração 12 - Potência ativa	27
Ilustração 13 - Potência reativa	28
Ilustração 14 - Montagem ADC0804	32
Ilustração 15 - Dado com valor zero	37
Ilustração 16 - Página do MRTG	38
Ilustração 17 - Medidor Yokogawa Multifunção	39
Ilustração 18 - Painel traseiro	40
Ilustração 19 - Ligação medidor à rede elétrica	41
Ilustração 20 - Ligação medidor ao conversor	41
Ilustração 21 - Ligação conversor à porta paralela	42
Ilustração 22 - Ligação completa	43
Ilustração 23 - Gráfico diário	46
Ilustração 24 - Gráfico semanal mensal e anual	46

## Lista de Tabelas

Tabela 1 - Métodos do HTTP	14
Tabela 2 - Códigos de status	15
Tabela 3 - Resolução	20
Tabela 4 - Valores testados I	44
Tabela 5 - Valores testados II	44

# Capítulo 1

## Introdução

### 1.1 - Considerações Iniciais

Sendo uma das mais importantes formas de energia, a energia elétrica é parte integrante da vida moderna do homem e indispensável ao progresso econômico do mundo, contribuindo para a satisfação das necessidades e desejos do consumidor, possibilitando-lhe uma enorme variedade de serviços que por sua vez, causa a conseqüente necessidade por mais e mais energia. Contudo, o aumento na oferta de energia se dá através de altos investimentos em centrais de produção sendo necessário um detalhado projeto de expansão.

O planejamento de expansão de um sistema de geração de energia elétrica deve então considerar aspectos como os custos (de investimento e de operação), bem como o impacto ambiental e outras questões de natureza técnica e econômica. Diante da evolução dos dispositivos de medição visto que há poucos anos tínhamos apenas os medidores eletromecânicos com apenas o registro de uma grandeza, hoje temos medidores eletrônicos capazes de medir diversas grandezas simultaneamente.

E esse processo de evolução tem caminhado para a comunicação por *Internet Protocol* (IP) e ao considerarmos a convergência das áreas funcionais do gerenciamento de redes e do gerenciamento de cargas em redes de distribuição, podemos pressupor a possibilidade de utilização de técnicas de gerência de redes aplicadas à gerência de carga. Assim, é em função desta possibilidade que este trabalho se justifica, propondo a utilização do conversor para o sistema de gerenciamento, disponibilizando esses dados através do *Multi Router Traffic Grapher* (MRTG).

### 1.2 - Objetivos e Justificativas

O objetivo desse trabalho é fazer a medição do consumo de energia elétrica dos departamentos da Universidade Federal de Lavras UFLA usando para isso um medidor de consumo, um aparelho que faça a comunicação entre o relógio e o computador e o *software* MRTG. De posse desses instrumentos será feita medições periódicas do consumo elétrico de um determinado departamento. Essas medições serão mostradas pelo *software* MRTG que disponibilizara gráficos em tempo real sendo possível ter o consumo em intervalos de tempo de dias, mês, e anos, onde será possível fazer uma leitura certa e precisa do consumo. O controle da energia elétrica é de grande importância já que o mundo moderno de hoje gira em torno dessa energia.

Com esse trabalho temos o intuito de fazer um controle maior e mais eficaz do consumo elétrico dos departamentos da Universidade Federal de Lavras UFLA analisando os horários de picos onde ocorre o maior consumo e também detectar se aquele local em questão está consumindo energia além da conta. De posse desses dados podemos tomar providências como verificar se temos alguma “fuga” de energia que esteja levando a esse consumo exagerado, ou outras providências que levem um melhor aproveitamento desse recurso.

### 1.3 - Escopo do Trabalho

Nos próximos capítulos temos informações que darão embasamento para o trabalho em questão. Na primeira seção do capítulo dois será analisado o *software* MRTG, dando informações de como ele apresentará os dados analisados e a geração de gráficos. Será visto como é o seu funcionamento e do que é preciso para que ele seja executado. Sendo assim vamos entrar na parte do servidor *web* que será um componente importante para disponibilizar os gráficos através da *internet*. Vamos ver como o servidor *web* trabalha e será falado um pouco da escolha de um servidor *web* específico para esse trabalho.

Também será falado sobre os protocolos *Transmission Control Protocol* (TCP) / *Internet Protocol* (IP) e o Apache. E de como esses protocolos interagem entre si e proporcionam que os dados medidos sejam disponibilizados através da *internet*. Na primeira seção do capítulo dois será visto como o Apache trabalha com o *Hyper Text Transfer Protocol* (HTTP) que é o protocolo padrão de transferência na *internet*.

Na segunda seção do capítulo dois será mostrado o método usado para fazer a aquisição de dados pelo computador.

A partir da quinta seção do capítulo dois será mostrado a parte de energia elétrica onde será visto alguns medidores de energia e de como a indústria de energia funciona no Brasil.

No capítulo três será visto como foi feito à pesquisa, ou seja, que métodos foram usados para a realização do projeto.

Por fim no capítulo quatro temos resultados e discussões onde serão apresentados detalhes dos resultados obtidos e todos os dados levantados.

No capítulo cinco será mostrado a conclusão e se os resultados obtidos foram alcançados.

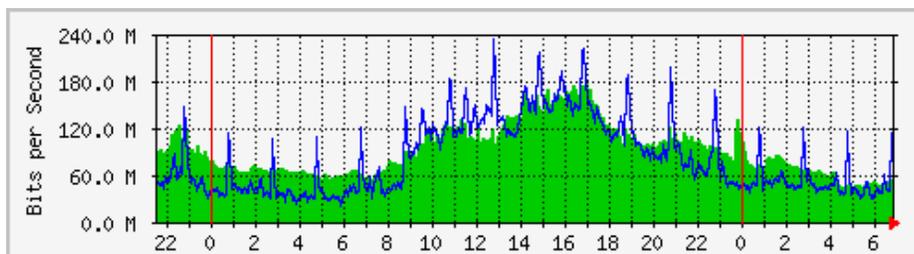
No capítulo seis temos propostas futuras para a continuação do trabalho.

## Capítulo 2

### Referencial Teórico

#### 2.1 - MRTG

O *Multi Router Traffic Grapher* (MRTG) foi escrito na linguagem *perl* primeiramente por Tobias Oetiker e Dave Rand [(Kretchmar, 2003)]. É uma ferramenta para monitorar a carga de tráfego em link de rede. Ele gera páginas HTML contendo imagens gráficas que possibilitam uma representação visual deste tráfego. O MRTG consiste em um script em *Perl* que usa o protocolo *Simple Network Management Protocol* (SNMP) para ler os contadores de tráfego de seus roteadores e um rápido programa em C que joga os dados do tráfego e cria belos gráficos representando o tráfego da conexão de rede monitorada [MRTG]. Os gráficos são criados no formato PNG (*Portable Network Graphics*) e podem ser incluídos em páginas *web* que podem ser visualizadas de qualquer navegador moderno [(Kretchmar, 2003)].



**Ilustração 1 - Gráfico gerado pelo MRTG**

O formato *Portable Network Graphics* (PNG) é um formato de dados utilizado para imagens, que surgiu em 1996 como substituto para o formato *Graphics Interchange Format* (GIF), devido ao fato de este último incluir algoritmos patenteados. O formato PNG permite comprimir as imagens sem haver perda de qualidade, ao contrário do que

acontece com outros formatos, como o *Joint Photographic Experts Group* (JPEG). A escolha do formato PNG se deu pelo fato de ser gratuito, possuir uma boa qualidade de imagem e seus arquivos são muito mais compactados que o seu concorrente, proporcionando que páginas que utilizam PNG carreguem mais rapidamente.

O MRTG não se limita a monitorar somente tráfego, é possível monitorar qualquer variável SNMP que você escolher. Você pode até usar um programa externo para pegar os dados que você deve monitorar via MRTG. As pessoas usam o MRTG, para monitorar coisas como Carga do Sistema, Sessões Logadas, Disponibilidade de modem e muito mais.

Através de scripts o MRTG pode monitorar outros equipamentos que não usam o protocolo SNMP. Os scripts para interface com o MRTG se caracterizam por sempre retornarem dois valores na saída padrão, um em cada linha, a cada vez que são executados. Estes valores correspondem às variáveis (sempre duas) monitoradas pelo script.

O MRTG usa scripts feitos em *Practical Extraction and Report Language* (PERL). A Linguagem Prática de Extração e Relatório é uma linguagem de script criada por Larry Wall em dezembro de 1987. A origem do Perl remonta ao shell scripting, Awk e à linguagem C e está disponível para praticamente todos os sistemas operacionais, mas é usado mais comumente em sistemas Unix e compatíveis.

Perl é uma preferência entre administradores de sistema e desenvolvedores *web*. Ela é excelente no que diz ao processamento de cadeias de caracteres (strings), manipulação de arquivos texto e utilização de expressões regulares, além de permitir tempos de desenvolvimento menores.

O MRTG é disponível sem custo sob os termos da GNU Licença Pública Universal. A GNU Licença Pública Universal pretende garantir a liberdade de compartilhar e alterar *softwares* gratuitos tornando-os de livre distribuição também para quaisquer usuários.

### **2.1.1 - Servidor Web**

Para disponibilizar os dados através do browser o MRTG é executado em cima de um servidor *web*. Um servidor *web* é um aplicativo responsável por fornecer ao computador do cliente (usuários de sites e páginas eletrônicas), em tempo real, os dados solicitados.

O processo se inicia com a conexão entre o computador onde está instalado o servidor e o computador do cliente; como na *web* não é possível prever a que hora se dará

essa conexão, os servidores precisam estar disponíveis dia e noite. A partir daí é processado o pedido do cliente, e conforme as restrições de segurança e a existência da informação solicitada, o servidor devolve os dados.

Quando falamos em servidor *web*, estamos na verdade falando de servidores capazes de lidar com o protocolo HTTP, que é o padrão para transmissão de hipertexto. Muitos servidores trabalham, paralelamente, com outros protocolos, como SHTTP (que é o HTTP "seguro"), File Transport Protocol (FTP) , etc. Além de transmitir páginas HTML, imagens e aplicativos, os servidores também podem executar programas e scripts, interagindo mais com o usuário.

## 2.1.2 - Apache

Apesar de haver vários tipos de servidores webs , vamos falar aqui sobre o mais usado, no caso o servidor Apache. Você tem que compreender que Apache está livre a seus usuários e é escrito por uma equipe de voluntários que não são pagos por seus trabalhos [(Laurie, B. (1997) - Laurie, P. (1997)].

Segundo Rob Hartill um dos integrantes do grupo, o Grupo Apache é uma coleção de indivíduos talentosos que estão tentando aperfeiçoar a arte de nunca terminar algo [(Engelschall, 2001)]. Essa descrição mostra que os membros dos grupos estão em um esforço conjunto para fornecer constantemente novas versões que melhorem a qualidade do programa.

O projeto de Apache é um esforço colaborativo do desenvolvimento de um *software* visado criar um código robusto, comercial e livre-disponível da fonte de um usuário do HTTP (*Web*). O projeto é controlado conjuntamente por um grupo de voluntários situados em torno do mundo, usando a *Internet* para comunicar, planejar, e desenvolver o servidor e sua documentação relacionada. Estes voluntários são denominados como o grupo Apache [APACHE].

O diretório local contém tipicamente quatro subdiretórios:

- *conf*

Contém os arquivos de configuração, do qual o `httpd.conf` é o mais importante.

- *htdocs*

Contem os *scripts* HTML que serão usados pelo cliente. Este diretório e todos aqueles abaixo dele, são acessíveis de qualquer lugar da *web*.

*logs*

Contêm *logs* de datas, informações de acessos e erros.

- *cgi-bin*

Contém *Computer Graphics Interface (CGI) scripts*. Estes são programas ou *shell scripts* escritos pelo webmaster que pode ser executado pelo Apache em nome de seus clientes. É o mais importante, por razões da segurança, este diretório não deve estar disponível na *Web*.

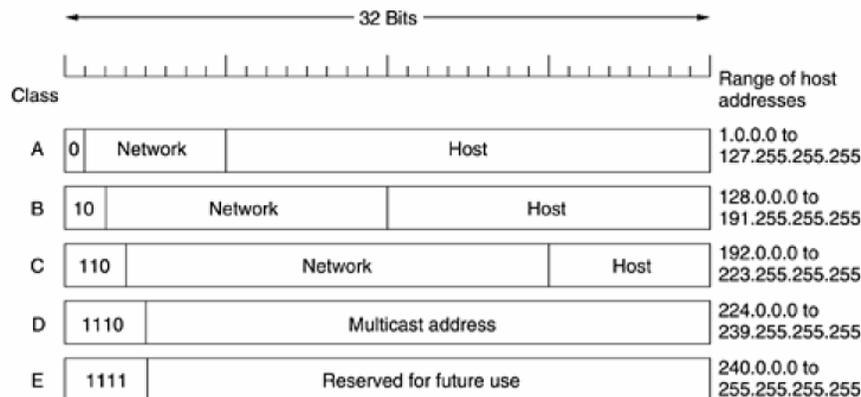
### 2.1.3 - TCP/IP e o Apache

O Apache utiliza o protocolo TCP/IP. Os endereços IP são números com 32 *bits*, escritos como quatro octetos (em decimal), ex: 127.0.0.1. A primeira parte do endereço identifica uma rede específica na inter-rede, a segunda parte identifica um *host* dentro dessa rede. Os endereços IP podem ser usados para nos referirmos tanto a redes quanto a um *host* individual. Por convenção, um endereço de rede tem o campo identificador de *host* com todos os *bits* iguais à zero. Podemos nos referir a rede através de um endereço por difusão. Um endereço com todos os 32 *bits* iguais a um é considerado um endereço por difusão para a rede do *host* origem do datagrama. O endereço 127.0.0.0 é reservado para teste (*loopback*) e comunicação entre processos da mesma máquina. O IP utiliza três classes diferentes de endereços.

Na primeira classe de endereços, classe A, o *bit* mais significativo é zero, os outros sete *bits* do primeiro octeto identificam a rede, e os 24 *bits* restantes definem o endereço local. Essa classe é usada para redes de grande porte, os endereços de rede variam de 1 a 126, e cada rede tem capacidade de endereçar cerca de 16 milhões de *hosts*.

Na classe B usa dois octetos para o número da rede e dois para endereços de *hosts*. Está na faixa de 128.0.0.0 a 191.255.255.255. Cada rede pode interligar cerca de 65 mil *hosts*.

Na classe C utilizam três octetos para identificar a rede e um para o *host*. Está na faixa de 192.0.0.0 a 223.255.255.255. Cada rede pode endereçar 254 *host*. Os endereços acima de 223 no primeiro octeto foram reservados para uso futuro.



**Ilustração 2 - Classes de ips**

Atualmente, há milhares de redes conectadas à *Internet* e esses números de redes são atribuídos pelo *Network Information Center* (NIC) que é um órgão responsável de distribuir esses endereços para evitar conflitos.

O TCP é um protocolo orientado à conexão que fornece um serviço confiável de transferência de dados fim a fim. O TCP foi projetado para funcionar com base em um serviço de rede sem conexão e sem confirmação. Ele foi projetado especificamente para oferecer um fluxo de *bytes* fim a fim confiável em uma inter-rede não-confiável. Uma inter-rede é diferente de uma única rede porque suas muitas partes podem ter topologias, larguras de banda, retardos, tamanhos de pacote e outros parâmetros completamente diferentes. O TCP foi projetado para se adaptar dinamicamente às propriedades da inter-redes e para ser robusto diante dos muitos tipos de falhas que podem ocorrer.

A camada IP não oferece qualquer garantia de que os datagramas serão entregues da forma apropriada; portanto, cabe à TCP administrar os temporizadores e retransmiti-los sempre que necessário. Os datagramas também podem chegar fora de ordem; o TCP também terá de reorganizá-los em mensagens na seqüência correta. O TCP deve fornecer a confiabilidade que a maioria dos usuários que, mas que o IP não oferece [Tanenbaum, 1997].

Para mostrarmos como o servidor *web* trabalha com o TCP/IP usaremos a seguinte analogia. Temos uma caixa em que há um computador, um *software*, e uma conexão ao mundo exterior. Está conexão é conhecida como *interface* e é conhecida para o mundo pelo seu endereço IP. O endereço IP é um protocolo usado pela fonte e *host* de destino para

comunicação de dados através de troca de pacotes dentro da rede. Os dados numa rede IP são enviados em blocos referidos como pacotes ou datagramas.

Os pedidos chegam a uma *interface* para números de diferentes serviços oferecidos pelo usuário usando protocolos diferentes.

- *Network News Transfer Protocol* (NNTP): notícias
- *Simple Mail Transfer Protocol* (SMTP): correio
- *Domain Name Service* (DNS): servidor de nomes
- HTTP: *World Wide Web*

O servidor pode decidir-se como segurar estes pedidos diferentes porque os quatro *bytes* do endereço IP que conduz o pedido para a *interface* é seguido pelo número da porta (dois - *bytes*). Diferentes serviços anexos para diferentes portas.

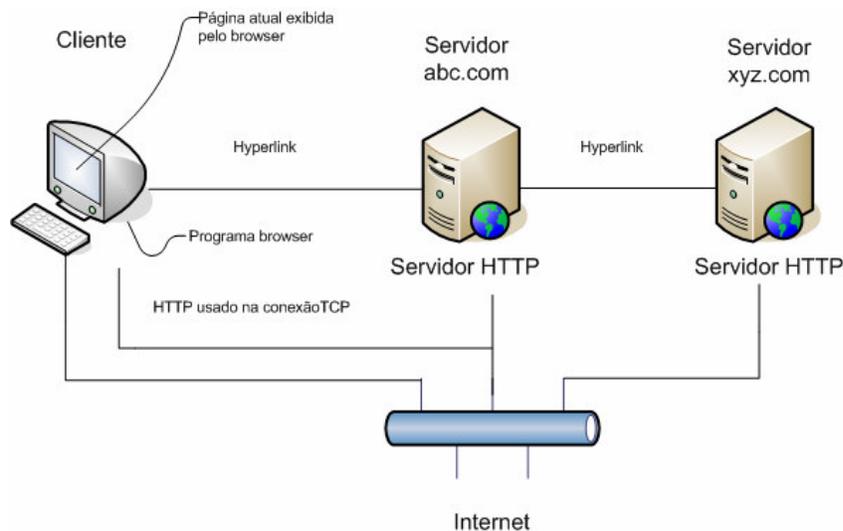
- NNTP: porta número 119
- SMTP: porta número 25
- DNS: porta número 53
- HTTP: porta número 80

Como o administrador ou o *webmaster* local, você pode decidir unir todo o serviço a qualquer porta. Naturalmente, se você se decidir pisar fora da convenção, você necessita certificar-se de que seus clientes compartilham de sua opinião. Nosso interesse aqui é apenas com HTTP e o Apache. Apache, por padrão, escuta a porta 80.

Os procedimentos que ocorrem desde o momento em que o usuário dá o clique e o momento em que a página é apresentada é o seguinte:

Como exemplo vai imaginar que o usuário acabou de dar um clique sobre um texto que aponte para a página cuja *Uniform Resource Locator* (URL) é <http://www.seila.org/exemplo/exemplo.html>. Uma URL possui três partes: o nome do protocolo (HTTP), o nome da máquina onde a página está localizada ([www.seila.org](http://www.seila.org)) e o nome do arquivo onde está a página (/exemplo/exemplo.html). Sendo assim os seguintes passos são executados:

- O *browser* determina o Universal Resource Locator (URL) (verificando qual foi selecionado).
- O *browser* pergunta ao *Dynamic Name System* (DNS) qual é o endereço IP de [www.seila.org](http://www.seila.org)
- O *Dynamic Name System* (DNS) responde 200.131.254.123
- O *browser* estabelece uma conexão TCP com a porta 80 em 200.131.254.123
- Em seguida, o *browser* envia um comando *GET /exemplo/exemplo.html*.
- O servidor [www.seila.org](http://www.seila.org) envia o arquivo exemplo.html.
- A conexão TCP é liberada.
- O *browser* apresenta todo o texto de exemplo.html
- O *browser* busca e apresenta todas as imagens de exemplo.html.



**Ilustração 3 - Funcionamento do Servidor Web**

## 2.1.4 - HTTP

O *Hyper Text Transfer Protocol* (HTTP) é o protocolo de transferência padrão da *WEB*. Consiste em dois itens razoavelmente distintos: um conjunto de solicitações dos *browsers* aos servidores e um conjunto de respostas que retornam no caminho inverso [(Tanenbaum, 1997)].

O HTTP aceita dois tipos de solicitações: simples e completas. Uma solicitação simples é apenas uma linha *GET* que identifica a página desejada, sem a versão do protocolo. A resposta é formada apenas pela página, sem cabeçalho, sem *Multipurpose Internet Mail Extensions* (MIME) e sem códigos. As solicitações completas são indicadas pela presença da versão do protocolo na linha de solicitação *GET*. As solicitações podem consistir em várias linhas, seguidas de uma linha em branco para indicar o fim da solicitação. A primeira linha de uma solicitação completa contém o comando (onde *GET* é apenas umas das possibilidades), a página desejada e o protocolo/versão. As linhas subsequentes contêm cabeçalhos. A primeira palavra na linha da solicitação completa é o nome do método (comando) a ser executado na página Web.

Os Métodos são os seguintes:

**Tabela 1 - Métodos do http**

Método	Descrição
<i>GET</i>	Solicita a leitura de uma página da Web
<i>HEAD</i>	Solicita a leitura de um cabeçalho de página Web
<i>PUT</i>	Solicita o armazenamento de uma página da Web
<i>POST</i>	Acrescenta a um recurso (uma página da Web)
<i>DELETE</i>	Remove a página da Web
<i>LINK</i>	Conecta dois recursos existentes
<i>UNLINK</i>	Desfaz uma conexão entre dois recursos

O método *GET* solicita ao servidor que envie a página (objeto), codificada no MIME. Se a solicitação for seguida de um cabeçalho *If-Modified-Since* (Modificada desde) o servidor só enviará os dados se eles tiveram sido alterados desde a data fornecida.

O método *HEAD* solicita apenas o cabeçalho da mensagem, sem a página propriamente dita. Esse método pode ser usado para obter a data da última modificação feita na página, para obter informações de índices, ou apenas para testar a validade de uma URL.

O método *PUT* é o inverso de *GET*: em vez de ler, ele grava a página. Esse método possibilita a criação de um conjunto de páginas da *Web* em um servidor remoto. O corpo da solicitação contém a página. Ela pode ser codificada usando o MIME. Os tipos MIME são usados pelos navegadores *Web* para saber como apresentar os dados enviados por um servidor *Web* para o utilizador. Assim as linhas depois de *PUT* podem conter *Content-Type*

e cabeçalhos de autenticação, para provar que o responsável pela chamada tem mesmo permissão para realizar a operação solicitada.

O *POST* é semelhante ao *PUT*. Ele também transporta a URL. No entanto, em vez de substituir os dados existentes, os novos dados são acrescentados.

O *DELETE* exclui a página. Nesse caso a permissão e a autenticação têm um papel fundamental. Não há garantia de que o *DELETE* tenha sido bem sucedido, pois mesmo que o servidor HTTP esteja pronto para excluir a página, o arquivo subjacente pode ter um modo que proíba o servidor HTTP de modificá-lo ou excluí-lo.

Os métodos *LINK* e *UNLINK* permitem que sejam estabelecidas conexões com páginas já existentes ou outros recursos.

Cada solicitação obtém uma resposta que consiste em uma linha de status e possivelmente informações adicionais. Os códigos de *status* são organizados nas seguintes categorias:

**Tabela 2 - Códigos de status**

Categoria	Descrição
Informativo	O pedido foi recebido e continua sendo processado. Nenhum corpo da entidade acompanha esta resposta
Sucesso	A requisição foi recebida com sucesso, compreendida e aceita.
Redirecionamento	Ações adicionais são necessárias para completar essa requisição
Cliente erro	A requisição contém erro de sintaxe ou a requisição não pode ser cumprida
Server erro	O servidor falhou para cumprir e validar a requisição

Além do HTTP temos o HTTPS que fornece a transferência de dados cifrados entre cliente e servidor (em ambas as direções) de uma forma segura e também permite ao próprio servidor autenticar-se perante o cliente. O HTTPS providencia serviços de segurança para transações HTTP. Durante as negociações entre o cliente e o servidor uma variedade de algoritmos é providenciada. Por exemplo, o utilizador pode selecionar se quer a pergunta e a resposta assinadas digitalmente, cifradas ou ambas. Qualquer mensagem pode ser assinada, autenticada, cifrada, ou qualquer combinação destas, incluindo nenhuma proteção. Os mecanismos de gestão de chaves incluem *passwords* e troca da chave pública.

O HTTPS é uma implementação segura do protocolo HTTP, sendo os dados transmitidos através de uma conexão cifrada. A porta TCP usado por norma para o

protocolo HTTPS é o 443. O protocolo HTTPS é normalmente utilizado quando se deseja evitar que a informação transmitida entre o cliente e o servidor seja visualizada por terceiros, como por exemplo no caso de compras *online*

## 2.2 – Conversor Analógico - Digital

O processo de conversão analógica digital consiste, basicamente, em entrarmos com a informação de forma analógica e recolhermos na saída essa mesma informação de forma digital.



**Ilustração 4 - Conversor Analógico Digital**

Freqüentemente torna-se necessário que certo dado obtido em um sistema físico seja conectado na forma digital. Esses dados surgem na forma elétrica analógica.

Existem os seguintes tipos de conversores Analógicos Digital:

- Paralelo
- Contador
- Aproximações Sucessivas
- Rampa Dupla
- Sigma Delta

No nosso trabalho utilizaremos o circuito *Analog-to-Digital Converter (ADC) 0804* que usa o método de aproximações sucessivas.

O conversor A/D por aproximações sucessivas é a técnica mais comumente usada, possuindo a vantagem de poder ser implementada por *software*.

*O conversor A/D do tipo aproximações sucessivas faz uma amostragem da tensão analógica de entrada, e depois compara com a tensão que cresce em incrementos até chegar ao valor da tensão de entrada. Este aumento incremental de voltagem é produzido por um "clock" que produz uma seqüência regular de impulsos que são*

*contados e convertidos em sinal analógico mediante um conversor analógico/digital. O sinal analógico resultante é comparado com a de entrada e quando resulta superior, os impulsos são cortados e a leitura do contador também, e, portanto, a leitura do contador é o equivalente digital da tensão de entrada analógica [A/D].*

## **2.2.1 - Especificações de um conversor A/D**

As especificações de um conversor A/D que normalmente são fornecidas pelos fabricantes podem ser enumeradas como:

### **A - Tensão Analógica de Entrada**

Esta especificação designa a máxima faixa de tensões analógicas de entrada permitidas.

Valores típicos são:

0 a 10 volts, etc.

### **B - Impedância de Entrada**

Os valores variam de 1k $\Omega$  a 1M $\Omega$ , dependendo do tipo de conversor A/D. A capacitância da entrada se situa na faixa das dezenas de picofarads.

### **C - Precisão do Conversor**

A precisão de um conversor A/D inclui o erro de quantização, o ruído do sistema digital incluindo o que está presente na tensão de referência (usado no conversor D/A), desvios de linearidade, etc. A precisão também inclui a soma de todas as outras fontes de erro. Valores típicos são da ordem de  $\pm 0,02\%$  da leitura de fim de escala. Conversores A/D de altíssima precisão, entretanto, podem ser adquiridos com precisões de 0,001% da leitura de fundo de escala. A precisão de um conversor geralmente determina o número de *bits* que podem ser utilizados. Como exemplo, considere um conversor com uma escala analógica cobrindo a faixa  $\pm 10$  volts. Se a precisão for de 0,02% do fundo de escala, o erro máximo devido a tal limitação de precisão é de 2mV. Para 9, 10, 11 e 12 *bits*, os erros de quantização (1/2 LSB) são 10, 5, 2,5 e 1,25 mV respectivamente. Há uma vantagem em

usarmos 10 *bits* no lugar de nove. Poderíamos ainda justificar o uso de 11 *bits*, mas 12 *bits* provavelmente não.

## D - Estabilidade

A precisão do sistema é geralmente dependente da temperatura. Como exemplo, se um sinal de 10 V é aplicado a 75°C, temos como resultado um erro de  $(20 \times 10^{-6}) \times (10) \times (75 - 25) = 10\text{mV}$ . Com um conversor A/D de 10 *bits*, o erro limita a resposta à de um dispositivo de nove *bits*.

## E - Tempo de Conversão

É o tempo necessário para se realizar a conversão completa. Os tempos típicos de conversão variam de 50s, para unidades de velocidade moderada, a 50ns para um dispositivo de alta velocidade.

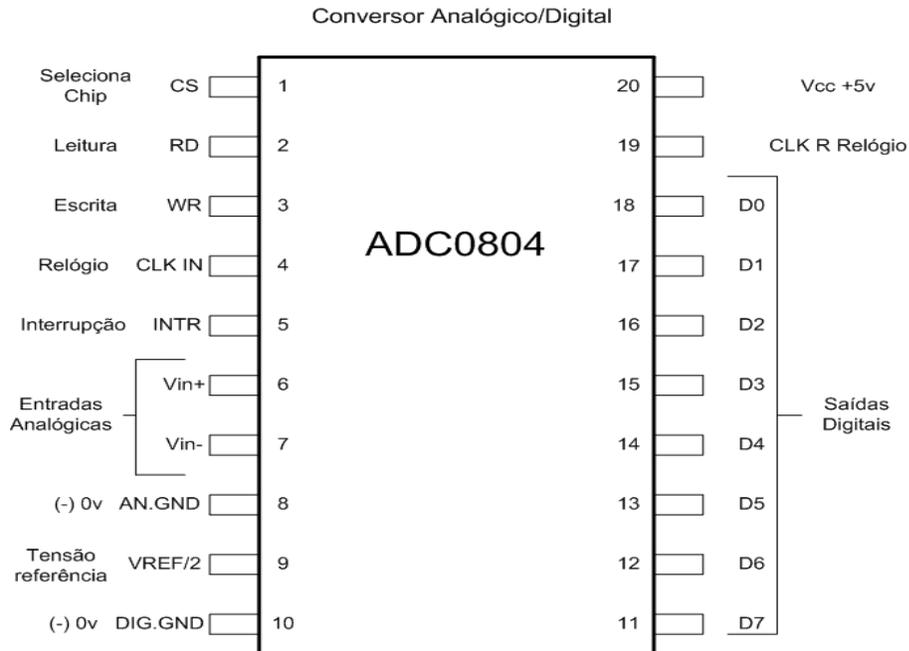
## F - Formatos

Um conversor A/D pode ser obtido praticamente para qualquer código usado em geral: binário unipolar, binário com desajuste, complemento de um, complemento de dois.

## 2.3 - Circuito Integrado ADC0804 oito *bits*

O conversor ADC0804 converte um sinal analógico em um correspondente dado digital de oito *bits* (D0 a D8), que pode ser interpretado e processado via *software*. O nível de tensão da entrada analógica pode variar de zero a cinco volts e o tempo de conversão é de  $100\mu\text{s}$ [ADC0804].

Na figura abaixo temos a imagem do diagrama do Circuito Integrado ADC0804.



**Ilustração 5 - ADC0804**

### 2.3.1 - Resolução

Para sabermos a resolução do conversor temos que saber o valor máximo que a entrada suporta e o tamanho máximo da saída em *bits*. No nosso caso temos o ADC0804 que possui uma entrada máxima de cinco *volts* e oito *bits*, sendo assim efetuamos a seguinte conta:

$$Resolução = \frac{ValorAnalógico}{2^8}$$

Onde temos  $2^8$  que representa a base binária elevada ao número total de *bits* do conversor.

$$Resolução = \frac{5v}{256}$$

Assim temos nossa entrada analógica de no máximo cinco *volts* que é a entrada máxima permitida pelo ADC0804.

Resolução = 0,0195v ou 19,5mv

**Tabela 3 – Resolução**

<i>Volts</i>	Decimal	Binário
0,0195	1	00000001
0,0390	2	00000010
0,0585	3	00000011
0,0780	4	00000100
...	...	...
4,9920	255	11111111

Podemos ver assim que a precisão da conversão será estipulada pela quantidade de *bits* presente no conversor. Assim se usarmos um conversor com um número maior de *bits* terá um valor bem maior no denominador que fará com que minha resolução seja cada vez menor.

## 2.4 - Porta Paralela

### 2.4.1 - Introdução

A porta paralela é uma *interface* de comunicação entre o computador e um periférico qualquer. Quando a *International Business Machines* (IBM) criou seu primeiro *Personal Computer* (PC) ou Computador Pessoal, a idéia era conectar a essa porta uma impressora, mas atualmente, são vários os periféricos que se utilizam desta porta para enviar e receber dados para o computador (exemplos: *scanners*, câmeras de vídeo, unidade de disco removível e outros).

A porta paralela é popular, pois é versátil e você pode usá-la para a saída e entrada de dados, e porque ela está disponível em todos os pcs.

### 2.4.2 - Modelos de Porta Paralela

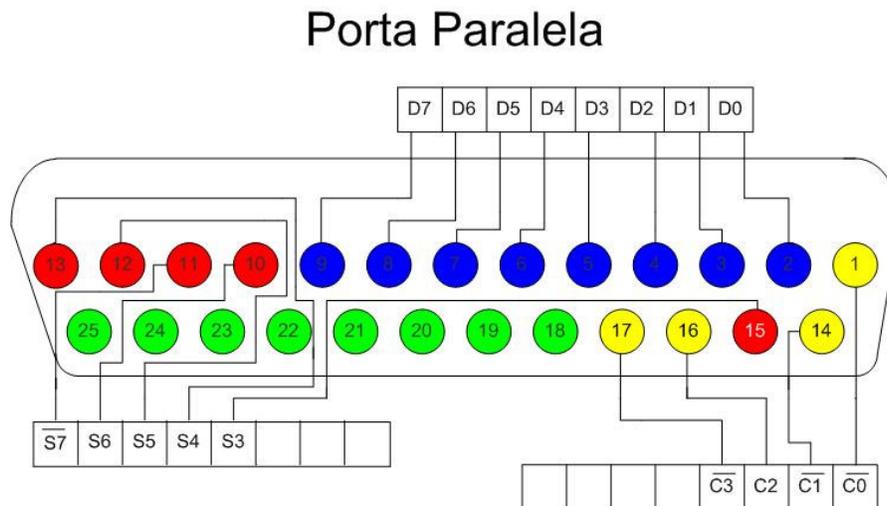
Existem dois modelos de transmissão de dados pela porta paralela. Temos a transmissão Unidirecional e Bidirecional.

No modo Unidirecional como o próprio nome diz transmite a informação em apenas um sentido. Temos assim a porta paralela *Standard Parallel Port* (SPP) que pode chegar a uma taxa de transmissão de dados a 150KB/s. Comunica-se com a *Central Processing Unit* (CPU) utilizando um barramento de dados de oito *bits* e para a transmissão de dados entre periféricos são usados quatro *bits* por vez [Axelson , 1997].

No modo Bidirecional podemos tanto receber como enviar informação pela porta. A porta avançada *Enhanced Parallel Port* (EPP) chega a atingir uma taxa de transferência de dois MB/s. Para atingir essa velocidade, será necessário um cabo especial. Comunica-se com a CPU utilizando um barramento de dados de 32 *bits* e para a transmissão de dados entre periféricos são usados oito *bits* por vez [Axelson , 1997]. A porta avançada *Enhanced Capabilities Port* (ECP) tem as mesmas características que a EPP, porém, utiliza DMA (acesso direto à memória), sem a necessidade do uso do processador, para a transferência de dados [Axelson , 1997].

### 2.4.3 - Descrição da Porta Paralela

Na figura abaixo temos uma visão da porta paralela e uma descrição da função de cada pino.



**Ilustração 6 - Porta Paralela**

A porta paralela possui 25 pinos sendo eles divididos em três grupos principais. Os pinos de *status* que compreende os pinos 15, 13, 12, 11,10 seguidos respectivamente dos seus seguintes códigos  $S3$ ,  $S4$ ,  $S5$ ,  $\overline{S7}$ ,  $S6$ . Os pinos de dados se iniciam no pino dois até o pino nove e seus códigos são  $D0$  até  $D7$ . E temos também os pinos de controle que são os pinos 1, 14, 16,17 e seus códigos são  $\overline{C0}$ ,  $\overline{C1}$ ,  $C2$ ,  $\overline{C3}$ . Os pinos restantes de cor verde 18 ao 25 são GND (*Ground* - Terra).

No modo SPP que é o modo unidirecional os pinos dois ao nove são pinos de saída de dados e eles podem ser usados apenas para enviar *bits* para algum dispositivo externo. No caso podem ser enviados oito *bits* por vez. Os pinos de controle também são de saída de dados e servem para controle de dispositivos. E os pinos de status servem para entradas de dados na porta paralela. Assim temos a possibilidade de enviar cinco *bits* por vez para dentro do computador.

Agora no modo EPP que é o modo bidirecional nos pinos dois ao nove podemos tanto enviar como receber informação para a porta. Esse modo é muito usado para transferir informação do mundo externo para dentro do computador, pois assim em vez de cinco *bits* do modo SPP podemos enviar oito *bits* de uma vez só. Os pinos de *status* e de controle continuam os mesmos no modo EPP.

## 2.5 - Medidores

O mercado de medidores de energia elétrica é hoje um dos mais concorridos em todo o mundo. As concessionárias controlam o consumo de energia elétrica de sua empresa por meio de verificações mensais feitas nos medidores (relógio de luz), que são instalados em seu estabelecimento. Periodicamente, a Empresa distribuidora acessa o local onde está o medidor e verifica o registro do consumo existente.

O leiturista utiliza um aparelho coletor de dados, com informações cadastrais aos clientes da região para digitar a leitura registrada no medidor. A leitura, digitada no coletor, é comparada automaticamente com os dados dos meses anteriores.

Se a leitura digitada for muita alta ou muita baixa em relação aos dados dos meses anteriores, o coletor emite um sinal. Neste caso, é necessário confirmar a leitura, digitando-a novamente para assegurar a qualidade do serviço.

Concluída a coleta de leituras, o coletor é levado para o escritório da Empresa, onde as leituras são transferidas para um microcomputador. No micro, o consumo a ser faturado

é calculado e enviado para o computador central da Empresa, onde as contas são impressas.

Existem dois tipos de medidores: ciclométrico, ponteiros. Deste valor de leitura, subtrai-se o valor da leitura anterior, obtendo-se o consumo registrado no período (normalmente um mês).

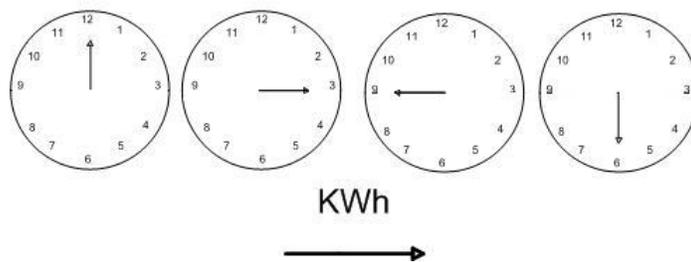
$$(\text{leitura atual}) - (\text{leitura anterior}) = \text{consumo do período}$$

No medidor ciclométrico o registro do consumo é numérico e já aparece diretamente no visor. Neste tipo de medidor, a leitura é simples e direta. Basta copiar os números que aparecem no visor. Neste caso, para saber o consumo, é só subtrair a leitura que aparece na sua última conta como "leitura atual" da que você fez agora.

O medidor de ponteiros possui quatro ou cinco relógios, com numeração de zero a nove. Observe o sentido de deslocamento dos ponteiros. Os ponteiros da unidade e da centena giram da esquerda para a direita (sentido horário). Os ponteiros da dezena e milhar giram da direita para a esquerda (sentido anti-horário). Para cada volta do ponteiro do relógio da direita, o ponteiro do relógio da esquerda anda uma unidade.

Para obter a leitura deve-se anotar a posição dos ponteiros e considerar o menor número.

### Medidor de Ponteiros



**Ilustração 7 - Medidor de Ponteiros**

### Medidor Ciclométrico



**Ilustração 8 - Medidor Ciclométrico**

## 2.6 - Linhas de Alimentação

Temos três fases em todo o caminho que a energia elétrica leva para chegar às casas dos consumidores. Essas três fases são a geração, transmissão e a distribuição.

A primeira delas é a geração onde como o próprio nome já diz é onde todo processo começa e onde a energia é gerada. Existem várias formas de se gerar energia elétrica. Temos a térmica, nuclear, hídrica, solar e eólica.

Vamos falar apenas daquela que representa a maior parcela de geração de energia no Brasil, que é a hídrica.

A energia potencial de uma queda d'água é usada para acionar turbinas que, por sua vez, acionam geradores elétricos. Essas quedas d'água são artificialmente construídas em formas de barragens, formando extensos reservatórios, necessários para garantir o suprimento em períodos de pouca chuva.

A segunda parte é a transmissão. Muitas vezes, a geração de energia elétrica ocorre em locais distantes dos centros consumidores. No caso predominante no Brasil a natureza impõe os locais onde sejam viáveis as construções das barragens. São comuns usinas geradoras distantes centenas ou milhares de quilômetros dos grandes centros.

Após o gerador, transformadores da subestação elevadora aumentam a tensão para um valor alto. Dependendo da região, pode variar de 69 a 750 kV. Finda a linha transmissão, transformadores de uma subestação redutora diminuem a tensão para um valor de distribuição.

No caso da energia elétrica, se transmitida com baixas tensões na potência necessária para atender milhares de consumidores, a bitola dos condutores precisaria ser tão grande que tornaria o sistema economicamente inviável.

Usinas normalmente dispõem de vários conjuntos turbina-gerador que trabalham em paralelo. As transmissões de diferentes usinas e diferentes centros consumidores são interligados de forma a garantir o suprimento em caso de panes e outros problemas.

A terceira e última parte é a de distribuição. Uma rede de distribuição deve fazer a energia chegar até os consumidores de forma mais eficiente possível.

Quanto mais alta a tensão menor a bitola dos condutores para transmitir a mesma potência. Assim, redes de distribuição em geral operam com, no mínimo, duas tensões. As mais altas para os consumidores de maior porte e as mais baixas para os pequenos.

A subestação redutora diminui a tensão da linha de transmissão para 13,8 kV, chamada distribuição primária, que é o padrão geralmente usado nos centros urbanos no Brasil. São aqueles três fios que se vê normalmente no topo dos postes. Essa tensão primária é fornecida aos consumidores de maior porte os quais, por sua vez, dispõem de suas próprias subestações para rebaixar a tensão ao nível de alimentação dos seus equipamentos.

A tensão primária também alimenta aqueles transformadores localizados nos postes que reduzem a tensão ao nível de ligação de aparelhos elétricos comuns (127/220 V), para consumidores de pequeno porte. É a chamada distribuição secundária.

A rede é formada pelos quatro fios (separados e sem isolação ou juntos e com isolação) que se observam na parte intermediária dos postes. É evidente que uma distribuição simples assim é típica de uma cidade de pequeno porte. Cidades maiores podem ser supridas com várias linhas de transmissão, dispondo de várias subestações redutoras e estas podem conter múltiplos transformadores, formando assim várias redes de distribuição. Também pode haver várias tensões de distribuição primária. Indústrias de grande porte, consumidoras intensivas de energia elétrica, em geral são supridas com tensões bastante altas, às vezes a da própria transmissão, para evitar altos custos da rede.

## 2.7 - Consumo elétrico

Para calcular o consumo de energia de algum aparelho eletrônico, primeiro verifique a potência em watts do aparelho; em seguida multiplique a potência encontrada pelo número de horas em que o aparelho foi utilizado no mês. Para isso, aplique a seguinte expressão:

$$\text{Consumo (kWh)} = \frac{\text{Potência (W)} \times \text{Horas/Dias} \times \text{Dias/Mês}}{1.000}$$

### 2.7.1 - Potência elétrica

Um Sistema de Energia Elétrica fornece energia aos consumidores que a utilizam, assegurando o nível de potência que estes requerem para alimentar o conjunto dos equipamentos ligados – a carga do sistema. A energia e a potência são contabilizadas separadamente na fatura de eletricidade, já que os custos associados a uma e outra são diferentes.

A relação básica entre energia e potência exprime-se matematicamente por:

$$P = \frac{dE}{dt}$$

### **Ilustração 9 - Relação entre Energia e Potência**

Onde  $E$  denota a energia,  $P$  a potência e  $t$  o tempo [(Paiva, 2004)]. Em unidades do SI (Sistema Internacional), a energia exprime-se em Joule (J) e a potência em Watt (W). Estas unidades correspondem a quantidades muito pequenas.

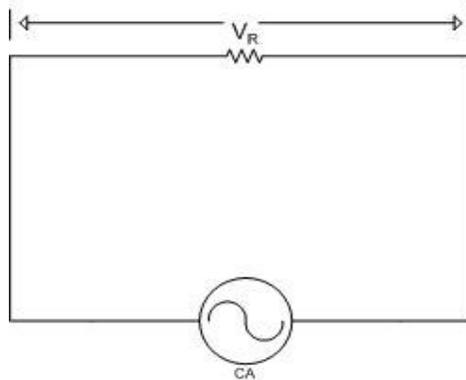
Assim, para a potência usam-se Kilowatt (1kW=10<sup>3</sup> W), Megawatt (1 MW=10<sup>6</sup> W), Gigawatt (1 GW=10<sup>9</sup> W) e Terawatt (1 TW=10<sup>12</sup> W). Para a energia, a unidade utilizada na prática é Watt/hora (1 Wh=3 600 J) e os seus múltiplos: kWh, MWh, GWh e TWh.

### **2.7.2 - Potência em Sistemas de Energia Elétrica**

Os sistemas de energia elétrica atualmente existente funcionam, na sua quase totalidade, em corrente alternada – com uma frequência de 60 Hz. [(Serway, 1996)].

Podemos representar um circuito elétrico pela figura abaixo:

### Circuito Elétrico com corrente alternada



**Ilustração 10 - Circuito Elétrico de Corrente Alternada**

Temos um circuito constituído por um resistor R ligado a um gerador de corrente alternada. Temos a representação da corrente no gráfico abaixo.



**Ilustração 11 - Gráfico de Corrente Alternada**

Temos a Potência Ativa e a Reativa:

Potência Ativa: Realiza trabalho, gerando calor ou movimento. [(Paiva, 2004)].

$$P = V I \cos \phi$$

**Ilustração 12 - Potência Ativa**

Potência Reativa: É a potência reativa indutiva que mantém o campo eletromagnético, que ocorre em motores, ou cargas indutivas. A potência reativa capacitiva ocorre em cargas cujas características têm componente capacitiva. Ocorre geralmente quando se usa banco de capacitores não automáticos para correção do fator de potência [(Paiva, 2004)].

$$Q = V I \text{ sen } \phi$$

### **Ilustração 13 - Potência Reativa**

Para a potência ativa reserva-se o Watt (W) e seus múltiplos (kW, MW, GW, TW); para a reativa usa-se o Volt. Ampere reativo (VAr) e seus múltiplos (kVAr, MVAr, GVAr, TVAr).

A unidade do *watt* recebeu o nome de James Watt pelas suas contribuições para o desenvolvimento do motor a vapor, e foi adotada pelo segundo congresso da associação britânica para o avanço da ciência em 1889.

# Capítulo 3

## Metodologia

### 3.1 – Tipo de Pesquisa

O objetivo desse trabalho é fazer medições de energia de um determinado departamento utilizando um *software* livre (MRTG).

O trabalho consiste em fazer configurações no *software* para que ele colete informações de medidores de energia e gere gráficos de como está sendo o consumo. Esses gráficos serão disponibilizados a partir de um servidor *web* que proporcionará que os dados sejam vistos a partir de qualquer computador com acesso a internet.

### 3.2 - Quanto aos procedimentos técnicos

Em relação aos procedimentos técnicos a pesquisa em si é uma pesquisa-ação onde terá o envolvimento de professores do Centro de Informática (CIN-UFLA), Departamento de Exatas (DEX), e Departamento de Ciência da Computação (DCC) da Universidade Federal de Lavras trabalhando de modo cooperativo para a resolução do projeto em questão.

### 3.3 - Implementação e Construção

O trabalho em si pode ser dividido em três partes: montagem de um conversor analógico digital para aquisição dos dados, implementação de um programa na linguagem C++ para a leitura dos dados convertidos e configuração do *software* MRTG para geração de gráficos.

Todo o trabalho foi realizado em ambiente UNIX, utilizando-se a distribuição Slackware 10 do sistema operacional Linux.

Foi utilizado para a aquisição dos dados um aparelho da empresa Yokogawa modelo 2354-10 (multi-função) que fará leitura da potência elétrica e disponibilizara os dados obtidos na forma analógica. Esses mesmos dados serão lidos pelo conversor analógico digital (ADC0804) que irá converter uma dada tensão de saída em sinais digitais.

De posse desses sinais digitais vamos enviar esses dados para o computador através da porta paralela. Um programa em C++ foi desenvolvido para ler os dados recebidos na porta paralela e serem disponibilizados para que o *software* MRTG monte gráficos de cinco em cinco minutos de como está sendo o consumo elétrico do local escolhido.

# Capítulo 4

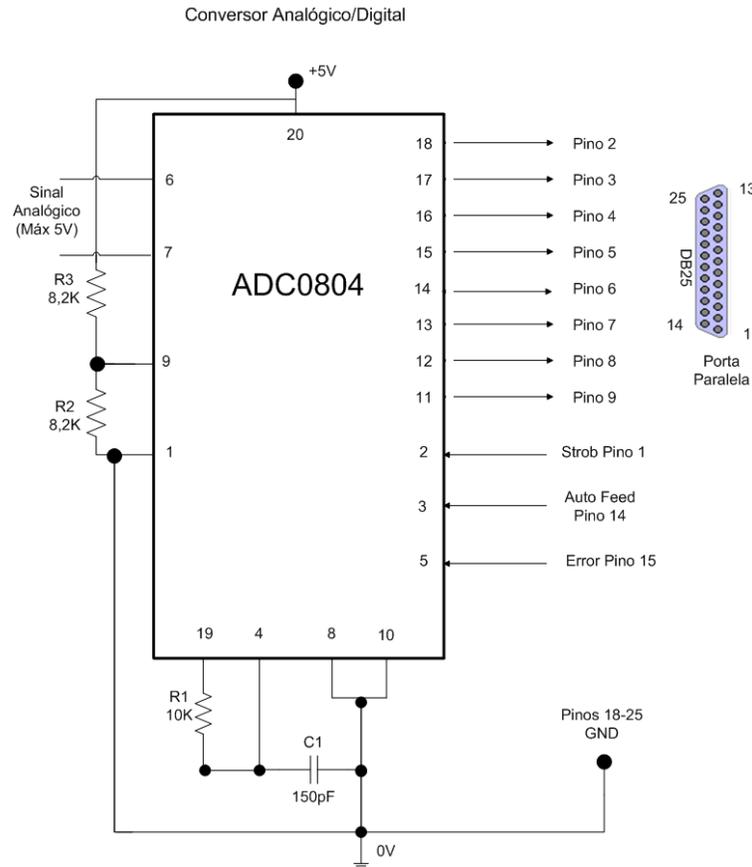
## Resultados e Discussões

Como foi dito na metodologia o trabalho foi dividido em três partes. Vamos começar aqui a mostrar todos os procedimentos desenvolvidos para a conclusão do nosso objetivo final que é fazer a leitura do consumo elétrico.

Nas próximas seções será demonstrado como foi o processo de construção, configuração, e adaptação de todos os componentes usados do início até o fim. E vamos discutir quais foram as dificuldades encontradas, as limitações impostas e os resultados obtidos na fase final de todo o processo.

### 4.1 – Conversor

Vamos descrever o conversor analógico que teve o papel de converter um sinal analógico para sua forma digital. O CI (circuito integrado) utilizado para esse propósito foi o modelo ADC0804 de oito *bits*. Esse conversor tem a função de receber uma tensão de entrada na faixa de zero a cinco *volts* e converter esse valor correspondente em oito *bits*. Nessa parte já encontramos a nossa primeira limitação, pois por ser um componente fabricado para trabalhar com baixa voltagem temos que ter cuidado para que o valor máximo de entrada seja de cinco *volts*. Um valor acima desse estipulado pelo fabricante pode causar danos ao nosso CI. Sendo assim devemos colocar um resistor para que a tensão de entrada não ultrapasse os cinco *volts*. Em conjunto com o nosso CI foi confeccionado um cabo paralelo com o conector DB25 (macho) para que os valores convertidos sejam repassados para a porta paralela do PC. Dos 25 pinos da porta paralela só foi necessário à utilização de onze pinos somente. Nos quais oito pinos são responsáveis para enviar os oito *bits* convertidos, um pino foi utilizado como terra e os outros dois restantes foram utilizados para efeito de controle do CI. Segue abaixo um diagrama completo do CI.



### 4.1.1 – Processo de aquisição de dados

Vamos descrever aqui o processo de conversão de um sinal analógico para a forma digital.

Antes vamos falar novamente sobre a resolução de um conversor AD. Como o conversor trabalha com oito *bits* temos uma resolução dada pela seguinte fórmula:

$$Resolução = \frac{5V}{256} \text{ ou } 19,5mV$$

Essa resolução me dá uma taxa de erro, ou seja, podemos fazer leituras de tensão nessa faixa de 0,0195V. Se for necessária uma maior precisão poderá ser utilizado um conversor com 12 *bits* ou mais, pois assim temos uma resolução bem menor que resultara numa precisão bem mais real ao valor analógico lido.

Sendo assim agora ao passarmos uma voltagem de entrada na faixa estabelecida esse sinal será disponibilizado em forma de sinais digitais. Esses sinais serão levados oito *bits* por vez através do cabo para a porta paralela.

O processo de conversão e aquisição dos dados será totalmente controlado por um *software* desenvolvido em C++ que fará o intermédio entre o conversor e o programa de geração de gráficos MRTG. Esses dois *softwares* serão detalhados mais adiante.

Nessa parte apenas vamos explicar como se executa um ciclo de conversão.

Para se trabalhar com o conversor basicamente se “chaveia” dois pinos. Esses pinos são os pinos dois (READ) e três (WRITE) do CI. Esse chaveamento consiste em enviar sinais lógicos zero (desligado) e um (ligado). O ciclo de conversão de um dado se inicia com o envio de um sinal zero lógico no pino três (WRITE) e depois o envio de um sinal zero lógico no pino dois (READ). Esse sinal enviado no pino dois disponibiliza os oito *bits* convertidos para a saída do conversor que então transporta esses dados pelo cabo até a porta paralela.

## 4.2 – Implementação

Trabalhando paralelamente junto com o conversor foi desenvolvido um programa na linguagem C++ que terá a função de fazer o controle do conversor e a aquisição dos dados necessários para nosso trabalho.

### 4.2.1 – Linux e a porta paralela

A diferença entre o acesso no Linux e Windows da porta paralela se diz respeito à parte de permissões. No Linux para se ter acesso a porta paralela é necessária pedir autorização para o *kernel* antes de realizar qualquer manipulação, seja ela de envio ou recebimentos de dados. Algumas vantagens dessa segurança em relação ao *Windows XP, NT e 2000* são a não necessidade de criarmos *drivers* e *dlls* que precisam de dezenas de código fonte. Podemos também no Linux, a qualquer momento, disponibilizar a porta para outro aplicativo que solicitar.

Esse pedido é realizado pela função **ioperm**. Essa função é definida na biblioteca **<unistd.h>**. Nessa função é necessário passar três parâmetros. Endereço inicial de

permissão de acesso, endereço final de permissão de acesso, e o *bit* setado em zero ou um que ativa e desativa a permissão da porta.

Esse endereço pode ser na forma hexadecimal. No caso da porta paralela o endereço padrão é o 0x378.

Sendo assim antes de iniciar qualquer requisição pedimos permissão para o sistema operacional usando a função **ioperm** que me retornara um se a permissão for aceita e zero se não for aceita.

Feito isso podemos começar a fazer o envio e recebimento de dados através da porta. Para isso utilizamos as funções **outb** e **inb**. Essas duas funções estão definidas na biblioteca **<sys/io.h>**.

Como foi dito na secção 2.4.3 a porta paralela possui três grupos de pinos. Os pinos de *status* (0x378+1) os de dados (0x378+0) e os de controle (0x378+2) onde cada grupo possui seus respectivos endereços. Quando utilizamos a função **outb** precisamos passar dois parâmetros. Sendo que o primeiro representara a ação que você deseja executar e o segundo valor representa em qual desses grupos de pinos você que executar essa ação. Já a função **inb** faz o papel contrário da **outb**. Ela é uma função que atua no recebimento de dados. Assim quando utilizamos ela apenas temos que especificar o endereço do grupo referente aos pinos de dados.

De posse dessas informações o programa tem o intuito de retornar os valores de entrada analógica para sua formar digital através do computador.

## 4.3 - Configuração do MRTG e do Apache

Nessa parte vamos mostrar como foi feita a sua configuração e como foi à integração entre o programa em C++ e o MRTG. Junto com o MRTG vamos também mostrar o processo de configuração do servidor *web* (Apache) que será responsável no processo de disponibilização dos gráficos na internet. Para a realização dos testes foi utilizada a versão 2.11.1 do MRTG disponível no site oficial dos desenvolvedores <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/pub/> e a versão 2.0.54 do servidor *web* Apache que pode ser encontrada no site <http://httpd.apache.org/download.cgi> .

### 4.3.1 – Configurando o Apache

Efetuada a instalação do Apache devemos começar a configurar as opções para que o servidor possa carregar as páginas HTML para serem disponibilizadas na internet.

Como na maioria dos programas na plataforma Linux devemos efetuar algumas mudanças em seu arquivo de configuração. O nome do arquivo é `http.conf` e ele contém todas as opções para configurar o apache de acordo com suas preferências. Mostraremos apenas as configurações mínimas para colocar o serviço *on-line*.

A primeira configuração importante é a portas TCP que será usada pelo servidor. Por padrão ele já vem setado para trabalhar na porta 80, se por acaso quiser trabalhar em outra porta basta mudar o valor.

A outra opção é a escolha do diretório onde o apache irá buscar as páginas HTML para mostrá-las na internet. Deve-se mudar na opção *DocumentRoot* o caminho onde as páginas do MRTG serão geradas. Foi utilizado o caminho `“/var/www/html”` que será usado pelo MRTG para gerar as páginas.

Outra opção que deve ser modificada é a opção *DirectoryIndex*. Aqui é onde o apache indica o nome da primeira página a ser chamada por ele. Por padrão o nome utilizado é `“index.html”`, mas deve-se modificar esse campo, pois o MRTG gera a página com o nome `“perf.html”`.

Bom com as configurações mínimas necessárias para que o serviço funcione vamos agora para a configuração do MRTG.

### 4.3.2 – Configurando o MRTG

O MRTG foi desenvolvido primeiramente para monitorar arquivos de rede, mas com o tempo foram descobertas outras formas de utilização de seus gráficos. Depois de configurado a peça principal do MRTG se torna os seus arquivos de configurações. Com a extensão `cfg` eles contêm todas as informações responsáveis para a geração das páginas HTML onde serão mostrados os gráficos.

Basicamente temos as opções padrões que são necessárias para uma mínima visualização da página e temos também as configurações opcionais que proporciona um maior controle e nível de detalhe da sua aplicação.

Temos as seguintes opções:

**WorkDir:** Aqui escolhemos em que parte do nosso computador será gerado os arquivos de *logs* e as páginas HTML.

**Refresh:** Essa opção seta de quanto em quanto tempo o navegador deve atualizar a página para mostrar os novos dados.

**Interval:** Tempo em que o MRTG executa o arquivo com os valores que serão gerados. Aqui temos uma restrição por parte do programa, pois o tempo mínimo de geração de gráficos é de cinco minutos. Nesse caso se precisássemos de valores em tempo real o MRTG seria inútil para a visualização de dados instantaneamente. Mais no nosso caso, um intervalo de cinco minutos para a leitura de potência elétrica é mais que suficiente para a visualização e identificação de horários de maiores consumo.

**Language:** Seleciona a linguagem para a geração das páginas. Com é um *software* aberto o MRTG possui vários colaboradores em várias partes do mundo. Assim temos várias linguagens disponíveis.

**RunAsDaemon:** Coloca o MRTG para rodar em *background*.

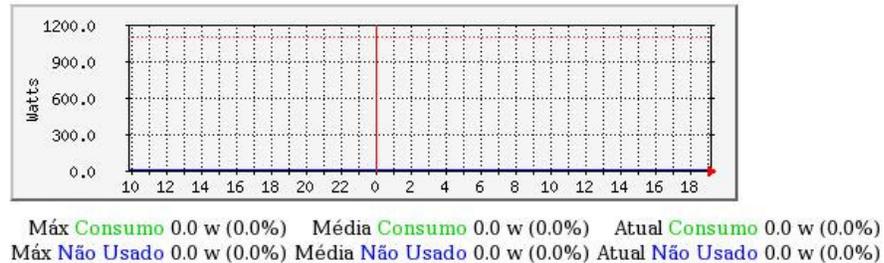
**Options[\_]:** É aqui que temos a possibilidade de escolher várias opções para melhorar a visualização das páginas. Dentro desse campo podemos colocar os seguintes valores:

**growright** - Por padrão os gráficos crescem da esquerda para a direita. Com esse comando invertemos esse parâmetro.

**noinfo** – Não coloca as informações *uptime* e nome do dispositivos na página. Como estamos utilizamos o MRTG para monitorar um equipamento que não é de rede temos que usar essa opção para que fique de acordo com o que foi proposto.

**Target[ ]:** Escolhemos o local de onde os dados serão lidos. È nessa parte que especificamos de onde o MRTG ira ler os dados. No nosso caso iremos obter as informações do nosso programa em C++ que lera os dados da porta paralela. Com o nome de “teste\_final” o arquivo de configuração do MRTG a cada cinco minutos executa esse programa que retorna os valores convertidos através da porta paralela. Como o MRTG foi construído para monitorar equipamentos de rede sempre teríamos que ter dois valores para serem mostrados. Teríamos a entrada e a saída do fluxo de dados. O nosso objetivo é fazer leitura da potencia elétrica, então apenas uma informação já é o suficiente. Assim colocamos na página do MRTG essa informação com o valor zero, pois esse dado não pode ser omitido em suas configurações. Segue abaixo como esse dado será mostrado.

### Gráfico 'Diário' (5 minutos - média)



### Ilustração 15 - Dado com valor zero

Podemos ver pela figura que aparece na cor azul a informação “Não Usada”. Em todos os gráficos na parte inferior irá aparecer essa informação que deve ser desconsiderada, pois como já foi dito esse dado não pode ser editado no arquivo de configuração do MRTG.

**Title[ ]:** Título que será mostrado pelo *browser*.

**PageTop[ ]:** Conteúdo que o usuário deseja mostrar na página. Aqui temos *tags* HTML comuns iguais em qualquer página da internet.

**MaxBytes[ ]:** Valor máximo para ser montado os gráficos

E para finalizar temos a parte de legenda onde colocamos as unidades dos dados lidos. Usamos a sigla W que corresponde aos Watts consumidos.

O MRTG possui várias outras opções que podem ser usadas para completar seu propósito. Mostramos aqui apenas aquelas informações que usamos para nosso projeto. Outros comandos podem ser encontrados no site oficial na parte *MRTG Configuration Reference* <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/mrtg-reference.html>.

Abaixo segue uma figura de um pedaço da página gerada pelo MRTG e o efeito de todas as configurações acima mostradas.

# Teste Consumo Elétrico

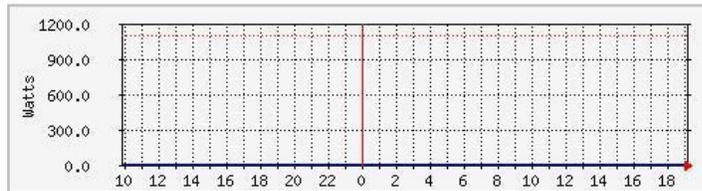
Contato: alissonrocio@gmail.com  
Descrição: TESTE PROJETO  
Tipo: Consumo Elétrico  
Consumo Máximo: 1000 W - 1Kw

---

Última atualização das estatísticas: **Sexta, 27 de Mai de 2005 às 19:14**

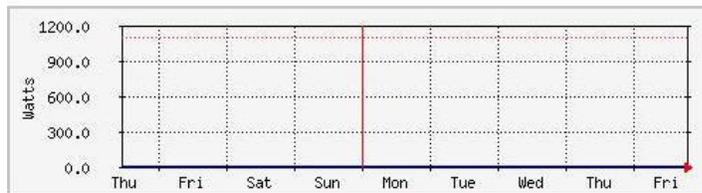
---

## Gráfico 'Diário' (5 minutos - média)



Máx **Consumo** 0.0 w (0.0%) Média **Consumo** 0.0 w (0.0%) Atual **Consumo** 0.0 w (0.0%)  
Máx **Não Usado** 0.0 w (0.0%) Média **Não Usado** 0.0 w (0.0%) Atual **Não Usado** 0.0 w (0.0%)

## Gráfico 'Semanal' (30 minutos - média)



Máx **Consumo** 0.0 w (0.0%) Média **Consumo** 0.0 w (0.0%) Atual **Consumo** 0.0 w (0.0%)  
Máx **Não Usado** 0.0 w (0.0%) Média **Não Usado** 0.0 w (0.0%) Atual **Não Usado** 0.0 w (0.0%)

### Ilustração 16 - Página do MRTG

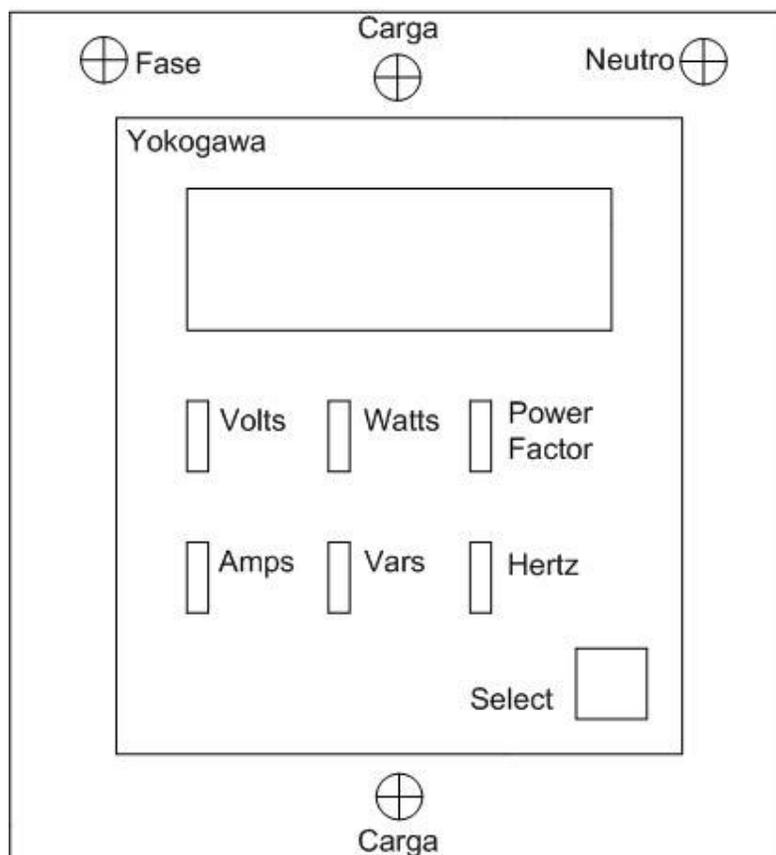
A página gerada é compreendida por quatro gráficos. Diário, Semanal, Mensal e Anual. Cada um deles fornece a informação do consumo máximo, médio e o consumo atual. Seguido de suas respectivas porcentagem. Temos também uma linha vermelha que indica nosso consumo máximo possível. Os gráficos cresceram da direita para esquerda e no eixo dos X temos uma variação na legenda dependendo de qual gráfico estamos lendo. No gráfico diário temos às horas em que foram realizadas as medidas, no gráfico semanal temos os dias da semana, no mensal temos as semanas do mês, e o anual temos os meses do ano. No eixo dos Y temos em todos os quatro gráficos os valores que está sendo lido no momento.

## 4.4 – Monitor multi-função YOKOGAWA

Veremos o aparelho responsável de fazer a leitura da potência da rede elétrica. Com ele é possível fazer a leitura das seguintes grandezas:

- VOLTS
- AMPS
- VARS
- POWER FACTOR
- HERTZ
- WATTS

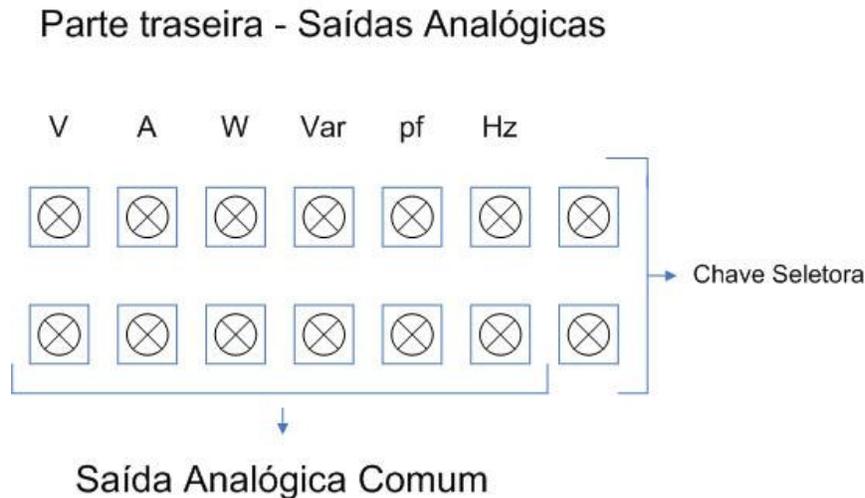
Nosso interesse é apenas na grandeza Watt que fornecerá o consumo da carga ligada ao aparelho. O aparelho possui quatro entradas. Duas serão usadas para a ligação de uma carga qualquer. As outras duas são a fase e o neutro que será usado para a ligação do aparelho na rede elétrica. Segue abaixo uma figura da parte frontal do aparelho.



**Ilustração 17 - Medidor Yokogawa Multifunção**

Nos dois pinos de cima temos a fase e o neutro. A carga se liga ao pino de baixo e no pino do meio. Podemos ver também os *leds* com as indicações das grandezas que podem ser mudadas pressionando o botão *select* na parte inferior direita.

Na parte traseira temos dois blocos de pinos. Os pinos superiores são usados pelo aparelho e nos inferiores temos as saídas analógicas.



**Ilustração 18 - Painel Traseiro**

Podemos ver na figura acima que cada pino do painel traseiro corresponde a uma grandeza medida. Para alcançar o objetivo proposto é apenas necessário ter o valor da potência elétrica. Para isso utilizaremos apenas os pinos que fornecem esse dado (Watts).

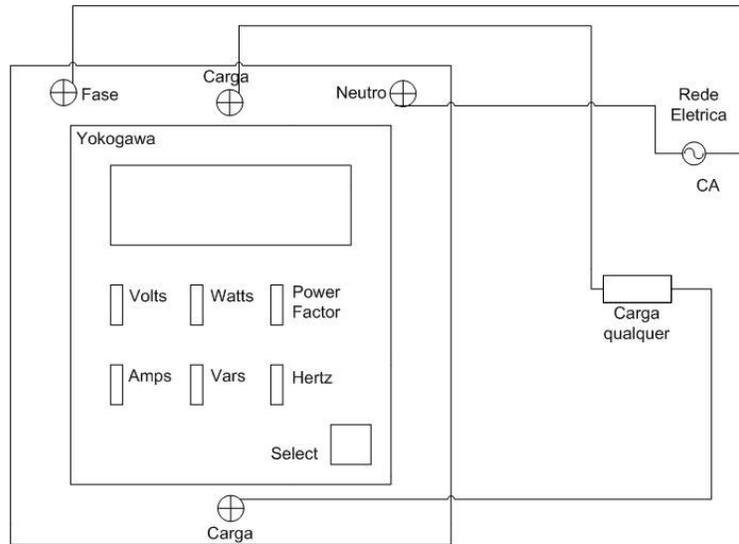
## 4.5 – Integração dos componentes

Já foram discutidos nas seções anteriores todos os componentes que são necessários para a realização do trabalho. Será visto agora como é a interação entre essas partes e como foi possível chegar ao resultado final.

### 4.5.1 Preparação

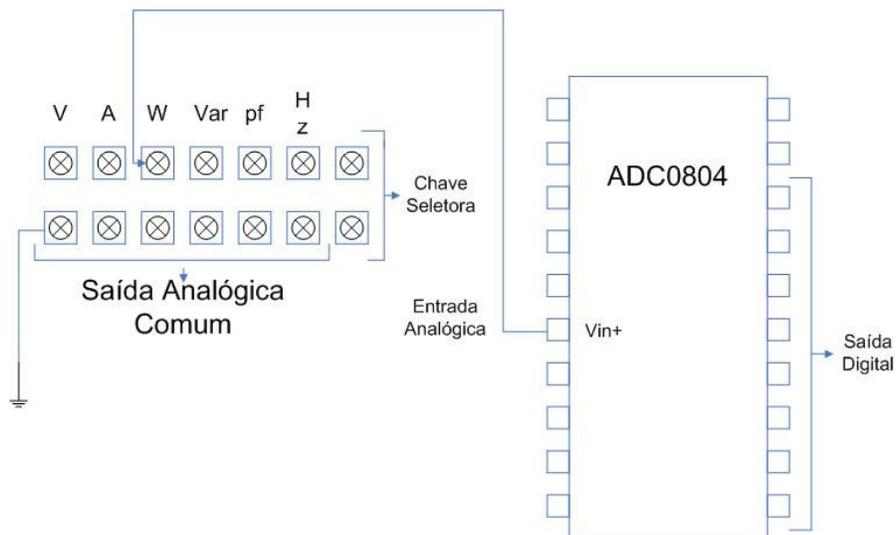
Para início do processo devemos fazer a ligação do medidor (yokogawa) à rede elétrica, pois ele será responsável em fazer a leitura do consumo elétrico de nossa carga. Essa ligação se estabelece normalmente devendo o aparelho ser ligado à rede elétrica de 127 *volts*. Uma tomada comum foi utilizada para esse propósito. Deve-se observar antes de

efetuar a ligação à identificação do neutro e fase da tomada. Depois de ligado deve-se fornecer uma carga para que o aparelho possa medir o consumo elétrico. Qualquer aparelho elétrico pode ser utilizado como carga (televisão, ventilador, computador), ou seja, qualquer aparelho que tenha certo consumo. Para ilustrar essa parte do processo podemos ver na figura abaixo essa ligação.



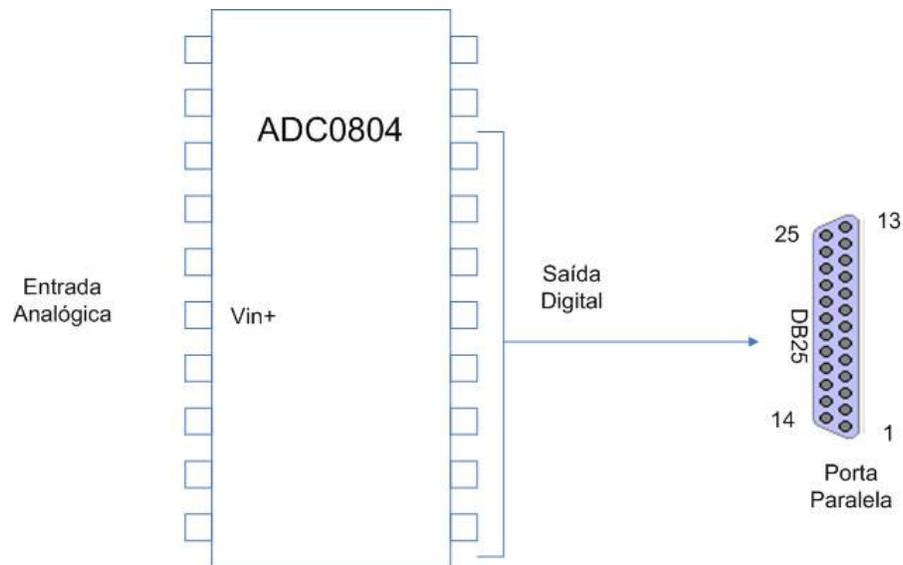
**Ilustração 19** Ligação medidor à rede elétrica

Em seguida após a preparação do medidor temos que fazer a ligação do conversor A/D ao medidor (Yokogawa). É nessa parte que se iniciara o processo de conversão dos dados lidos analogicamente para o digital.



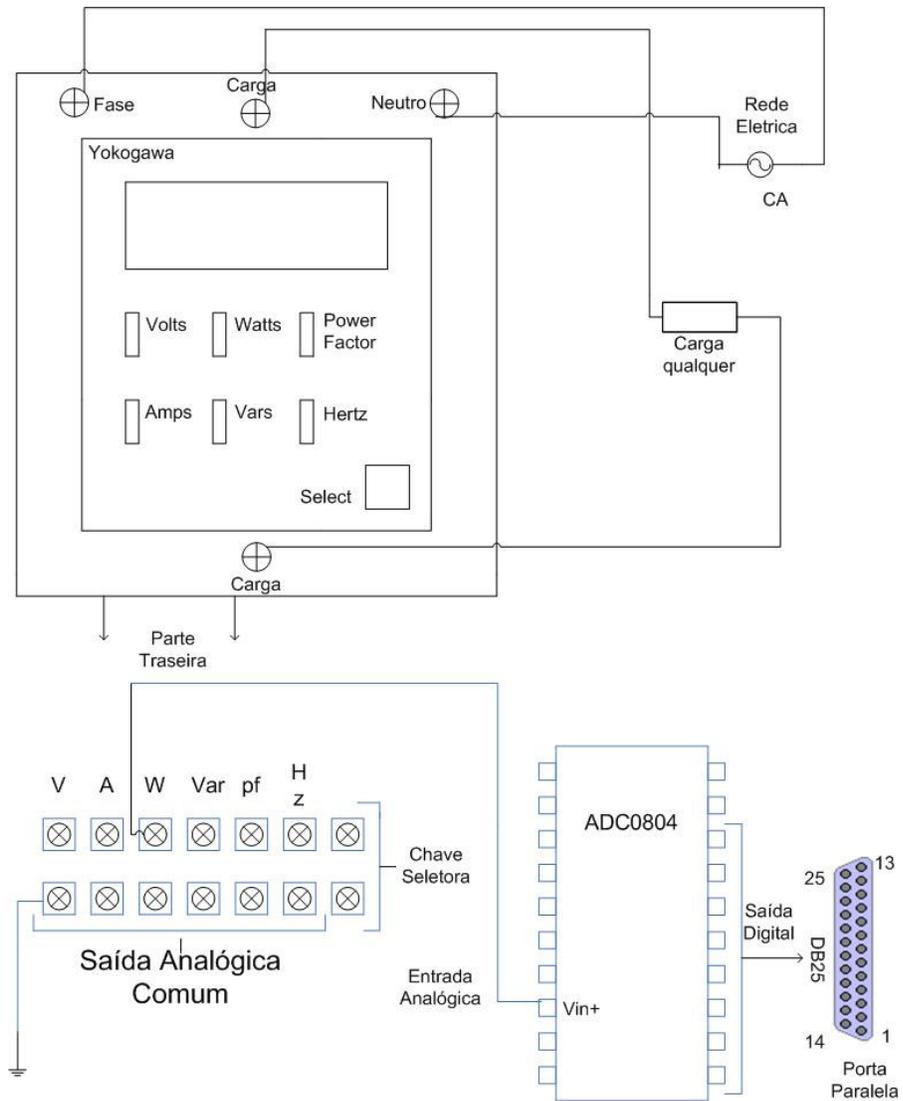
**Ilustração 20** - Ligação medidor ao conversor

Temos na figura acima a ligação da parte traseira do medidor ao conversor A/D. Temos uma voltagem de saída do medidor (Yokogawa) que servirá de entrada para o conversor. Após o processo de conversão o sinal digital estará disponível em oito *bits* nos pinos onze ao dezoito. Esse sinal digital será levado para o computador através de um cabo. Para efetuar essa ligação foi usado um cabo com o conector DB25.



**Ilustração 21 - Ligação conversor à porta paralela**

Na figura abaixo temos um diagrama completo de todos os componentes envolvidos no processo. Desde o processo de leitura dos dados (potência elétrica) até a aquisição desses dados pelo computador.



**Ilustração 22 - Ligação completa**

## 4.6 - Resultados

A proposta aqui é fazer medição de energia elétrica dos departamentos da Universidade Federal Lavras – UFLA. Considerando que cada departamento possui suas características próprias com diferença em seus equipamentos; boa parte do consumo está nas lâmpadas e nos outros equipamentos ligados a rede (computadores, equipamentos de laboratórios, etc). Para teste foi medido o consumo de uma carga ligada ao medidor Yokogawa.

Iremos fazer uma comparação entre o valor mostrado pelo fabricante da carga, o valor mostrado no display do medidor yokogawa e o valor lido pelo arquivo de configuração do MRTG que será usado para a montagem dos gráficos. Considerando as perdas provenientes de todo o processo de aquisição de dados será mostrado de quando seria essa perda e de como poderíamos melhorar esses resultados.

Foi verificada a voltagem na saída analógica com o multímetro e também o valor convertido pelo conversor A/D. Foram realizados vários testes com várias cargas diferentes para estabelecer uma relação entre a voltagem de saída e o valor mostrado no *display*. Abaixo temos uma tabela com os valores testados e seus respectivos valores.

**Tabela 4 - Valores testados I**

Valor	Lâmpada 100 w - 127 v	Lâmpada 150 w - 220 V	Ventilador
Fabricante	100 w	150 w - 220 V	45 w
Yokogawa (display)	93 w	61 w	44 w
Multímetro	2.0 v	1.4 v	1.0 v
Conversor	1.9 v a 2.1 v	1.2 v a 1.3 v	0.87 a 0.90
Valor convertido / Voltagem	44.2 a 48.9	46.9 a 50.8	48.8 a 50.5

Temos na tabela acima os valores que os fabricantes fornecem aos consumidores em relação ao consumo de seus produtos. O valor mostrador pelo *display* digital do medidor Yokogawa. O valor lido pelo multímetro. O valor na saída do conversor. E no fim a relação entre o valor convertido e o valor mostrado pelo display.

**Tabela 5 - Valores testados II**

Valor	Lâmpada + Ventilador	Lâmpada + Ventilador	Ventilador + Lâmpada + TV
Fabricante	100 w + 45 w	75 w + 45 w	45 w + 100 w + 72 w
Yokogawa (display)	136 w	106 w	205 w
Multímetro	2.8 v	2.2 v	3.8 v
Conversor	2.7 v a 2.8 v	2.1 v a 2.2 v	4.0 v a 4.1 v
Valor convertido / Voltagem	48.5 a 50.3	48.1 a 50.4	50 a 51.25

Pelos testes realizados com as seguintes cargas se pode observar que essa relação fica na casa de 50. Pode-se notar uma pequena variação entre o valor lido pelo multímetro e o valor retornado pelo conversor A/D. Isso se deve ao fato de se ter usado um multímetro analógico para chegar às voltagens de saída. Com o conversor A/D se tem uma maior precisão na leitura e por ser bastante sensível pode-se notar uma pequena variação nos valores lidos. Essas variações são poucas ficando na faixa de 0.1 v a 0.2 v.

Considerando as perdas no processo de leitura podemos verificar uma pequena taxa de erro em relação ao valor mostrado pelo display e o valor retornado pelo conversor A/D. Essa taxa de erro fica na faixa de 5% podendo ser melhorada utilizando um conversor com um número maior de *bits*.

Com a relação estabelecida entre o valor retornado pelo conversor e o valor mostrado pelo *display* do medidor Yokogawa, no programa desenvolvido deve-se multiplicar a voltagem convertida na saída analógica pelo fator de 50 para que o valor de saída do programa já seja a potência consumida pela carga.

Assim o MRTG já recebera os dados na forma de potência elétrica bastando apenas gerar os gráficos de consumo no período de cinco a cinco minutos.

Com tudo estabelecido iremos partir para a parte final do nosso objetivo que é a geração dos gráficos pelo MRTG.

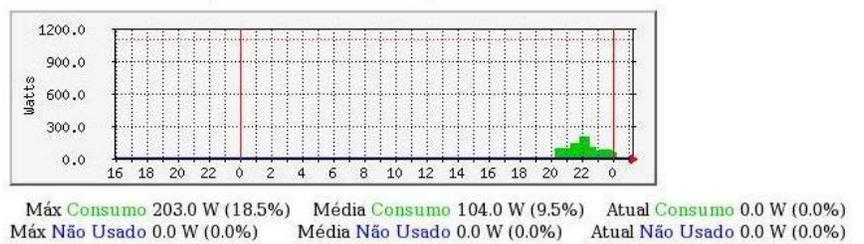
Com o arquivo de configuração do MRTG devidamente configurado (exemplo3.cfg) e a modificação no programa de leitura (teste\_final). Deve-se executar o comando para o início da geração dos gráficos. No terminal do linux deve-se executar o comando:

```
mrtg exemplo3.cfg
```

Para realizar esses testes foram utilizados vários valores de cargas. As cargas usadas foram às mesmas da tabela quatro e cinco.

O primeiro gráfico é o diário onde se tem às horas em que a medição foi realizada.

**Gráfico 'Diário' (5 minutos - média)**

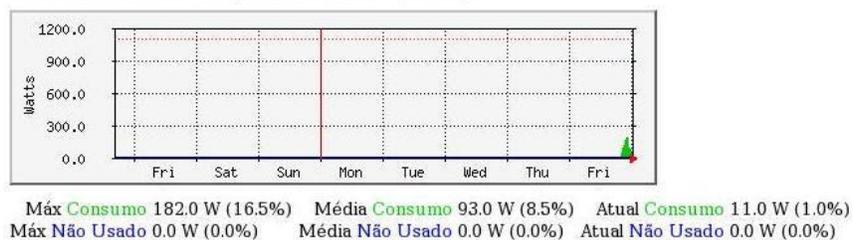


### Ilustração 23 - Gráfico diário

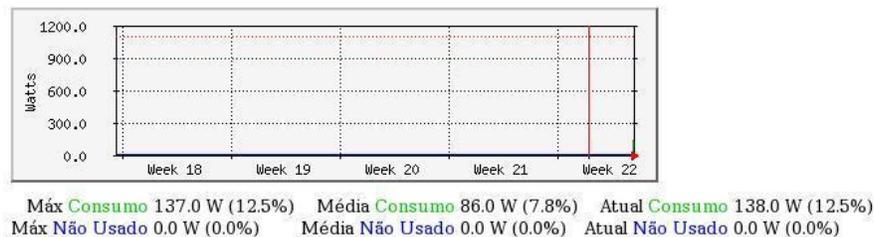
O início se deu a partir das 20h00min com a carga de 100 w. A partir das 21h00min a carga foi elevada. E as 22h00min foi elevada novamente. A partir das 22h30min esse valor foi reduzido gradativamente até o fim da medição as 00h00min.

Seguido do gráfico diário temos o semanal, mensal e o anual.

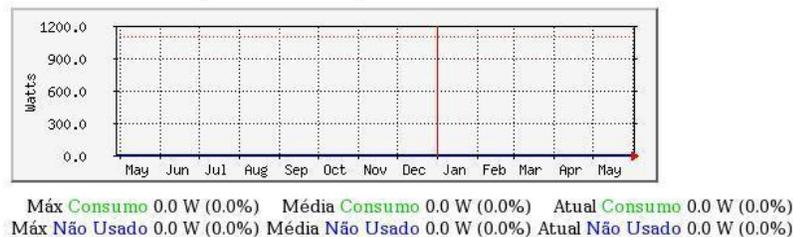
**Gráfico `Semanal' (30 minutos - média)**



**Gráfico `Mensal' (2 horas - média)**



**Gráfico `Anual' (1 dia - média)**



VERDE ### Watts  
AZUL ### Não Usado

### Ilustração 24 - Gráfico semanal mensal e anual

Podemos ver que o MRTG cumpre bem o seu papel em gerar os gráficos possibilitando uma boa visualização do comportamento do consumo elétrico.

## Capítulo 5

### Conclusão

A proposta inicial de análise de consumo elétrico foi alcançada com êxito. Os resultados apresentados nos mostra que a implementação do conversor A/D para aquisição de sinais analógicos se mostra uma opção barata e eficiente, e a geração de gráficos pelo MRTG nos traz uma boa opção para a visualização e controle de consumo elétrico. Temos também a vantagem de poder ter o acesso a essas páginas com os gráficos de consumo de qualquer lugar que tenha acesso à internet.

Considerando o fato de esse projeto ter sido desenvolvido para a análise de consumo da Universidade Federal de Lavras ele se mostra bastante eficiente para seu propósito. Tendo em vista que a UFLA possui vários departamentos com suas próprias características, os arquivos de configurações devem ser personalizados para que possam se adaptar ao local de medição.

Com esse projeto em execução pode-se identificar os locais que possuem um maior consumo e adotar medidas que possam trazer uma maior economia para a universidade.

# Capítulo 6

## Propostas Futuras

Como proposta futura tem-se as seguintes sugestões. Para uma análise com mais precisão pode-se utilizar um conversor com um número maior de *bits*, para que a conversão dos dados lidos analogicamente seja mais precisa.

Fazer a montagem do conversor em uma placa de circuito integrado para facilitar o manuseio do aparelho visto que aqui o conversor foi montado em um *protolabor*.

Utilizar um medidor para aquisição de dados com um valor maior de potência já que o aparelho utilizado nesse trabalho apresenta uma limitação de leitura de até 1000 w.

E para finalizar pode-se desenvolver um trabalho onde todo o processo de aquisição de dados seja feito pelo próprio converso A/D eliminando o uso do medidor para aquisição dos dados da rede elétrica. Nesse caso seria preciso saber duas grandezas: Voltagem e Corrente. De posse desses dados pode-se descobrir o valor da potência elétrica que será usada para a geração dos gráficos.

# Referências Bibliográficas

[Tanenbaum, (1997)], Andrew S. “Redes de Computadores”. Tradução da 3 edição. Editora Campus. 1997.

[Kretchmar, (2003)], James. “Open Source Network Administration”. Publicado por Prentice Hall PTR.2003.

[MRTG] “MULTI ROUTER TRAFFIC GRAPHER” [online] Available from <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>. Cited 28 November 2004.

[(Laurie, B. (1999) - Laurie, P. (1999)], Ben, Peter. “Apache: The Definitive Guide”. Second Edition, February 1999.

[(Engelschall, 2001)], Ralf S. “Apache Desktop Reference”. Pearson Education; 1st edition (December 15, 2000)

[APACHE], “THE APACHE SOFTWARE FOUNDATION” [online], Available from <http://www.apache.org/> , Cited 25 November 2004.

[(Paiva, 2004)], José Pedro Sucena, “Fundamentos dos Sistemas de Energia Elétrica”, Coleção Ensino da Ciência e da Tecnologia - n.º 6.

[(Serway, 1996)], Raymond A, “Física três para Cientistas e Engenheiros com Física Moderna”, Livros Técnicos e Científicos Editora S. A.

[(Axelson, 1997)], Jan. “Parallel Port Complete”, Programming, Interfacing, & Using the PC's Parallel Printer Port, Lakeview Research; Bk&Disk edition (February 1, 1997).

[ADC0804], “National Semiconductor” [online] Available from [www.national.com](http://www.national.com). Cited 10 April 2005.

[A/D], “Conversores D/A e A/D” [online] Notas de aula COM 145 Available from <http://www.comp.ufla.br/~giacomini/>. Cited 6 April 2005.

## Resumo Estendido

O objetivo desse trabalho consiste em fazer a adaptação do *software* livre MRTG para fazer a geração de gráficos de como está sendo o consumo de energia elétrica dos departamentos da Universidade Federal de Lavras. O MRTG consiste em um *software* livre construído inicialmente para monitorar o tráfego de equipamentos de redes, que foi utilizado nesse trabalho para gerar gráficos que possam ser acessados de qualquer local que tenha acesso a internet.

Primeiramente para a aquisição dos dados da rede elétrica foi usado um medidor da empresa Yokogawa que tem a função de ler a potência elétrica da rede, e disponibilizar esses dados na forma analógica. O medidor multifunção Yokogawa disponibiliza várias outras informações em relação à rede elétrica mais para nosso propósito apenas um dado é necessário. Esse dado é a potência elétrica que indicará como está sendo o consumo das cargas ligadas ao medidor. De posse dessas informações o aparelho disponibilizará em sua parte traseira esse valor lido na forma analógica.

De posse desses dados analógicos foi construído um conversor A/D (Analógico/Digital) e um programa na linguagem C++ para fazer o controle do conversor. Esse conversor tem a função de transformar aquele sinal analógico em sinais digitais, que serão usados pelo MRTG para a geração de gráficos do consumo elétrico do local. Esse conversor utiliza-se um circuito integrado modelo ADC0804 oito bits, que tem como função transformar uma voltagem analógica inicial em seu correspondente digital. E através da porta paralela iremos jogar esses dados para o MRTG para que inicie a geração dos gráficos.