

BRUNO CAMPANA ARAGON

**GERENCIAMENTO REMOTO A SERVIDORES DE REDES LOCAIS:
GERENCIAMENTO VIA WEB x GERENCIAMENTO VIA TERMINAL**

Monografia de projeto orientado 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. Luiz Henrique Andrade Correia

LAVRAS
MINAS GERAIS – BRASIL
2001

BRUNO CAMPANA ARAGON

**GERENCIAMENTO REMOTO A SERVIDORES DE REDES LOCAIS:
GERENCIAMENTO VIA WEB x GERENCIAMENTO VIA TERMINAL**

Monografia de projeto orientado 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 29 de Junho de 2001.

Prof. Jones Oliveira de Albuquerque

Prof. Anderson Bernardo dos Santos

Prof. Luiz Henrique Andrade Correia
UFLA
(Orientador)

LAVRAS
MINAS GERAIS – BRASIL

DEDICATÓRIA

Dedico primeiramente a Deus pela sorte de sempre ter encontrado pessoas boas que de uma forma ou de outra ajudaram-me nesta conquista.

Dedico aos meus pais, Jaime e Anna, meus familiares, a minha companheira Daniella, e a todos os meus amigos e colegas que tornaram essa jornada possível e, as vezes, divertida.

“A vida só pode ser compreendida olhando para trás, mas deve ser vivida olhando para frente.”

(Kierbegaard)

AGRADECIMENTOS

Agradeço aos meus pais pela oportunidade e confiança, aos meus companheiros de turma pelos momentos de alegria ao longo desses anos e pelas batalhas vencidas, a minha companheira pelo compartilhamento de todos os momentos de alegria e tristeza e pela eterna atenção, e a seus familiares.

Agradeço também aos meus mestres pelos ensinamentos passados. Em especial ao meu orientador Luiz Henrique responsável direto pelo meu ingresso na área de redes de computadores.

Agradeço aos meus colegas de trabalho no centro de informática pelos conhecimentos adquiridos através deles.

E por fim a todos os meus amigos que ao longo deste, sempre estiveram em minha convivência.

SUMÁRIO

LISTA DE FIGURAS	iii
LISTA DE TABELAS	iv
RESUMO	v
1. INTRODUÇÃO	1
2. REVISÃO LITERATURA	3
2.1. Redes Locais	3
2.2. Análise Heurística de Usabilidade	3
2.3. Gerenciamento Remoto	7
3. PROPOSIÇÃO	11
4. MATERIAIS E MÉTODOS	13
4.1. A Plataforma Webmin	13
4.2. O Protocolo SSL	15
4.2.1. Por que utilizar transações seguras ?	17
4.3. Compilação de Mensagens – MD5	17
4.4. O Protocolo SSH	19
4.4.1. Por que SSH ?	19
4.5. Serviços Implementados	20
4.5.1. Domain Name System	20
4.5.2. O Protocolo SMB – Service Message Block	22
4.5.3. Servidor WWW – Apache	23
4.5.4. File Transfer Protocol – FTP	25
4.5.5. Majordomo	26
4.5.6. Metodologia	26
5. RESULTADOS E DISCUSSÕES	28
5.1. Domain Name System	31
5.2. O Protocolo SMB – Service Message Block	35
5.3. Servidor WWW – Apache	37
5.4. File Transfer Protocol – FTP	38
5.5. Majordomo	39
5.6. Discussões	40
6. CONCLUSÕES	43
6.1. Limitações do Estudo	44

6.2.	Sugestões para Novos Estudos	45
7.	REFERÊNCIAS BIBLIOGRÁFICAS	47
	APÊNDICES	49

LISTA DE FIGURAS

Figura 1 – Exemplo do espaço de nomes de domínios da <i>Internet</i>	21
Figura 2 – Tela de serviços do Webmin	28
Figura 3 – Tela de gerenciamento do Majordomo	29
Figura 4 – Tela inicial do serviço DNS	32
Figura 5 – Tela para criação de uma zona para o serviço de DNS	33
Figura 6 – Tela de gerenciamento de uma zona específica	34
Figura 7 – Tela inicial de gerenciamento do serviço SMB	36
Figura 8 – Tela inicial de gerenciamento do serviço WWW	38
Figura 9 – Tela de gerenciamento do serviço FTP	39
Figura 10 – Tela de criação de uma lista para o serviço Majordomo	40

LISTA DE TABELAS

Tabela 1 – Sistemas Operacionais suportados pelo Webmin

14

RESUMO

O aumento na utilização de redes de computadores tem exigido cada vez mais dos Administradores de sistemas mecanismos eficientes para gerenciar os serviços oferecidos por seus servidores. Nesta pesquisa são discutidas duas formas de fazer gerenciamento: a primeira via terminal e a segunda via interface gráfica, e demonstra-se que a segunda apresenta melhores resultados simplificando tal tarefa.

ABSTRACT

The increase in the use of computer networks has demanded more of the Systems administrators efficient mechanisms to manage the services offered by their servers. In this research has been discussed two ways to make management: the first is by terminal and the second is by graphical interface, and has been showed better results in the second feature simplifying this task.

1. INTRODUÇÃO

Alguns anos atrás, a Universidade Federal de Lavras – UFLA – iniciou a construção de sua rede local de computadores – *Local Area Network* – LAN. O primeiro prédio a possuir sua rede montada foi o do Departamento de Ciências Exatas – DEX. Em seguida, os outros departamentos, aos poucos, tiveram suas redes montadas. Logo, deu-se início à interligação desses departamentos por meio de fibra óptica ao prédio do Departamento de Administração da UFLA, com o propósito de interligar a LAN à rede mundial de computadores, *Internet*. Os serviços essenciais para tal era realizado pelo servidor central que encontrava-se no prédio do Departamento de Administração da UFLA.

A medida que a obra foi se consolidando, notou-se, pelas dimensões do projeto, que um servidor central não comportaria no futuro a demanda dos serviços por ele prestado; serviços como DNS, FTP, WWW, E-mail, entre outros que serão apresentados nas seções específicas. A solução encontrada foi colocar em cada departamento um computador no qual seria um *subservidor*, que realizaria os serviços descritos acima, ao invés de ser realizado pelo servidor central. Essa solução nos permite fazer a seguinte analogia: seria como uma *terceirização* dos serviços oferecidos pelo servidor central, sendo agora realizado pelos subservidores.

Uma vez realizada a tarefa de migração desses serviços do servidor central para os subservidores, a verificação da funcionalidade desses serviços, ou seja, o *gerenciamento*, dar-se-á de forma centralizada, uma vez que os administradores de sistemas da UFLA realizam suas tarefas no Centro de Informática da UFLA – CIN-UFLA, configurando-se então em um gerenciamento remoto dos subservidores que atualmente está em um número de 13 máquinas.

Na área de redes de computadores e sistemas distribuídos o tema de gerenciamento tem sido de grande destaque no Brasil. Nos últimos anais da conferência mais importante na área de redes, o Simpósio Brasileiro de Redes de Computadores, verifica-se que já existem seções específicas da área de gerenciamento com artigos completos sobre o assunto. Isso é muito considerado, visto que a área de redes é muito ampla e compreende várias sub-áreas[JMARCOS].

Neste trabalho é realizada uma análise comparativa sobre duas formas de fazer gerenciamento remoto para com os serviços que serão posteriormente descritos, uma através de uma interface gráfica – via Web – utilizando-se um software gratuito e poderoso – o *Webmin* – e outra sem interface gráfica – via terminal – utilizando o SSH. O primeiro utiliza-se da segurança do *Secure Socket Layer* – SSL, e o segundo, a segurança do *Secure Shell* – SSH.

Este trabalho está organizado como se segue. Na seção 2 apresenta-se alguns conhecimentos básicos sobre o assunto, a Revisão de Literatura. Na seção 3 apresenta-se a hipótese e o objetivo do trabalho, a Proposição. Na seção 4 apresenta-se os materiais e a metodologia adotada para interpretação dos resultados. Na seção 5 apresenta-se os resultados e discussões. E na seção 6 apresenta-se a conclusão.

2. REVISÃO DE LITERATURA

2.1. Redes Locais

As redes locais, também chamadas de LANs, segundo [TAN], são redes privadas contidas em um prédio ou em um campus universitário que possuem alguns quilômetros de extensão. Elas são largamente utilizadas para conectar computadores e estações de trabalho em escritórios, laboratórios, etc, permitindo o compartilhamento de recursos, impressoras por exemplo, e troca de informações. Para [TAN], existem três características que diferenciam as redes locais das demais: tamanho, tecnologia de transmissão e topologia.

As LANs têm tamanho restrito, significando que o pior tempo de transmissão é limitado e conhecido com antecedência. O conhecimento desse limite, facilita seu projeto, além de simplificar seu gerenciamento.

A tecnologia de transmissão das LANs, consiste de cabos aos quais todas as máquinas são conectadas. As LANs tradicionais são executadas a uma velocidade que pode variar de 10 a 100 Mbps – megabits por segundo – têm um baixo retardo e cometem poucos erros. Existem LANs mais modernas que atingem velocidades mais altas, alcançando centenas de Mbps.

As LANs de difusão¹ aceitam diversas topologias. Duas mais conhecidas são: barramento e anel. A rede *ethernet* é um exemplo de rede de transmissão de barramento. E a rede *Token Ring* da IBM é uma rede local popular em formato de anel.

2.2. Análise Heurística de Usabilidade

Esta seção apresentará os conceitos de usabilidade utilizadas nas seções que se seguem para uma análise de uma ferramenta gráfica.

¹ Redes de difusão têm apenas um canal de comunicação, compartilhado por todas as máquinas. 3

A propriedade de uma interface com o usuário é que permite classificá-la quanto a sua qualidade é conhecida na literatura como *usabilidade*, conceito definido tradicionalmente como a conjunção de cinco atributos:

1. Facilidade de aprendizado: O sistema deve permitir que o usuário aprenda a executar suas tarefas no prazo mais curto possível;
2. Eficiência de uso: O sistema, uma vez dominado pelo usuário, permite um alto grau de produtividade;
3. Retenção: O sistema deve ser lembrado facilmente mesmo pelo usuário casual, de forma que o retorno ao sistema após cada período de ausência não implique um reaprendizado extensivo;
4. Minimização de erros: O sistema deve ter uma taxa baixa de erros de utilização. Além disso, os erros cometidos pelo usuário devem ser facilmente recuperáveis;
5. Satisfação: O sistema deve ser agradável de usar, ou seja, seus usuários ficam subjetivamente satisfeitos com ele.

Segundo [NIELSEN], os seguintes princípios devem ser seguidos por uma interface com o usuário para que esta venha a ser *usável*:

Diálogo Simples e Natural

A interface com o usuário deve ser tão simples quanto possível. Cada elemento ou item de informação colocado numa tela representa um item a mais para aprender, uma fonte a mais de possível confusão. A realização de tarefas com o sistema deve minimizar a navegação do usuário pelo sistema.

O *design* gráfico da interface deve por si só mostrar as relações entre os elementos do diálogo. Isto é, elementos de um mesmo grupo devem estar juntos em uma linha, ou caixa, etc.

Princípios de *design* gráfico também devem ser usados quando é necessário priorizar a atenção do usuário. Por exemplo, a informação é lida da

esquerda para a direita, de cima para baixo, portanto, os elementos do diálogo no topo da tela devem receber mais atenção.

O uso da cor nos diálogos deve ser moderado. O uso de 5 a 7 cores numa interface é suficiente para a maioria das aplicações.

Apenas informações relevantes para uma determinada tarefa do usuário deve aparecer na tela correspondente.

Fale a Língua do Usuário

Os diálogos devem ser expressos claramente em palavras, expressões e conceitos, não em termos orientados ao sistema. Por exemplo, não é conveniente exigir que os usuários especifiquem códigos do sistema para determinadas ações, mas sim, se for o caso, mascarar esses códigos para que a tarefa seja facilitada para o usuário.

Consistência

Os usuários não devem ficar em dúvida se diferentes palavras, situações ou ações realizarão a função desejada. A consistência é um dos princípios mais básicos de usabilidade. Se os usuários souberem que o mesmo comando ou mesma ação obterá sempre o mesmo efeito, eles se sentirão mais confiantes e o aprendizado do sistema ficará mais facilitado, aumentando a produtividade do usuário. Outro item a ser observado é que as telas e caixas de diálogo devem ser formatadas da mesma maneira, assim facilitando o seu reconhecimento.

Retorno

O sistema deve informar ao usuário continuamente sobre o que está sendo feito e como a entrada do usuário está sendo interpretada. O retorno não deveria esperar que um erro ocorra. O retorno torna-se especialmente importante quando o sistema tem tempos longos para algumas operações. Para atrasos entre 1 a 10 segundos, o sistema deve dar alguma indicação que um processamento está ocorrendo. É importante prover também uma forma de cancelar a operação. Outro ponto em que o retorno torna-se essencial é na ocorrência de falhas do

sistema. Deve ficar claro para o usuário que a falha foi do sistema e o que poderá se fazer a respeito.

Saídas Claramente Marcadas

Não é aconselhável para o sistema fazer o usuário sentir-se encurralado pelo sistema. Deve-se prover saídas fáceis e explícitas de tantas situações possíveis, assim como, possibilitar ao usuário voltar a um estado anterior.

Atalhos

Também conhecidos como aceleradores, estes incluem abreviações de comandos, combinações de teclas que são mapeadas em comandos do sistema, clique duplo com o mouse sobre um elemento para realizar sua ação mais comum e menus de botões. É interessante para o sistema obedecer os valores já conhecidos pelo usuário.

Boas Mensagens de Erros

Situações de erros são críticas do ponto de vista da usabilidade por duas razões. Primeiro, elas representam por definição um ponto em que o usuário obteve problemas e potencialmente será impedido de atingir seu objetivo original. Segundo, elas apresentam uma oportunidade de ensinar ao usuário, pois ele está motivado a prestar atenção na mensagem apresentada e o sistema tem geralmente o conhecimento do que causou o problema. Deve-se escrever mensagens de erros em linguagem clara e evitar o uso de códigos obscuros. Deve ser possível para o usuários entende-la por si só sem recorrer a manuais ou tabelas de código.

As mensagens de erros devem ser precisas e não vagas ou genéricas, construtivas e ajudar o usuário a resolver o problema, e sobretudo educadas, termos fortes como *fatal*, *ilegal*, etc. devem ser evitados.

Ajuda e Documentação

É preferível que um sistema seja tão fácil de usar que nenhuma ajuda adicional ou documentação seja necessária, mesmo assim, a existência de ajuda e documentação não reduz os requerimentos de usabilidade da interface em si.

A regra fundamental para a documentação é que a maioria dos usuários não as lêem. O corolário dessa regra é que quando os usuários recorrem aos manuais, eles estão em algum tipo de emergência e precisam de ajuda imediata. Tendo esse conhecimento, é essencial que o sistema de ajuda e o manual do usuário seja orientado por tarefas, liste passos concretos e use uma linguagem o mais clara e concisa possível.

2.3. Gerenciamento Remoto

[JMARCOS] relata que os sistemas computacionais estão se estruturando cada dia mais em forma de redes de elementos que se comunicam através de uma estrutura de transporte de dados. Esta estruturação tem ocorrido em todos os níveis, seja em pequenas empresas ou departamentos, seja em corporações médias ou grandes, onde muitas vezes há separação geográfica entre os setores que a constituem.

Ainda de acordo com [JMARCOS], as plataformas de gerenciamento podem possibilitar, entre outras coisas, a coleta e o envio de dados para os elementos da rede, a operação remota de elementos, o armazenamento de informações, a apresentação das informações de forma adequada, o diagnóstico e a correção de problemas. As plataformas de gerenciamento servem também como ambiente para execução de aplicações específicas.

Segundo [JMARCOS] a arquitetura de plataformas divide-se basicamente em três tipos: centralizada, hierárquica e distribuída. A *arquitetura centralizada* consiste de elementos de processamento e armazenamento

centralizados e elementos de coleta e atuação remotos, próximos aos elementos de redes. As decisões de gerenciamento são tomadas a partir de um ponto central.

A *arquitetura hierárquica* consiste de elementos de processamento centralizado, elementos de armazenamento distribuídos segundo uma hierarquia e de elementos de coleta/atuação remotos. As decisões de gerenciamento são tratadas ainda de forma centralizada.

A *arquitetura distribuída* tem todos os elementos distribuídos, sejam os elementos de processamento, as bases de dados e a coleta/atuação. As decisões de gerenciamento podem ser tomadas de forma distribuída.

A maioria das plataformas existentes ainda são centralizadas, havendo entretanto algumas hierárquicas e poucas distribuídas.

Para [JMARCOS], a maior parte das redes de computadores tende a ser local e corporativa, presta mais serviços de informação e transferência de dados e não há contabilização do uso de seus recursos, salvo em casos de provedores de *Internet*.

Segundo [SMP], a fim de garantir uma certa qualidade de serviço – *Quality of Service* – a seus usuários, as redes de computadores devem ser gerenciadas. Em essência, o gerenciamento de redes busca assegurar que sistemas de informações, disponíveis em redes, estejam operacionais a todo instante.

O gerenciamento de redes envolve o monitoramento e o controle de recursos distribuídos em redes. O monitoramento está relacionado com a observação e a análise do estado e do comportamento da configuração de redes e seus componentes, seja recursos físicos ou lógicos. Por outro lado, o controle está relacionado com a alteração da configuração de redes e de seus componentes.

Ainda de acordo com [SMP], o gerenciamento de redes de computadores, a exemplo do gerenciamento de qualquer outro sistema, tem sua complexidade dependente do porte e da heterogeneidade do sistema gerenciado. Dentre os problemas que implicam uma maior complexidade no gerenciamento de sistemas distribuídos e heterogêneos estão: a diversidade das tecnologias, recursos e serviços disponíveis; e a dificuldade de se gerenciar tais sistemas a partir de um único ponto de controle. Por isso, o gerenciamento de redes de computadores tem se tornado uma atividade cada vez mais complexa devido, principalmente à diversidade de plataformas de hardware e software que compõem as redes. A complexidade da atividade de gerenciamento requer a utilização de um sistema de gerenciamento remoto eficiente.

Para [SJC], uma ferramenta de apoio gerencial deve possuir uma interface homem-máquina flexível, fácil de utilizar e ser acessada, que permita uma boa navegação pelos dados e apresente os resultados de forma satisfatória. Uma ferramenta com as características descritas acima é de grande auxílio para as redes de computadores, uma vez que possibilita a melhoria do processo de gerenciamento, a melhoria da qualidade de serviço prestado aos usuários, etc.

Ainda para [SJC], o uso da tecnologia Web veio muito bem ao encontro da necessidade de interfaces de fácil uso remoto, vista uma vez a heterogeneidade de uma rede possibilitando sua fácil visualização.

Para [ROGERIO e MARCOS], a forma de como é realizada a interação homem-máquina assume fundamental importância quando tem-se grandes volumes de informações para tratar, principalmente quando são levadas em conta as horas em que o usuário expõe-se ao serviço podendo ocorrer falhas devido a essa exposição. E ainda definem supervisão como “sendo o processo de acompanhamento e controle sistemáticos de um conjunto predeterminado de condições operacionais em um sistema ou equipamento.”

Segundo [ROGERIO e MARCOS], as interfaces gráficas desses sistemas devem prover aos usuários facilidades para que esse possa rapidamente determinar onde está ocorrendo um problema, sua identificação e uma possível solução. Essas aplicações devem ser de grande valia tanto para a supervisão – gerenciamento – como para sua configuração do sistema.

3. PROPOSIÇÃO

Duas formas de apresentação de gerenciamento remoto a servidores de redes locais estarão sendo discutidas, em nível de comparação, para apontar-se qual seria melhor aproveitada pelos administradores de sistemas da UFLA conhecida a situação problemática descrita na introdução. Uma dessas formas é uma apresentação – via Web – que conta com uma interface gráfica, e a outra forma de apresentação seria o gerenciamento remoto sem a interface gráfica, seria uma conexão aberta via um terminal.

Ao final desse estudo será pretendido que fique compreensível para o leitor a percepção de que, pelas circunstâncias do problema, a melhor forma a ser adotada pelos administradores de sistemas para a realização do gerenciamento remoto será a apresentação via Web com uma interface gráfica auxiliando a gerência dos serviços que serão citados no próximo capítulo.

Em um gerenciamento via terminal – sem interface gráfica – o administrador de sistemas realizará suas atividades de monitoração utilizando apenas um teclado, não estará disponibilizado nenhum recurso visual para auxiliá-lo, forçando uma maior concentração e possivelmente um desgaste em tempo mais curto, uma vez que, por exemplo o profissional precisa fazer alterações em todos os serviços de DNS, o recurso de “copiar e “colar” é perigoso para o sistema, pois alguns servidores têm configurações diferentes dos demais e simplesmente fazer uma cópia, o administrador poderá não reconhecer essas diferenças em determinado instante e comprometer o funcionamento do mesmo. Para realizar o exemplo descrito é recomendável realizá-lo servidor-a-servidor e assim podendo ocorrer desgaste físico e mental no administrador, situação que poderá induzir o mesmo ao erro.

Ainda no mesmo exemplo anterior, com a utilização da Web e uma poderosa ferramenta gráfica para gerência remota, a realização dessa tarefa

fluirá de forma menos desgastante para o administrador, uma vez que o trabalho ficará resumido a preenchimento dos campos necessários, sem a preocupação de colocar todos os parâmetros e terminações – como ponto, ponto-e-vírgula, etc – que são solicitados pelo serviço em questão. Uma vez que quem realizará esta tarefa árdua será a ferramenta de gerenciamento apresentada neste estudo.

Visto esse caso e outros que aparecerão nas próximas seções, a hipótese a ser apresentada é a de que gerenciamento remoto com a utilização de interfaces gráficas é menos propício a execução de erros pelo administrador de sistemas, que por sua vez dominada a ferramenta, permitirá um alto grau de produtividade.

4. MATERIAL E MÉTODOS

Nesta seção, estará sendo apresentado os materiais utilizados para a obtenção dos posteriores resultados. Os respectivos materiais no qual auxiliaram o estudo foram: a plataforma Webmin, o protocolo SSL – Secure Socket Layer – o algoritmo MD5, o protocolo SSH – Secure Shell – e finalmente os serviços que foram implementados utilizando das duas formas de se fazer gerenciamento descrito na Seção 3: DNS , SAMBA – SMB , APACHE, FTP, e Majordomo. E abordando a metodologia utilizada para a comparação entre as interfaces.

4.1. A Plataforma Webmin

O Webmin é uma poderosa interface de administração baseada em Web para sistemas Unix/Linux. Usando Webmin o administrador pode configurar os serviços acima descritos e outros serviços utilizando o seu *Web browser*.

O Webmin consiste de um servidor Web e um número de programas CGI no qual pode-se alterar diretamente os arquivos de sistemas ou de configuração. O *Web server* e todos os programas CGI são escritos na linguagem Perl e não utilizam módulos externos significando que o sistema somente necessita de Perl binário para rodar o Webmin, garantindo assim a portabilidade dos módulos para um diferente número de sistemas Unix/Linux e tipos de CPUs. Segundo Jamie Cameron, um analista da Caldera Systems – responsável pela distribuição do software – a linguagem Perl foi utilizada por causa de sua poderosa manipulação de strings, editoração de arquivos e capacidade de expressões regulares. Tudo isso é importante quando se tenta editar os arquivos de configuração Unix/Linux. Em particular, a plataforma Webmin, utiliza dois módulos perl: o SSLeay e o MD5. Ambos serão descritos nas sub-seções 4.2 e 4.3 respectivamente.

Quanto a sua distribuição, todas as passadas e futuras versões do Webmin estão avaliadas sob a licença BSD, isto significa que no Linux e outra plataforma, o Webmin pode ser distribuído gratuitamente e modificado para uso comercial ou não comercial. Isto é possível porque o Webmin suporta a concepção de módulos, como os pluggins PhotoShop e Netscape, qualquer pessoa pode desenvolver e distribuir seus próprios módulos facilmente sem modificar os códigos já existentes para qualquer propósito, e esta distribuição pode ser sob qualquer licença, como por exemplo a GLP, comercial ou shareware.

O Webmin suporta um bom número de sistemas operacionais. Na tabela 1 encontram-se os sistemas operacionais suportados pela atual versão do Webmin.

Tabela 1 – Sistemas Operacionais Suportados pelo Webmin

Sistema Operacional	Versão Suportada
Sun Solaris	2.5 , 2.5.1 , 2.6 , 7 , 8
Caldera OpenLinux eServer	2.3
Caldera OpenLinux	2.3 , 2.4 , 2.5 , 3.1
Redhat Linux	4.0 , 4.1 , 4.2 , 5.0 , 5.1 , 5.2 , 6.0 , 6.1 , 6.2 , , 7.0 , 7.1
Slackware Linux	3.2 , 3.3 , 3.4 , 3.5 , 3.6 , 4.0 , 7.0 , 7.1
Debian Linux	1.3 , 2.0 , 2.1 , 2.2
SuSE Linux	5.1 , 5.2 , 5.3 , 6.0 , 6.1 , 6.2 , 6.3 , 6.4 , 7.0, 7.1
Corel Linux	1.0 , 1.1 , 1.2
TurboLinux	4.0 , 6.0
Cobalt Linux	2.2 , 5.0 , 6.0
Mandrake Linux	5.3 , 6.0 , 6.1 , 7.0 , 7.1 , 7.2
Delix DLD Linux	5.2 , 5.3 , 6.0
Conectiva Linux	3.0 , 4.0 , 4.1 , 4.2 , 5.0 , 5.1 , 6.0
MkLinux	DR2.1 , DR3
LinuxPPC	2000

Tabela 1 – Sistemas Operacionais Suportados pelo Webmin (Continua)

XLinux	1.0
LinuxPL	1.0
Linux From Scratch	2.2
Trustix	1.1 , 1.2
Cendio LBS Linux	3.1 , 3.2 , 3.3
Ute Linux	1.0
FreeBSD	2.1 , 2.2 , 3.0 , 3.1 , 3.2 , 3.3 , 3.4 , 3.5 , 4.0, 5.0
OpenBSD	2.5 , 2.6 , 2.7
BSDI	3.0 , 3.1 , 4.0
HP/UX	10.01 , 10.10 , 10.20 , 10.30 , 11
SGI Irix	6.0 , 6.1 , 6.2 , 6.3 , 6.4 , 6.5
DEC/Compaq OSF/1	4.0 , 5.1
IBM AIX	4.3
SCO UnixWare	7 , 2
SCO OpenServer	5
MacOS Server X	1.0 , 1.2 , 1.3

4.2. O Protocolo SSL

Originalmente desenvolvido pela Netscape, o SSL tem sido universalmente aceito na Web para autenticar e encriptar comunicações entre clientes e servidores. Ele usa uma forma de chave pública, onde a informação pode ser codificado pelo browser usando uma válida chave pública, mas somente será decodificada por alguém que conhecer a correspondente chave privada.

A autenticação do servidor SSL permite ao usuário confirmar a identidade do servidor. O software do cliente com o SSL habilitado pode usar a técnica padrão de criptografia de chave pública para verificar se o certificado do servidor e seu ID público são válidos pela autoridade de certificação (Certificate authority – CA). Em nosso caso, esta confirmação é importante pois estaremos enviando a *username* e a senha da ferramenta e o lado cliente quer saber se está enviando para o servidor correto.

A autenticação do cliente SSL permite ao servidor confirmar a identidade do usuário. Utilizando da mesma técnica descrita na autenticação do servidor, este irá verificar-se que o certificado do cliente e seu ID público são válidos pela autoridade de certificação. Em nosso exemplo, esta confirmação é importante para o servidor, pois irá mandar informações contidas em seus arquivos e quer certificar-se que está enviando para o cliente válido.

Uma conexão encriptada SSL requer que todas informações tramitadas entre cliente e servidor seja encriptada pelo software de envio e desencriptada pelo software de destino, ambos provendo um alto grau de confidencialidade. Confidencialidade é importante para ambas as partes de qualquer transação privada. E também, todos os dados enviados sob uma conexão SSL encriptada está protegida por um mecanismo de detecção sensitiva, isto é, para automaticamente determinar quando um dado está sendo alterado durante a transmissão.

O protocolo SSL inclui dois sub-protocolos: o protocolo SSL *record* e o protocolo SSL *handshake*. O primeiro define o formato usado para transmitir os dados. O segundo envolve o protocolo SSL *record* para trocar uma série de mensagens entre o servidor habilitado SSL e o cliente habilitado SSL quando sua primeira conexão SSL for estabelecida. Esta troca de mensagens é designada para facilitar as seguintes ações: autenticar o servidor para o cliente; permitir ao servidor e cliente selecionar o algoritmo de criptografia, ou cifra, que ambos

suportem; opcionalmente autenticar o cliente para o servidor; usar técnicas de encriptação de chave pública para gerar compartilhamentos secretos; estabelecer uma encriptada conexão SSL.

Nenhum produto pode incorporar a tecnologia SSL sem pagar algum tipo de *royalties*. Mas existe uma implementação livre do SSL para propósitos de programação chamado *SSLey*. Este é o utilizado no serviço de encriptação SSL utilizado pela plataforma *Webmin*.

4.2.1. Por que utilizar transações seguras?

Muitas das informações passadas pela *Internet* não são particularmente atrativas para espíões ou *hackers*, elas podem ser lidas tão quanto possível. Mas existem algumas informações que são bastante atrativas, por exemplo, números de cartões de créditos e em nosso caso em especial a senha do sistema. Estas informações são transmitidas pela *Internet* do browser ao servidor. Em teoria, terceiros poderiam interceptar essas informações em algum ponto da rede entre o browser e o servidor. Para evitar esses possíveis inconvenientes, algumas formas de encriptação poderão ser usadas, de forma que se estas informações forem interceptadas, o autor dessa ação não terá como voltar à informação original. Dedutivamente, ambos, *browser* e servidor, necessitam do mesmo método de encriptação. Mundialmente, o sistema de encriptação mais implementado para a Web foi o descrito acima, SSL.

4.3. Compilações de mensagem – MD5

Uma crítica aos métodos de assinatura é a de que, com frequência, eles reúnem duas funções distintas: autenticação e sigilo. Geralmente, a autenticação é necessária, mas o sigilo não. Como a criptografia é lenta, normalmente as

peessoas preferem enviar documentos em textos simples assinados. A seguir, descreveremos um esquema de autenticação que não exige a criptografia da mensagem inteira [DE JONGE e CHAUM].

Esse esquema baseia-se na idéia de uma função de *hash* unidirecional que extrai um texto qualquer do texto simples e a partir dele calcula um string de tamanho fixo. Essa função de *hash*, geralmente chamada de *compilação de mensagem* – *message digest* – tem três importantes propriedades:

1. Se P for fornecido, o cálculo de MD(P) será muito fácil;
2. Se MD(P) for fornecido, será efetivamente impossível encontrar P;
3. Ninguém pode gerar duas mensagens que tenham a mesma compilação de mensagem.

Para atender ao critério 3, a função de *hash* deve ter pelo menos 128 bits; preferencialmente, mais que isso.

Calcular uma compilação de mensagem a partir de um trecho do texto simples é muito mais rápido do que criptografar esse texto simples com um algoritmo de chave pública; portanto, as compilações de mensagem podem ser usadas para agilizar algoritmos de assinatura digital.

Foram propostas diversas funções de compilação de mensagem. Uma das mais utilizadas é a MD5 [RIVEST]. MD5 é a quinta de uma série de funções de *hash* criadas por Ron Rivest. Ela funciona embaralhando os bits de uma forma complexa, onde todos os bits de saída são afetados por todos os bits de entrada. Resumindo, a função começa aumentando o tamanho da mensagem para 448 bits (módulo 512). Em seguida, o tamanho original é anexado como um inteiro de 64 bits, a fim de gerar uma entrada total cujo tamanho seja um múltiplo de 512 bits. A última etapa antes de os cálculos serem efetuados é inicializar um buffer de 128 bits com um valor fixo.

Agora começa os cálculos. Em cada rodada, um bloco de entrada de 512 bits é extraído e colocado num buffer de 128 bits. Para que os cálculos sejam feitos com uma maior precisão, também é incluída uma tabela criada a partir da função seno. Há quatro rodadas para cada bloco de entrada. Esse processo continua até que todos os blocos de entrada tenham sido consumidos. O conteúdo do buffer de 128 bits forma a compilação de mensagem [TAN].

O MD5 na plataforma Webmin é utilizada para encriptar pequenas mensagens, como por exemplo os comandos UNIX que são passados da ferramenta para o sistema, via Web.

4.4. O protocolo SSH

O *Secure Shell* é uma substituição segura para *Telnet*, *rlogin*, *rcp*, entre outros e provê tunelamento seguro para FTP. O SSH também pode ser usado junto com outros esquemas de autenticação como por exemplo o *SecurID*, *Kerberos*, *S/Key*, entre outros para prover um alto grau de segurança em acessos remotos para servidores UNIX.

4.4.1. Por que SSH?

Os comandos *Telnet*, *rlogin*, *rcp* têm uma grande fraqueza de segurança: toda a comunicação é realizada com textos limpos, qualquer máquina que estiver *escutando* o barramento poderá capturar as informações sem algum tipo de dificuldade.

O SSH usa autenticação forte RSA de chave pública/privada para checar as identidades de comunicação entre as máquinas, e encriptando todos os dados que estão sendo trafegados pela rede, assim como executar comandos em uma

máquina remota e copiar arquivos de uma máquina para outra, com algoritmos de autenticação forte, tal como, *blowfish*, 3DES, IDEA, etc.

A versão do SSH cliente utilizada para realizar o gerenciamento remoto para sistemas LINUX é a *ssh-1.2.27.5i* e o lado cliente utilizado para realizar gerenciamento remoto em ambiente *Windows* foi o *SSH32*.

4.5. Serviços implementados

A seguir estará descrito os serviços que foram gerenciados remotamente para a obtenção dos resultados alcançados. São eles:

4.5.1. Domain Name System – DNS

Raramente os usuários e programas fazem referências a *hosts*, *mailboxes* e outros recursos utilizando os endereços binários de rede. Em vez de números binários, ele utilizam strings ASCII, como *exemplo@subdomínio.domínio.com*. Todavia, a rede em si só compreende endereços binários, portanto, é necessário algum tipo de mecanismo para converter os strings ASCII em endereços de rede.

Na ARPANET, havia simplesmente um arquivo, *hosts.txt*, que listava todos os *hosts* e seus endereços IP. Para redes de algumas centenas de grandes máquinas, essa estratégia funcionava razoavelmente bem.

No entanto quando milhares de estações de trabalho foram conectados à rede, todos imaginaram que está estratégia não poderia ser utilizada para sempre. Por um lado, o arquivo poderia se tornar grande demais. No entanto o mais importante é que poderia haver conflitos de nomes em *host* constantemente, a menos que os nomes fossem gerenciados de uma forma central – algo totalmente fora de cogitação em uma enorme rede internacional. Para resolver esses problemas, foi inventado o DNS.

A essência do DNS é a invenção de um esquema de atribuição de nomes hierárquico, baseado em domínios. Ele é principalmente usado para mapear nomes de *hosts* e destinos de mensagens de correio eletrônico em endereço IP, mas também pode ser usados para outros fins.

Para resumir, o DNS é usado da seguinte maneira: Para mapear um nome em um endereço IP, o aplicativo – um browser, por exemplo – chama um procedimento de biblioteca denominado resolvidor – *resolver* – e passa seu nome para ele como um parâmetro. O resolvidor requisita uma consulta a um servidor de DNS local, este por sua vez irá pesquisar o nome e retornar o endereço IP para o resolvidor. Em seguida, o resolvidor retorna o endereço IP para o aplicativo que fez a chamada. Munido do endereço IP o programa pode estabelecer a conexão com o destino.

A *Internet* é dividida em muitas centenas de domínios de primeiro nível, onde cada domínio cobre muitos *hosts*. Os domínios são particionados em subdomínios, que também são particionados e assim por diante. Todos esses domínios podem ser representados por uma árvore, como mostra a figura a seguir. As folhas significam domínios que não possuem subdomínios, mas podem conter vários *hosts*.

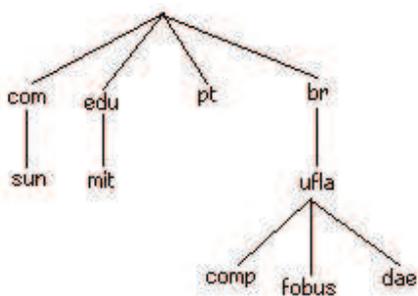


Figura 1 – Exemplo do espaço de nome de domínios da *Internet*

Agora pode-se ver claramente como estão distribuídos alguns domínios pela rede, como por exemplo a *sun.com*, *mti.edu*, *comp.ufla.br*, e assim por diante, sempre seguindo a estrutura de uma árvore. As consultas são realizadas utilizando-se do mesmo método de busca em árvores.

O espaço de nomes é dividido em *zonas* independentes. Cada zona contém uma parte da árvore e também servidores de nomes que armazenam informações oficiais referentes a essa zona. Normalmente, uma zona terá seu servidor de nomes principal, que obtém suas informações a partir de um arquivo contido em sua unidade de disco e um ou mais servidores de nomes secundários, que obtêm suas informações a partir do servidor de nomes principal. Para melhorar a confiabilidade, alguns servidores de uma zona poderão estar focalizados fora dela.

A localização das fronteiras fica a cargo dos administradores de sistemas, essa decisão é realizada de acordo com o número de servidores de nomes desejados. Por exemplo, no domínio UFLA, tal decisão pode ser tomada quando um departamento, como o de administração, não deseja ter seu próprio servidor de nomes, mas outro, como o de ciência da computação, deseja. Conseqüentemente, *comp.ufla.br* é uma zona separada e *dae.ufla.br* não é. [TAN]

4.5.2. O protocolo SMB – Server Message Block

SMB é um protocolo no qual uma quantidade de máquinas PC conectadas podem compartilhar arquivos e impressoras ou outras informações entre alguns tipos de sistemas operacionais, tais como, *Windows9x*, NT. E que permite que uma máquina UNIX/LINUX atue como servidor ou estação de trabalho em redes mistas. O SMB é utilizado por pessoas que queiram integrar

seus sistemas *Microsoft* ou IBM com seus servidores UNIX/LINUX, servidores *Microsoft* com servidores UNIX/LINUX, por exemplo.

Em nosso estudo, o SMB é utilizado de maneira a substituir as características de um servidor NT por um servidor Linux. Assim, as estações de trabalho, tanto *Windows9x*, *Windows NT* quanto Estações Linux, efetuam seus *logons* e o servidor Linux por intermédio do protocolo SMB faz a autenticação dos usuários comportando-se como em um servidor NT, assim disponibilizando os arquivos pessoais do respectivo usuário, assim como impressoras, e outros recursos que venham a estar configurados para tal.

Os serviços do Samba são executados por dois *daemons*¹. O *daemon* SMB – *smbd* fornece os serviços de compartilhamento de arquivos e impressões. E o *daemon* Servidor de Nomes NetBIOS – *nmbd* fornece o serviço de nome NetBIOS para endereço IP, assim, mapeando nomes para seus clientes.

4.5.3 – Servidor WWW – APACHE

A World Wide Web é a arquitetura que permite o acesso a documentos vinculados espalhados por milhares de máquinas na *Internet*. Alguns anos atrás, ela deixou de ser um meio para distribuir dados extremamente técnicos para se tornar a aplicação que milhões de pessoas consideram ser *A Internet*. Sua enorme popularidade se deve à sua interface gráfica colorida, que é de fácil utilização para principiantes. Além disso, ela possui uma imensa variedade de informações sobre quase todos os assuntos imagináveis.

Do ponto de vista do usuário, a Web é uma vasta coleção mundial de documentos, geralmente chamados de páginas. Cada página pode conter links para outras páginas ou arquivos para *download* em qualquer lugar do mundo. Essas páginas são visualizadas com o auxílio de um programa denominado

¹ Daemons são processos cuja função é detectar uma solicitação do serviço por ele prestado.

browser, estes oferecendo muitos botões e recursos para tornar a navegação pela Web mais fácil.

No lado do servidor existem processos que escutam a porta 80 TCP aguardando conexões dos clientes. Depois de estabelecida a conexão, o cliente envia uma solicitação e o servidor envia uma resposta. A conexão, por sua vez, é liberada. O protocolo que define as solicitações e respostas válidas é chamado de HTTP – *Hiper Text Transfer Protocol*. Será agora mostrado por meio de um exemplo, como ocorre uma conexão. Supondo que o usuário tenha digitado a URL `http://www.comp.ufla.br/pasta1/teste.html`. Uma URL possui três partes: o nome do protocolo – http – o nome da máquina onde está localizada – `www.comp.ufla.br` – e o nome do arquivo onde está a página - `/pasta1/teste.html`. os procedimentos que irão ocorrer são os seguintes [TAN]:

1. O browser determina a URL;
2. O browser pergunta ao DNS qual é o endereço IP de `www.comp.ufla.br`;
3. O DNS responde 200.131.251.170;
4. O browser estabelece uma conexão TCP com a porta 80 em 200.131.251.170;
5. O browser envia um comando GET/pasta1/teste.html;
6. O servidor envia o arquivo teste.html;
7. A conexão TCP é liberada;
8. O browser apresenta todo o texto de teste.html;
9. O browser apresenta todas as figuras de teste.html se houver.

Em nosso estudo foi utilizado o APACHE como servidor www.

O APACHE é um software desenvolvido para ser robusto e principalmente, uma implementação gratuita de um servidor Web. Ele surgiu em 1995, e é atualmente o servidor Web mais utilizado na *Internet* brasileira,

segundo a pesquisa realizada pelo Insite.com.br. O APACHE suporta CGI e PERL, bem como servidores *proxy*. Através do APACHE é viável a criação de domínios virtuais, aqueles domínios que não têm referência com o domínio do servidor, por exemplo, no domínio ufla.br, comp.ufla.br é considerado um subdomínio, mas uflatec.com.br, é um domínio virtual.

No projeto APACHE qualquer pessoa pode se inscrever para poder contribuir auxiliando na implementação, principalmente em correção de bugs, bastando para isso contribuir freqüentemente em sua lista de discussão, ou ser aprovado por votação dos membros efetivos.

4.5.4 File Transfer Protocol – FTP

Como diz o nome, Protocolo de Transferência de Arquivos, o protocolo FTP é usado para acessar arquivos por FTP, é o protocolo de arquivos da *Internet*. O FTP já existe há mais de duas décadas e conquistou o seu espaço. Um grande número de servidores FTP em todo o mundo permite que pessoas de qualquer lugar da *Internet* estabeleçam *login* e façam *downloads* de quaisquer arquivos que estiverem armazenados nos servidores de FTP e que esteja configuradas as permissões para tal.

Um usuário da *Internet* não precisa necessariamente ser um usuário do servidor de FTP requisitado para prestar este serviço, basta para isso o servidor em questão estar configurado para *logons anônimos*, e é desta maneira que funciona a maioria das requisições de transferências de arquivos pela *Internet* ou intranet.

4.5.5. Majordomo

Majordomo é um programa de distribuição gratuita que automatiza o gerenciamento de listas de *e-mail*. Os comandos do Majordomo são enviados para o servidor de *e-mail* que o tem implementado através de *e-mails*, por exemplo *majordomo@fobus.ufla.br* para executar a função contida no mesmo. Por exemplo, um usuário pede para cadastrar-se em uma determinada lista gerenciada pelo Majordomo. Então, o usuário receberá do Majordomo uma confirmação do desejo de cadastrar-se na lista, bastando ao usuário confirmá-la. Estes e outros comandos podem ser passados para o Majordomo e este a realizará ao invés de ser executada manualmente, ou seja, uma vez certa lista está configurada, virtualmente todas as operações podem ser realizadas remotamente por *e-mail* não requerendo uma suposta intervenção de um administrador. O criador da lista, insere um e-mail para este ser o moderador da lista, ou seja, qualquer ação que ocorrer na utilização daquela lista que violou as regras de configuração, seja reportada para esse moderador.

Majordomo é escrito em linguagem Perl e desenvolvido para executar em ambientes UNIX/LINUX, mas ele também pode trabalhar em outras plataformas desde que esta possua um Perl para compilar e rodar em seu sistema.

Majordomo é um projeto *groupware*.

4.6. Metodologia

A metodologia utilizada para defender a hipótese apresentada na Seção 3 foi uma Análise Heurística de Usabilidade, descrita na Seção 2, da ferramenta de interface gráfica Webmin. Outro método que foi utilizado para tal, são os resultados práticos que foram obtidos com o gerenciamento dos serviços

descritos nas seções 4.5.1 a 4.5.5 via Webmin com os resultados obtidos com o gerenciamento dos serviços via terminal.

Os resultados práticos são analisados quanto ao números de arquivos que os serviços citados nas seções 4.5.1 a 4.5.5 necessitam ser configurados para o seu funcionamento correto, ilustrando melhor o comportamento da ferramenta gráfica, no nível de sua funcionalidade. Assim, deixando claro, que em nosso espaço amostral, com 13 servidores a serem gerenciados, cada um atendendo às necessidades de cada departamento, a interface gráfica será uma poderosa ferramenta no apoio ao gerenciamento remoto.

5. RESULTADOS E DISCUSSÕES

Iniciaremos esta seção com uma Análise Heurística de Usabilidade superficial, conforme descrito na Seção 2, da ferramenta gráfica Webmin, para que possam ser apresentados ao leitor algumas qualidades, e também alguns defeitos encontrados na ferramenta.

Observando as telas extraídas do Webmin, onde estão localizados os serviços que estão sendo apresentados ao longo deste, serão comentados alguns tópicos da Análise Heurística de Usabilidade. Note que são somente duas telas em seqüência, pois inserir todas ficaria inviável. A ferramenta Webmin, segue um padrão para todas as telas, de modo que a partir das apresentadas aqui pode-se perceber como seguem as demais telas da ferramenta em questão.

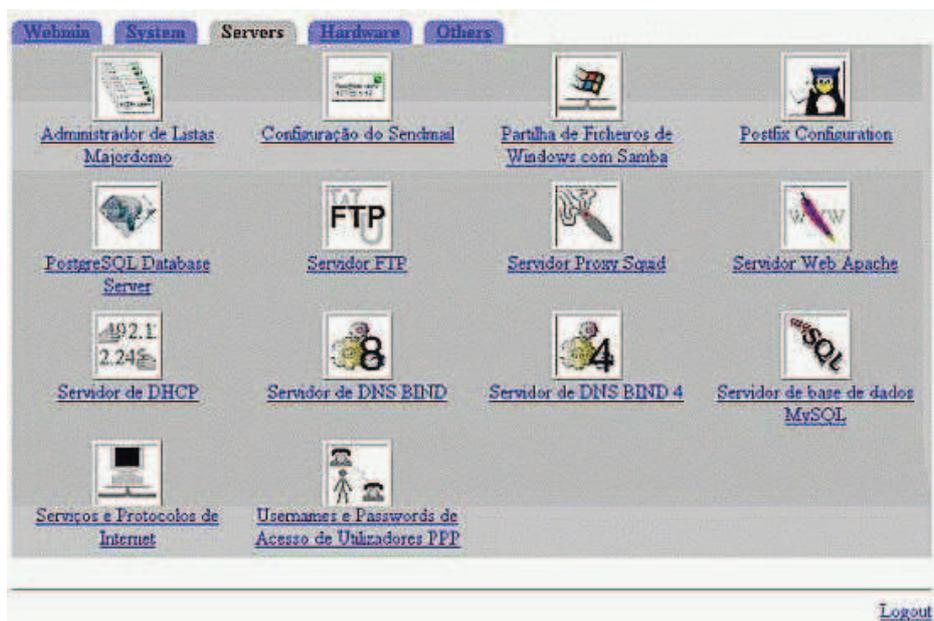


Figura 2 – Tela de serviços do Webmin

A tela apresenta um *Diálogo simples e natural* pois se o usuário necessita gerenciar um serviço, basta escolher a aba *servers* e a tela da Figura 2 irá aparecer. Abordando ainda sobre este tópico, se o usuário continuar pelos botões de navegação, os objetos e operações do sistema a serem realizadas, estarão sendo acessadas numa *seqüência* que maximize a eficiência e produtividade do usuário.

Esta interface também está *falando a linguagem do usuário*, lembrando sempre que o usuário em questão é o administrador de sistemas, pois as imagens que são ilustradas junto aos nomes dos serviços utilizam de uma metáfora que é conhecida pelos administradores, por exemplo, administrador de listas Majordomo é ilustrada por “vários envelopes”, o servidor Web Apache por uma “pena indígena” – indicando o serviço www realizado pelo software Apache, entre outros.

Quando a interface está em suas telas iniciais, sempre irá aparecer para o usuário, a opção de *logout*, para o usuário realizar sua saída do sistema. Quando não, haverá sempre a opção de retornar no canto inferior da tela. Essa ação se enquadra nas *saídas claramente marcadas*, o sistema deve prover para o usuário uma saída fácil e explícita.

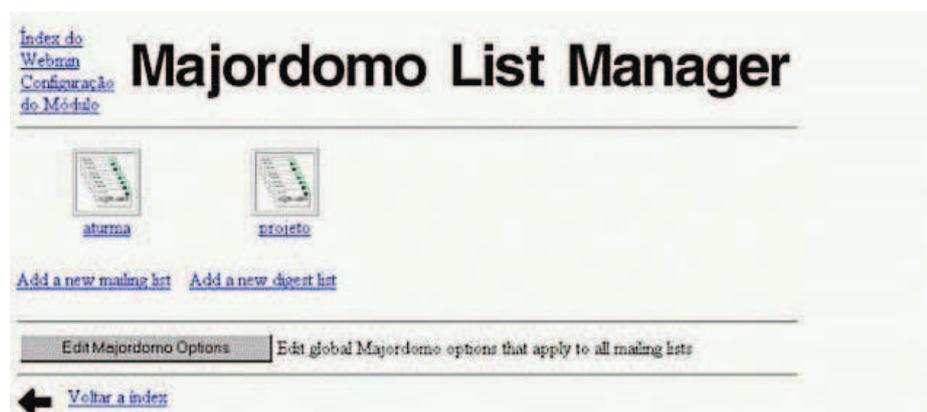


Figura 3 – Tela de gerenciamento do Majordomo

Para fixar o que foi mostrado nos parágrafos anteriores, a tela da Figura 3 ilustra melhor a seqüência do sistema, sempre apresentando uma saída para o usuário. E também *fala a linguagem do usuário* com as ações claramente marcadas, como por exemplo, adicionar uma nova lista.

A ferramenta Webmin apresenta algumas deficiências quanto a sua usabilidade, como por exemplo, não existe o *retorno*, pois quando o sistema necessita de tempos de respostas longos, digamos de 5 a 10 segundos, não existe para o usuário a opção de cancelar a operação que está utilizando desse longo período. Outra desvantagem diz respeito às *mensagens de erros*, pois alguns erros são provenientes do sistema operacional sob o qual a ferramenta está operando. Quanto ao tópico de mensagens de erro, o Webmin possui uma vantagem quanto ao gerenciamento sem interface, se o usuário preenche um campo incorretamente, o Webmin exibe uma mensagem de erro indicando que o campo não pode ser inserido naquele formato, e não salva o dado incorreto nos arquivos de configuração do sistema, ao contrário do gerenciamento sem interface, que salvaria e na hora de aplicar as alterações reinicializando o serviço simplesmente não inicializaria ou inicializaria com erros.

Agora serão ilustrados alguns exemplos extraídos dos resultados práticos quanto ao gerenciamento remoto aos serviços mencionados nas duas formas levantadas pelo trabalho. Serão apresentados os serviços e quais os passos mais significativos, em termos de configuração de seus arquivos, que o administrador de sistemas teria de realizar nas duas formas de gerenciamento. Assim, mostrando que na situação descrita do problema de gerenciamento da UFLA, a gerência remota para esses serviços com o auxílio de uma interface gráfica, seria menos árdua para os administradores de sistemas da UFLA.

Os arquivos de configurações completos e qual a função dos mesmos estarão nos anexos correspondentes a cada serviço a partir daqui descrito.

5.1. Domain Name System – DNS

Depois de instalado e iniciado o serviço [APÊNDICE A], criou-se zona chamada *fobus.ufla.br* e criou-se um *alias* *www*, nome para correio eletrônico, e o cadastro de um nome de uma máquina.

Este é um dos serviços que mais necessita de gerenciamento, pois quando novas máquinas são adicionadas na rede seu número IP necessita estar configurado no DNS, além de outras tarefas que são realizada por esse serviço.

Via terminal: primeiramente o administrador deverá editar o arquivo *named.conf* em */etc* e digitar as seguintes linhas de comando:

```
zone "fobus.ufla.br" {           //nome da zona
type master;                   // tipo da zona, se é mestre ou escrava
file "/var/named/fobus.forward"; // endereço de onde estará o arquivo de
                                configuração da zona
}; //Zona criada
```

Depois, deverá ser criado um arquivo com o nome de *fobus.forward* em */var/named* com as seguintes linhas de comando:

```
$ttl 38400
fobus.ufla.br. IN      SOA   fobus.ufla.br. root.fobus.ufla.br. (
                        976379845
                        10800
                        3600
                        432000
                        38400 ) // Valores default
fobus.ufla.br.        IN     NS    fobus.ufla.br.
aragon.fobus.ufla.br. IN     A     200.131.250.49 //máquina cadastrada
fobus.ufla.br.        IN     A     200.131.250.47
www.fobus.ufla.br.    IN     CNAME  fobus.ufla.br. //alias
fobus.ufla.br.        IN     MX     5 fobus.ufla.br. //nome de correio
```

Finalmente o último passo seria dar o comando de *restart* no serviço *named*.

Via Webmin: Na tela de serviços mostrada pela Figura 2, o administrador daria um *click* para entrar no DNS BIND 8. Logo apareceria a tela da Figura 4:

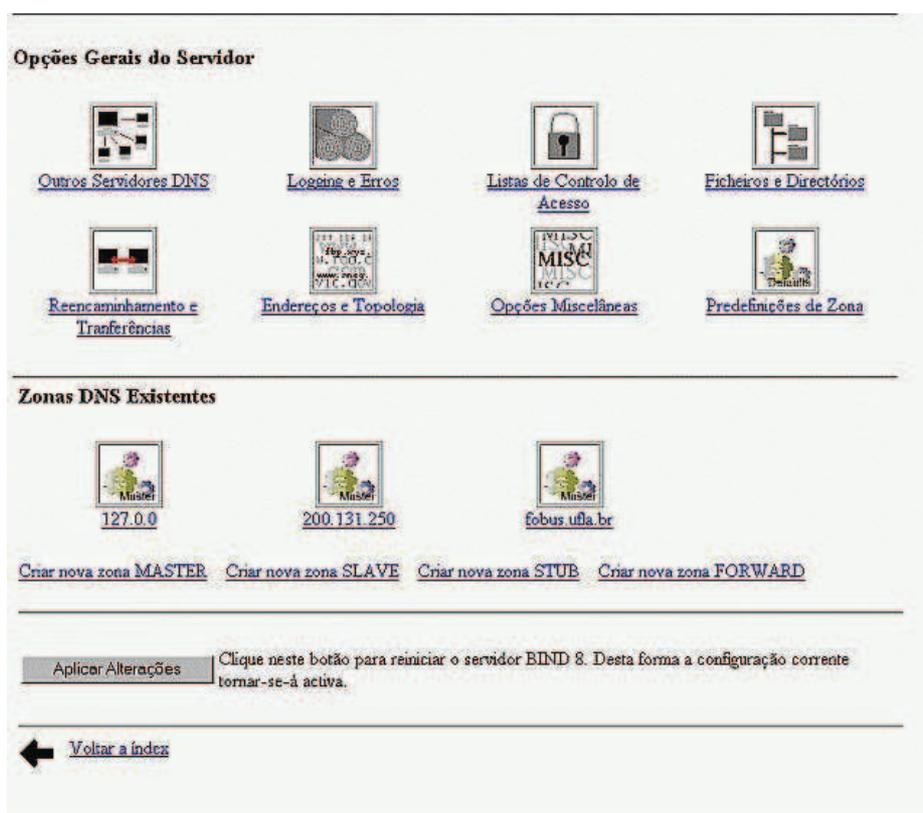


Figura 4 – Tela inicial do serviço DNS

Em seguida, um *click* em *Criar nova zona master* depois preencheria os campos referentes a tipo de zona, nome da mesma, o arquivo onde estarão os dados referentes a essa zona e os valores *default* são pré-estabelecidos. E *clicaria* em *criar* como na Figura 5:

Novas opções de zona master

Tipo de zona Forward (Nomes para Endereços) Reverse (Endereços para Nomes)

Nome do domínio / Network

Ficheiro de registos Automático

Servidor Master Add NS record for master server?

Endereço de email

Use zone template? Sim Não IP address for template records

Tempo de refrescagem seconds seconds

Tempo de expiração seconds seconds

Figura 5 – Tela para criação de uma zona para o serviço de DNS

Então retornará a tela da Figura 4, bastando clicar no ícone correspondente a zona criada e então aparecerá uma nova tela, ilustrada na Figura 6.

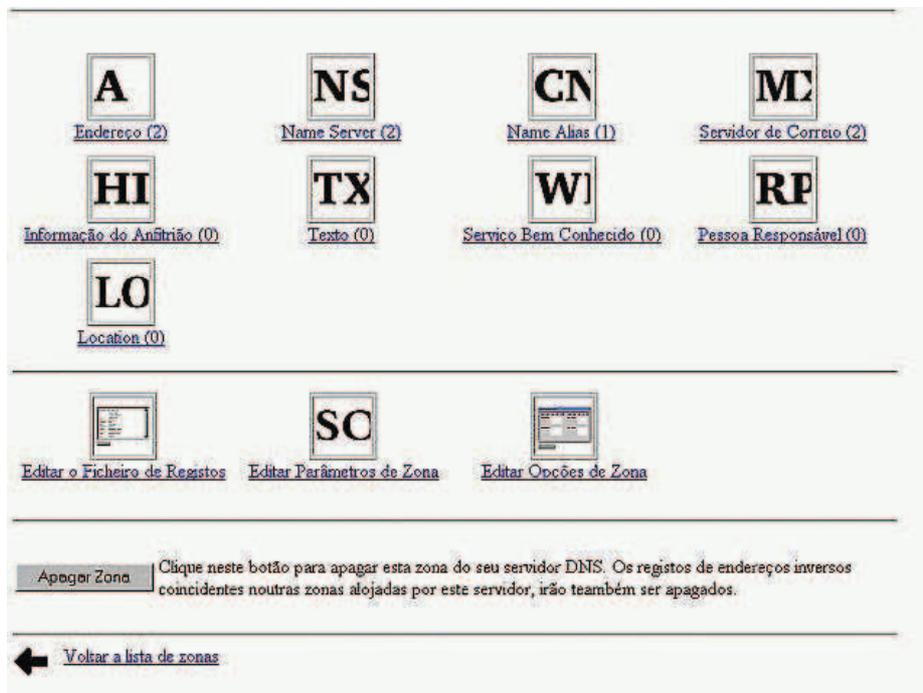


Figura 6 – Tela de gerenciamento de uma zona específica

Onde o administrador *clikaria* em *name alias* preenchendo o campo com o *alias* pretendido e salvando em seguida. *Clikaria* em *servidor de correio* e preencheria os campos para a criação do nome de correio, e finalmente *clikaria* em *endereço* e preencheria os campos para o cadastro da máquina. Depois retornando a Figura 4 e *clikaria* em aplicar as alterações.

Uma outra vantagem da ferramenta, é que para *chamar* uma máquina pelo nome e não pelo número IP, a mesma tem de estar cadastrada na *zona* citada acima e em outra que contém endereços dos números para nomes, vide APÊNDICE A, e o Webmin tem uma opção que logo cadastrado um nome, a ferramenta atualiza automaticamente a outra *zona* de endereços IP, uma tarefa que via terminal estaria a cargo do administrador de sistemas.

Note que a navegação é bem intuitiva, e configuraria os arquivos mostrados anteriormente da mesma maneira, somente com o preenchimento dos campos necessários o restante é inserido pelo *script* da ferramenta.

5.2. O protocolo SMB – Server Message Block

O arquivo de configuração [APÊNDICE B] do SMB é o *smb.conf*. Seu formato *default* já é bastante completo e bem comentado, deixando para o administrador, apenas a função de completar ou substituir as configurações *defaults*. No Webmin, não poderia ser diferente pois a ferramenta realiza a leitura do arquivo e somente fica a cargo do administrador substituir as configurações para os valores de que necessita.

Mas o grande auxílio da ferramenta é quando a opção de encriptação de senhas é ativada. Nesse caso será necessário utilizar um arquivo de senhas diferente do arquivo padrão do Linux, o *passwd*, no qual seria o arquivo *smbpasswd*. Quando utilizada a ferramenta Webmin, o administrador poderá realizar as configurações no novo arquivo de senhas com mais facilidades do que via terminal, pois a ferramenta apresenta três importantes funções que estão ilustradas na Figura 7.

Share Name	Path	Security
homes	All Home Directories	Read/write to all known users
printers	All Printers	Printable to all known users
FTP	/home/ftp/pub	Read only to everyone

[Create a new file share](#)
 [Create a new printer share](#)
 [Create a new copy](#)
 [View All Connections](#)

Global Configuration


[Unix Networking](#)


[Windows Networking](#)


[Authentication](#)


[Windows to Unix Printing](#)


[Miscellaneous Options](#)


[File Share Defaults](#)


[Printer Share Defaults](#)


[SWAT](#)

Encrypted Passwords

- [Edit Samba users and passwords](#)
- [Convert Unix users to Samba users](#)
- [Configure automatic Unix and Samba user synchronisation](#)

Restart Samba Servers

Click this button to restart the running Samba servers on your system. This will force the current configuration to be applied.

 [Voltar a index](#)

Figura 7 – Tela inicial de gerenciamento do serviço SMB

Funções estas que são: *Editar os usuários do samba*, esta configura diretamente o arquivo de senhas do SMB; *Converter os usuário Unix para o samba*, esta é realizada normalmente somente uma vez, pois converte os usuários já existentes no servidor para usuários SMB, então estes poderão usufruir do serviço em questão, uma vez que via terminal seria feito um-a-um. E finalmente a que pode ser considerada a mais importante *Sincronizar os arquivos passwd e smbpasswd*, isto é, o administrador poderá escolher pelas opções: Adicionar um usuário SMB quando um usuário do sistema for adicionado; as alterações que forem feitas para o usuário do sistema será feita para o usuário SMB; e excluir usuário SMB quando excluir usuário do sistema.

As demais opções de configuração do serviço são simples nas duas formas de gerenciamento, pois como já foi mencionado, o arquivo padrão de configuração já é bastante comentado e intuitivo.

5.3. Servidor WWW – Apache

Igualmente ao caso do SMB, o arquivo de configuração do Apache [APÊNDICE C], *httpd.conf*, é bem comentado e de fácil entendimento. Contudo, este arquivo de configuração é bastante extenso, assim utilizando-se mais uma vez do auxílio da interface gráfica, a sua configuração em 13 servidores ocorreria de forma menos cansativa para os administradores. Configurações essas, como por exemplo, as indexações dos diretórios, isto é, quais arquivos devem ser procurados no diretórios, controle de acesso, criação de domínios virtuais, ou seja, aqueles que não pertencem ao seu domínio, criar alias para páginas, entre outros.

Na Figura 8 podemos conferir a tela principal do servidor www Apache.

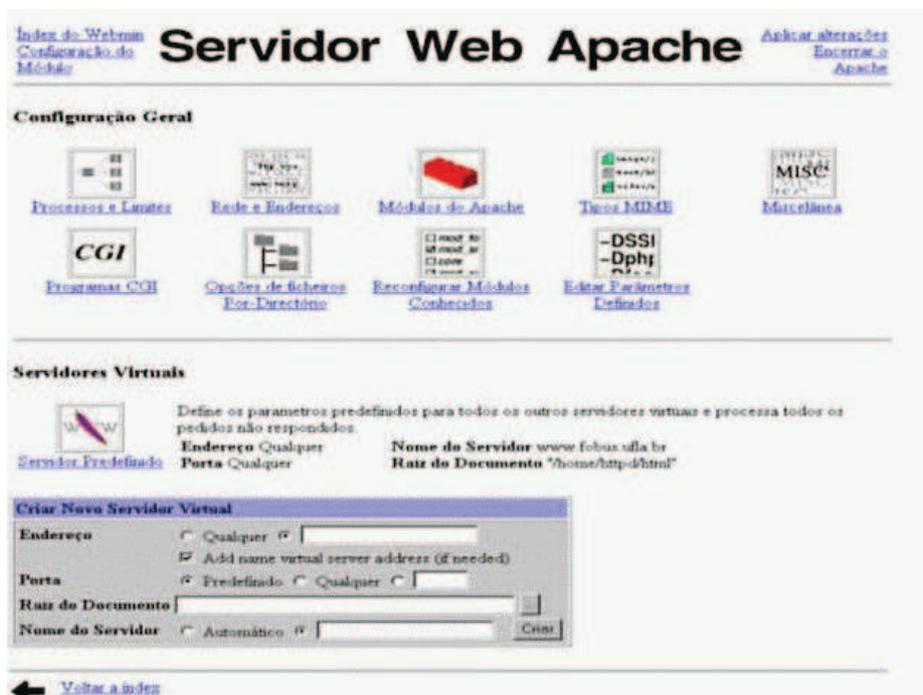


Figura 8 – Tela inicial de gerenciamento do serviço www

Aqui o que pode ser destacado é a seção de servidores www, onde estão as configurações do servidor *default* e as configurações dos servidores virtuais, quando existirem. A criação de um domínio virtual com a ferramenta Webmin é menos árdua que a sua criação via terminal, pois, a quantidade de linhas de comandos inseridas no arquivo de configuração pode possuir grandes números, enquanto que os *scripts* do Webmin podem fazê-lo sem menores preocupações.

5.4. File Transfer Protocol – FTP

Dos serviços apresentados neste trabalho, o FTP é incontestavelmente o mais simples de configurar [APÊNDICE D]. Seu arquivo de configuração *ftppass* é bastante simples de entender e configurar. A criação de alias para

FTP, controle de acesso, entre outros, também podem ser realizadas sem menores dificuldades tanto no Webmin quanto via terminal.

A Figura 9 está mostrando a tela de configuração do serviço de FTP.

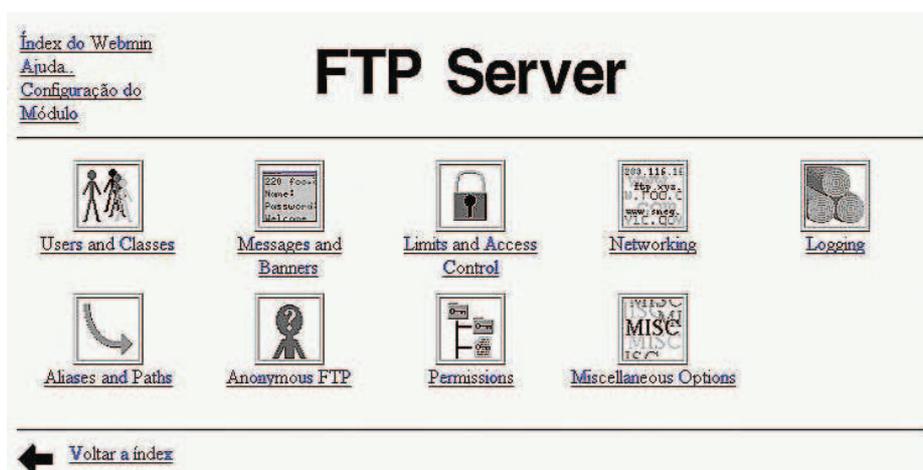


Figura 9 – Tela de gerenciamento do serviço FTP

5.5. Majordomo

A ferramenta gráfica Webmin é bastante útil no auxílio ao gerenciamento desse serviço. A configuração do Majordomo [APÊNDICE E] consulta vários arquivos em seu funcionamento.

O administrador é acionado para criar uma lista chamada *teste*, os passos via console seria o seguinte: criar dois arquivos, *teste* onde estariam armazenados os *e-mails* dos membros da lista, *teste.config* onde estaria a configuração da lista, por exemplo controle de acesso, política de cadastramento, tamanho máximo da mensagem, entre outros, no diretório de listas configurado em *majordomo.cf*; criar alias de correio no arquivo *aliases*, dizendo para onde será encaminhado as mensagens que vierem para este endereço, no caso, estaria

apontando para o arquivo com os membros da lista, e aliases indicando quem é a pessoa responsável pela lista.

Já na ferramenta Webmin, o administrador apenas precisaria *clicar* em *add new mailing list* como mostrado na Figura 3 e preencher os campos mostrados na Figura 10. Então, o Webmin criou os arquivos no local especificado, criou os *aliases* de correio, e um ícone simbolizando a lista criada, que irá conter as configurações da lista, ficando a cargo do administrador configurá-la de acordo com as necessidades do serviço.

The image shows a web form titled "New list details". It contains several input fields and radio buttons. The fields are: "List name", "List maintainer's address", "Maintenance password", "Description", "Introductory message", and "Forwarded mail footer". The "Introductory message" and "Forwarded mail footer" fields are larger and have scrollbars. Below these fields, there are two radio button options: "Moderated list?" with options "Sim" and "Não", and "Moderator's address" with options "Same as maintainer" and an empty input field. At the bottom left of the form is a button labeled "Criar". Below the form, there is a link that says "Voltar a mailing lists".

Figura 10 – Tela de criação de uma lista para o serviço Majordomo

5.6. Discussões

Seguindo as plataformas de gerenciamento descritas por [JMARCOS], pode-se descrever o tipo de arquitetura que é utilizada na UFLA. Ela está enquadrada na arquitetura hierárquica, pois existem máquinas atuando como

subservidores distribuídas pelos departamentos, com atuação remota de gerenciamento e as decisões de gerência são tomadas de um ponto central.

A escolha de uma ferramenta de apoio à gerência de sistemas como o Webmin, por exemplo, no estudo de caso da UFLA, está relatado por [SMP] onde este ressalta que o gerenciamento de redes de computadores tem se tornado uma atividade cada vez mais complexa devido, principalmente, à diversidade de plataformas de hardware e software que compõem as redes. A complexidade da atividade de gerenciamento requer a utilização de um Sistema de Gerenciamento de Redes – SGR – eficiente. Dentre esses SGRs podem estar inclusos ferramentas de gerenciamento remoto.

A utilização de uma aplicação gráfica no auxílio a gerência de sistemas foi introduzido nos trabalhos de [SJC] e [ROGERIO e MARCOS].

O primeiro relata a importância da tecnologia Web, como uma facilidade na arte de gerenciamento remoto, destacando que dessa maneira há uma independência do sistema operacional do administrador, pois através de sua interface Web a portabilidade entre as plataformas é garantida.

O segundo explica que o uso de sistemas gráficos em gerenciamento provê uma maior eficiência e robustez na atuação dos utilizadores, no qual são aqui representados pelos administradores de sistemas. Podendo interagir com os elementos gerenciados de uma forma única e amigável. Estes elementos gerenciados são aqui representados pelos serviços nesta listados.

Apesar dos trabalhos mencionados na Seção 2 estarem refletindo a importância do gerenciamento remoto e a utilização de ferramentas gráficas na realização de tal tarefa, a questão da comparação entre gerenciamento remoto via Web e gerenciamento remoto via terminal tem um certo caráter inovador por abordar detalhes de usabilidade das ferramentas gráficas e gerenciamento ao nível de serviços em servidores de redes locais. Apesar de [SJC] estar abordando um sistema de suporte ao nível de serviço, existe uma restrição do estudo aqui

apresentado em relação ao de [SJC]: o último abrange esses serviços em uma área geograficamente maior que a apresentada aqui. Enquanto este refere-se no nível de serviços em servidores de redes locais, ou seja em LANs.

6. CONCLUSÕES

A LAN da Universidade Federal de Lavras necessita de um processo de migração dos serviços realizados pelo servidor central para os subservidores instalados em cada departamento pelo fato da rápida expansão da rede e suas centenas de usuários.

Como, por enquanto, não há condições de alocar um administrador de sistemas em cada departamento, os administradores de sistemas da UFLA deverão gerenciar os subservidores de todos os departamentos no Centro de Informática, CIN. Então está caracterizada a situação de gerenciamento remoto, pois não há necessidade de deslocamento dos administradores por todo o *campus* para solução de problemas com os serviços descritos aqui ou atender as necessidades do departamento quanto a configurações dos mesmos.

O número de servidores que estará sob a gerência do CIN é 13, portanto deve ser pensado numa forma de realizar a tarefa de gerenciamento, além do terminal remoto, que auxilie os administradores e maximize suas produtividades na realização de tal tarefa. Para tal auxílio, pode-se pensar em uma ferramenta que possua uma interface gráfica amigável, ou seja, de fácil aceitação pelo usuário, que no caso seriam os administradores de sistemas, de preferência gratuita e que funcione de forma adequada.

A ferramenta que atendeu bem as necessidades em questão é o Webmin, pois num problema como do tráfego gerado no servidor central, a migração desses serviços descritos nesta para os subservidores e a gerência das 13 máquinas a serem realizadas por um número de pessoas que é menos de um quarto desse montante, o Webmin tem atendido bem aos resultados obtidos até o momento.

O uso da tecnologia Web é bem aproveitada devido a heterogeneidade da rede UFLA, garantindo assim sua fácil visualização e sua disponibilidade e

funcionalidade em qualquer arquitetura ou sistema operacional no qual for solicitada.

Além de atender quanto a sua funcionalidade e disponibilidade, ela atendeu também quanto ao quesito segurança, pois além de utilizar da encriptação SSL, existem opções que restringem o controle de acesso a ferramenta, impedindo ações maliciosas e garantindo a integridade dos dados.

Contudo, deve ser salientado que em poucos casos foi necessária alguma intervenção via terminal, não por uma falha do software de gerência ou que o serviço não funcionava, mas sim, como um complemento para a configuração do serviço. Nesses casos, foi apenas uma configuração extra à necessária, mas tendo a ferramenta Webmin feito o serviço *pesado*, deixando para o administrador, apenas inserir pequenas instruções no arquivo de configuração em questão.

Portanto, conclui-se que, quando uma LAN possui um bom número de máquinas a serem gerenciadas de um ponto central, existe a necessidade de gerenciamento remoto e uma ferramenta que possa aumentar a produtividade do administrador, otimizando seu tempo de serviço, sem que o profissional desperdice uma parte de seu tempo com detalhes de configurações, sendo que existe uma ferramenta para tal. Uma ferramenta gráfica trabalhando em conjunto com um terminal são armas valiosas para qualquer administrador de sistemas em seu gerenciamento remoto.

6.1. Limitações do estudo

O cenário teve como estudo somente o caso da Universidade Federal de Lavras. Contudo, a conjectura do problema de gerenciamento ocorrido na instituição pode, naturalmente, ocorrer em qualquer outro estabelecimento onde sua LAN tenha um crescimento como o aqui descrito ou em uma instituição em

que sua rede de computadores seja criada com um grande número de estações de trabalho ou seus servidores estejam com grandes distâncias geográficas como por exemplo, cidades diferentes.

Outra limitação que é louvável ser destacada é com respeito ao número de ferramentas com interfaces gráficas que foram escolhidas para o estudo. Existem alguns fabricantes de ferramentas de gerenciamento de redes, mas as suas soluções são pagas, portanto inviabilizando sua aquisição. Dentre alguns encontrados destacam-se: InfoVistaTM, Netsys, HP IT Service Management.

Existiu outra limitação. Todo o estudo foi desenvolvido em um servidor de teste, das situações descritas nas seções 5.1 a 5.5, onde os resultados simularam o que irá ser feito no processo de migração dos serviços do servidor central para os subservidores. Esses resultados simulados estão de acordo com o que será esperado na migração dos serviços.

6.2. Sugestões de novos estudos

Os endereços IPs em sua forma atual – IPv4 – estão com os dias contados. Essa afirmação é verdadeira pelo fato da explosão de máquinas que estão interligadas à *Internet* atualmente, e em um futuro próximo existirão ainda mais usuários, algumas empresas de televisão por assinatura estão investindo para que aparelhos de televisão possam ter acesso a *Internet*. Isso significa que os endereços IPs se esgotarão um dia. Sob essas circunstâncias, ficou claro que o IP precisa ser aperfeiçoado para se tornar mais flexível.

Depois de lançada a questão, existiram várias propostas de solução, mas a que obteve maior aceitação foi a de [DEERING] e [FRANCIS], à qual foi atribuída a designação IPv6 ou IPNg – *IP Next Generation*.

Dentre os principais recursos do IPv6 estão: o IPv6 tem endereços mais longos que o IPv4. Eles têm 16 bytes, o que resolve o problema a que o IPv6 se

propõe: oferecer um número ilimitado de endereços para a *Internet*; Seu cabeçalho é mais simples que o IPv4, essa mudança permite que os roteadores processem os pacotes com maior rapidez, entre outras.

O objetivo aqui não é entrar em detalhes do IPv6, mas sim apresentar a nova tendência. Quem quiser se aprofundar no assunto vide [TAN-IPv6].

Visto a nova tendência para endereços IPs, sugere-se fazer um estudo sobre como ficaria a implementação dos serviços aqui apresentados na futura versão dos endereços IPs, e seu gerenciamento remoto.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- [DE JONGE e CHAUM] De Jonge, M.; Chaum, D. *Some Variators on RSA signatures and their security*. **Advances in criptografy**. Odlyzko, A.M. New York : Springer – Verlang, 1987.
- [RIVEST] Rivest, R.L. *The MD5 message digest algorithm*. **RFC 1320**. abril de 1992.
- [TAN] Tanenbaum, A.S. **Redes de Computadores**. 4.ed. Rio de Janeiro : Campus, 1997. cap.7.
- [TAN-IPV6] Tanenbaum, A.S. **Redes de Computadores**. 4.ed. Rio de Janeiro : Campus, 1997. cap.7. 498-513p.
- [JMARCOS] Nogueira, J.M.S. **Artigo** : *O estado atual e desenvolvimento em gerenciamento de redes no Brasil*. **2º Seminário Franco Brasileiro de Sistemas Informáticos Distribuídos**, Fortaleza, CE, 03-07/11/1997.
- [NIELSEN] Nielsen, J. **Usability Engineering**. Chestnut, MA : Academic Press, 1993.
- [SMP] Teixeira, S. de Q. R.; Oliveira, M.; Cunha, P.R.F. **Artigo** : *Disponibilização de conhecimento em redes de computadores*. **XXIII SEMISH**, Recife – PE. agosto de 1996.
- [SJC] Miranda, S. V. de; Nogueira, J.M.S.; Machado, C. M. **Artigo** : *Um sistema de suporte ao gerenciamento do nível de serviço*. **18º SBRC**, Belo Horizonte – MG. 2000.
- [ROGÉRIO e MARCOS] Barros, R.A. de; Nogueira, J.M.S. **Artigo** : *Desenvolvimento e uso de aplicações gráficas no auxílio a sistemas de gerência de redes de telecomunicações*. **XV SBT**, Recife – PE. 8-11/09/1997.

- [DEERING] Deering, S.E. *SIPP: Simple Internet Protocol Plus*. **IEEE Network Magazine**. vol. 8. 16-28p. maio/junho de 1990.
- [FRANCIS] Francis, A.G. *A Near-Term Architecture for Deploying Pip*. **IEEE Network Magazine**. 30-37p. maio/junho de 1993.

APÊNDICE A

Arquivo *resolv.conf*. Neste arquivo configura o cliente DNS de um sistema UNIX/LINUX.

```
//resolv.conf
search fobus.ufla.br ufla.br // domínios onde serão realizadas consultas
nameserver 127.0.0.1 // IP do servidor de DNS, no caso minha própria
// máquina
```

O arquivo *named.ca* também faz parte dos arquivos de configuração do serviço DNS. Também chamado de arquivo *hints*, este apenas contém registros de servidor de nomes e de endereços. Cada registro identifica um servidor de nome para o domínio raiz (.). Este arquivo é fornecido na instalação do *Red Hat*, não necessita de modificações.

```
.           3600000 IN NS   A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000 A    198.41.0.4
.           3600000 NS   B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000 A    128.9.0.107
.           3600000 NS   C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET. 3600000 A    192.33.4.12
.           3600000 NS   D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET. 3600000 A    128.8.10.90
.           3600000 NS   E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET. 3600000 A    192.203.230.10
.           3600000 NS   F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET. 3600000 A    192.5.5.241
.           3600000 NS   G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET. 3600000 A    192.112.36.4
.           3600000 NS   H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET. 3600000 A    128.63.2.53
.           3600000 NS   I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET. 3600000 A    192.36.148.17
.           3600000 NS   J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET. 3600000 A    198.41.0.10
.           3600000 NS   K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET. 3600000 A    193.0.14.129
```

```

.           3600000   NS   L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET. 3600000   A   198.32.64.12
.           3600000   NS   M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET. 3600000   A   202.12.27.33

```

Todo servidor de nome é o *master* de seu próprio domínio *loopback*. O motivo para criar o domínio *loopback* é reduzir o tráfego da rede.

```

//Arquivo named.local
@   IN   SOA   localhost. root.localhost. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400 )    ; Minimum
    IN   NS   localhost.

1   IN   PTR   localhost.

```

O arquivo *named.conf* é o arquivo de configuração do serviço DNS.

```

//Arquivo modificado por Bruno Aragon
options {
    directory "/var/named"; // Diretório onde estará os arquivos listados
//abaixo
    forwarders {
        200.131.250.1; //Para onde será encaminhados consultas
    };
};

zone "." {
    type hint;
    file "named.ca";
};

zone "0.0.127.in-addr.arpa" {
    type master; // tipo de zona – master, slave ou hint
    file "named.local"; // nome do arquivo que contém as configurações da
//zona respectiva zona
};

```

```

zone "250.131.200.in-addr.arpa" {
    type master;
    file "fobus.reverse";
};

zone "fobus.ufla.br" {
    type master;
    file "fobus.forward";
};
zone "ufla.br" {
    type slave;
    file "uflabr.sec";
    masters {
        200.131.250.1;
    };
};

```

Agora será apresentado os arquivos de configurações das duas zonas principais, a de nome para endereços *fobus.forward* e a de endereços para nomes *fobus.reverse*. Todo endereço IP que foi substituído por nome deve constar em ambos os arquivos, como ilustrado abaixo.

```

//Arquivo fobus.forward
$ttl 38400
fobus.ufla.br. IN SOA fobus.ufla.br. root.fobus.ufla.br. (
    976379845
    10800
    3600
    432000
    38400 )
fobus.ufla.br. IN NS fobus.ufla.br. //nome da zona
fobus.ufla.br. IN A 200.131.250.47 //máquina cadastrada
aragon.fobus.ufla.br. IN A 200.131.250.49 // máquina cadastrada
www.fobus.ufla.br. IN CNAME fobus.ufla.br.
fobus.ufla.br. IN MX 5 fobus.ufla.br.

//Arquivo fobus.reverse
$ttl 38400
250.131.200.in-addr.arpa. IN SOA fobus.ufla.br. root.fobus.ufla.br.
(

```

```

          976379763
          10800
          3600
          432000
          38400 )
250.131.200.in-addr.arpa.  IN    NS    fobus.ufla.br.
47.250.131.200.in-addr.arpa.  IN    PTR   fobus.ufla.br.
49.250.131.200.in-addr.arpa.  IN    PTR   aragon.fobus.ufla.br.

```

SOA significa *Start of Authority* e marca o começo dos dados de uma zona. NS significa *Name Server* e identifica um servidor de domínio. A significa *Adresses* e mapeia um *host name* para um endereço. PTR significa *Pointer* e mapeia um endereço para um *host name*. MX significa *Mail Exchanger* e identifica um servidor de nome para um domínio. CNAME significa *Canonical Name* e define um *alias* para um *host name*.

APÊNDICE B

A seguir estará listado as configurações do serviço SMB. A maioria das configurações são *defaults*. As configurações que foram modificadas por Bruno Aragon estão marcadas.

Parâmetros globais

[global]

```
workgroup = CIN-UFLA #Modificado por Bruno Aragon
netbios name = FOBUS #Modificado por Bruno Aragon
server string = Servidor Samba #Modificado por Bruno Aragon
security = USER
encrypt passwords = Yes #Modificado por Bruno Aragon
allow trusted domains = Yes
min passwd length = 5
map to guest = Never
null passwords = No
smb passwd file = /etc/smbpasswd
passwd program = /bin/passwd
unix password sync = Yes #Modificado por Bruno Aragon
restrict anonymous = No
log level = 2
syslog = 1
syslog only = No
log file = /var/log/samba/log.%m
max log size = 50 #Modificado por Bruno Aragon – em kb
timestamp logs = Yes
debug hires timestamp = No
read raw = Yes
write raw = Yes
nt smb support = Yes
nt pipe support = Yes
nt acl support = Yes
announce as = NT
name resolve order = lmhosts host wins bcst
max packet = 65535
change notify timeout = 60
deadtime = 0
getwd cache = Yes
keepalive = 300
```

```
lpq cache time = 10
max disk size = 0
max open files = 10000
read prediction = No
read size = 16384
socket options = TCP_NODELAY SO_RCVBUF=8192
SO_SNDBUF=8192
stat cache size = 50
load printers = Yes
printcap name = /etc/printcap
printer driver file = /etc/printers.def
domain groups =
domain admin group =
domain guest group =
domain admin users = baragon #Modificado por Bruno Aragon
domain guest users =
logon path = \\%N%\%U\profile
logon drive = z: #Modificado por Bruno Aragon
logon home = \\%N%\%U
domain logons = Yes
os level = 20
preferred master = Yes #Modificado por Bruno Aragon
local master = Yes
domain master = Yes #Modificado por Bruno Aragon
browse list = Yes
dns proxy = No
wins proxy = No
wins server =
wins support = No
smbrun = /usr/bin/smbrun
lock dir = /var/lock/samba
guest account = nobody
admin users = baragon #Modificado por Bruno Aragon
create mask = 0744
force create mode = 00
security mask = -1
force security mode = -1
directory mask = 0755
force directory mode = 00
directory security mask = -1
force directory security mode = -1
guest only = No
```

```

guest ok = No
only user = No
hosts allow = 127. 200.131.250. #Modificado por Bruno Aragon
hosts deny =
max connections = 100 #Modificado por Bruno Aragon
printing = bsd
print command = lpr -r -P%p %s
lpq command = lpq -P%p
lprm command = lprm -P%p %j
printer name =
printer driver = NULL
printer driver location =
default case = lower
case sensitive = Yes
preserve case = Yes
short preserve case = Yes
browseable = Yes
share modes = Yes
fstype = NTFS
set directory = No
wide links = Yes
follow symlinks = Yes
delete readonly = No
dos filetimes = No
dos filetime resolution = No

[homes] #Compartilhamento do Home. Modificado por Bruno Aragon
comment = Directorio Home
read only = No
browseable = No

[printers] #Compartilhamento de impressoras
comment = All Printers
path = /var/spool/samba
print ok = Yes
browseable = No

[FTP] #Compartilhamento FTP. Criado por Bruno Aragon
comment = FTP Publico
path = /home/ftp/pub
guest ok = Yes

```

APÊNDICE C

Arquivo de configuração do serviço de www. Devido arquivo possuir ser bastante extenso, com muitas configurações comentadas, somente serão inseridas as configurações não comentadas. Este arquivo é bem comentado pelo seu fornecedor.

```
#Arquivo httpd.conf
#Pode ser inetd ou standalone. O primeiro somente em servidores Linux.
ServerType standalone
ServerRoot "/etc/httpd"
LockFile /var/lock/httpd.lock
PidFile /var/run/httpd.pid
ScoreBoardFile /var/run/httpd.scoreboard
#Segundos antes de receber ou enviar timeouts
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15
MinSpareServers 5
MaxSpareServers 20
StartServers 8
#Número de conexões
MaxClients 150
MaxRequestsPerChild 100
#Módulos que estão sendo carregados
# LoadModule foo_module modules/mod_foo.so
#LoadModule mmap_static_module modules/mod_mmap_static.so
LoadModule vhost_alias_module modules/mod_vhost_alias.so
LoadModule env_module modules/mod_env.so
LoadModule config_log_module modules/mod_log_config.so
LoadModule agent_log_module modules/mod_log_agent.so
LoadModule referer_log_module modules/mod_log_referer.so
#LoadModule mime_magic_module modules/mod_mime_magic.so
LoadModule mime_module modules/mod_mime.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule status_module modules/mod_status.so
LoadModule info_module modules/mod_info.so
LoadModule includes_module modules/mod_include.so
LoadModule autoindex_module modules/mod_autoindex.so
```

```

LoadModule dir_module      modules/mod_dir.so
LoadModule cgi_module      modules/mod_cgi.so
LoadModule asis_module     modules/mod_asis.so
LoadModule imap_module     modules/mod_imap.so
LoadModule action_module   modules/mod_actions.so
#LoadModule speling_module modules/mod_speling.so
LoadModule userdir_module  modules/mod_userdir.so
LoadModule alias_module    modules/mod_alias.so
LoadModule rewrite_module  modules/mod_rewrite.so
LoadModule access_module   modules/mod_access.so
LoadModule auth_module     modules/mod_auth.so
LoadModule anon_auth_module modules/mod_auth_anon.so
LoadModule db_auth_module  modules/mod_auth_db.so
LoadModule digest_module   modules/mod_digest.so
LoadModule proxy_module    modules/libproxy.so
#LoadModule cern_meta_module modules/mod_cern_meta.so
LoadModule expires_module  modules/mod_expires.so
LoadModule headers_module  modules/mod_headers.so
LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule example_module modules/mod_example.so
#LoadModule unique_id_module modules/mod_unique_id.so
LoadModule setenvif_module modules/mod_setenvif.so
#LoadModule bandwidth_module modules/mod_bandwidth.so
#LoadModule put_module     modules/mod_put.so

# Extra Modules
LoadModule perl_module     modules/libperl.so
#LoadModule php_module     modules/mod_php.so
#LoadModule php3_module    modules/libphp3.so
#E lá vai mais módulos
#AddModule mod_mmap_static.c
AddModule mod_vhost_alias.c
AddModule mod_env.c
AddModule mod_log_config.c
AddModule mod_log_agent.c
AddModule mod_log_referer.c
#AddModule mod_mime_magic.c
AddModule mod_mime.c
AddModule mod_negotiation.c
AddModule mod_status.c
AddModule mod_info.c
AddModule mod_include.c

```

```
AddModule mod_autoindex.c
AddModule mod_dir.c
AddModule mod_cgi.c
AddModule mod_asis.c
AddModule mod_imap.c
AddModule mod_actions.c
#AddModule mod_speling.c
AddModule mod_userdir.c
AddModule mod_alias.c
AddModule mod_rewrite.c
AddModule mod_access.c
AddModule mod_auth.c
AddModule mod_auth_anon.c
AddModule mod_auth_db.c
AddModule mod_digest.c
AddModule mod_proxy.c
AddModule mod_expires.c
AddModule mod_headers.c
AddModule mod_usertrack.c
AddModule mod_so.c
AddModule mod_setenvif.c

# Extra Modules
AddModule mod_perl.c
#AddModule mod_php.c
#AddModule mod_php3.c

Port 80
User nobody
Group nobody
ServerAdmin root@fobus.ufla.br
ServerName www.fobus.ufla.br
DocumentRoot "/home/httpd/html"
#Diretórios ao qual o apache tem acesso
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
<Directory "/home/httpd/html">
    Options Indexes Includes FollowSymLinks
    AllowOverride None
```

```

Order allow,deny
    Allow from all
</Directory>
#Configuração de diretórios pessoais
UserDir public_html
DirectoryIndex index.html index.htm index.shtml index.cgi
AccessFileName .htaccess
<Files ~ "\.ht">
    Order allow,deny
    Deny from all
</Files>
UseCanonicalName on
TypesConfig /etc/mime.types
DefaultType text/plain
<IfModule mod_mime_magic.c>
    MIMEMagicFile share/magic
</IfModule>
HostNameLookups off
#Configurações de log
ErrorLog /var/log/httpd/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
ServerSignature On
#Configurações de alias
Alias /icons/ /home/httpd/icons/

<Directory "/home/httpd/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
ScriptAlias /cgi-bin/ "/home/httpd/cgi-bin/"
<Directory "/home/httpd/cgi-bin">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all

```

```

</Directory>
IndexOptions FancyIndexing
#Associa cada extensão com um ícone
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core

AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^
DefaultIcon /icons/unknown.gif
ReadmeName README
HeaderName HEADER
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
AddEncoding x-compress Z
AddEncoding x-gzip gz tgz
AddLanguage en .en
AddLanguage fr .fr
AddLanguage de .de
AddLanguage da .da
AddLanguage el .el

```

```

AddLanguage it .it
LanguagePriority en fr de
<IfModule mod_php3.c>
    AddType application/x-httpd-php3 .php3
    AddType application/x-httpd-php3-source .phps
</IfModule>
<IfModule mod_php.c>
    AddType application/x-httpd-php .phtml
</IfModule>

AddType application/x-tar .tgz
AddType text/html .shtml
AddHandler server-parsed .shtml
AddHandler imap-file map
#Reconhecimento de Browser
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4.0" force-response-1.0
BrowserMatch "Java/1.0" force-response-1.0
BrowserMatch "JDK/1.0" force-response-1.0
<IfModule mod_perl.c>
Alias /perl/ /home/httpd/perl/
    <Location /perl>
        SetHandler perl-script
        PerlHandler Apache::Registry
        Options +ExecCGI
    </Location>
</IfModule>
#Criado por Bruno Aragon
Alias /ftp /home/ftp/pub
<Location /ftp>
    order deny,allow
    allow from all
    Options Indexes FollowSymLinks
</Location>
Alias /doc/ /usr/doc/
<Location /doc>
    order deny,allow
    allow from all
</Location>
#End of file

```

APÊNDICE D

Arquivo de configuração do serviço FTP. O mais simples de configurar.

```
class all real,guest,anonymous *
email root@fobus.ufla.br
loginfails 5
readme README* login
readme README* cwd=*
message /welcome.msg login
message .message cwd=*
compress yes guest,real,anonymous
tar yes guest,real,anonymous
chmod no guest,anonymous
delete no guest,anonymous
overwrite no guest,anonymous
rename no guest,anonymous
log transfers real,anonymousinbound,outbound
shutdown /etc/shutmsg
passwd-check rfc822 warn
greeting full
chmod yes real
delete yes real
overwrite yes real
rename yes real
anonymous /home/ftp all
```

APÊNDICE E

Arquivo de configuração do Majordomo.

```
#Arquivo majordomo.cf
$whereami = "fobus.ufla.br";
$whoami = "Majordomo\@$whereami";
$whoami_owner = "Majordomo-Owner\@$whereami";
#Diretório Home do Majordomo
if ( defined $ENV{"HOME"} ) {
    $homedir = $ENV{"HOME"};
} else {
    $homedir = "/usr/lib/majordomo";
}
#Onde estão os arquivos de configuração de cada lista criada
$listdir = "/var/lib/majordomo/lists";
$digest_work_dir = "/var/lib/majordomo/digest";
$log = "/var/lib/majordomo/log";
$sendmail_command = "/usr/sbin/sendmail";
$mailer = "$sendmail_command -oi -oee -f$sender";
$bounce_mailer = "$sendmail_command -oi -oee -f$sender -t";
$max_loadavg = 10;
($avg_1_minute, $avg_5_minutes, $avg_15_minutes) =
    $uptime =~ /average:\s+(\S+),\s+(\S+),\s+(\S+)/;
exit 75 if ($avg_15_minutes >= $max_loadavg);
$config'default_subscribe_policy = "open+confirm";
$no_x400at = 0;
$no_true_x400 = 0;
$filedir = "$listdir";
$filedir_suffix = ".archive";
$index_command = "/bin/ls -IRL";
$return_subject = 1;
$majordomo_request = 0;
$max_which_hits = 0;
umask(007);
$config_umask = 007;
die "Perl version $] too old\n" if ($] < 4.019);
```

```

@safedirs = ();
$TMPDIR = "/var/lib/majordomo/tmp";
$MAX_HEADER_LINE_LENGTH = 128;
$MAX_TOTAL_HEADER_LENGTH = 1024;
#Configuração de subjects
$admin_headers = <<'END';
/^subject:\s*subscribe\b/i
/^subject:\s*unsubscribe\b/i
/^subject:\s*uns\w*b/i
/^subject:\s*.*un-sub/i
/^subject:\s*help\b/i
/^subject:\s*.*\bchange\b.*\baddress\b/i
/^subject:\s*request\b(.*\b)?addition\b/i
/^subject:\s*cancel\b/i
END
$admin_body = <<'END';
^bcancel\b/i
^badd me\b/i
^bdelete me\b/i
^bremove\s+me\b/i
^bchange\b.*\baddress\b/
^bsubscribe\b/i
/^sub\b/i
^bunsubscribe\b/i
/^unsub\b/i
^buns\w*b/i
/^\s*help\s*$/i
/^\s*info\s*$/i
/^\s*info\s+\S+\s*$/i
/^\s*lists\s*$/i
/^\s*which\s*$/i
/^\s*which\s+\S+\s*$/i
/^\s*index\s*$/i
/^\s*index\s+\S+\s*$/i
/^\s*who\s*$/i
/^\s*who\s+\S+\s*$/i
/^\s*get\s+\S+\s*$/i
/^\s*get\s+\S+\s+\S+\s*$/i

```

```

/^s*approve\b/i
/^s*passwd\b/i
/^s*newinfo\b/i
/^s*config\b/i
/^s*newconfig\b/i
/^s*writeconfig\b/i
/^s*mkdigest\b/i
END
$global_taboo_headers = <<'END';
/^subject: ndn: /i
/^subject:\s*RCPT:/i
/^subject:\s*Delivery Confirmation\b/i
/^subject:\s*NON-DELIVERY of:/i
/^subject:\s*Undeliverable Message\b/i
/^subject:\s*Receipt Confirmation\b/i
/^subject:\s*Failed mail\b/i
/^subject:\s*Returned mail\b/i
/^subject:\s*unable to deliver mail\b/i
/^subject:\s.*\baway from my mail\b/i
/^subject:\s*Autoreply/i
END
$global_taboo_body = <<'END';
END
$majordomo_dont_reply = '(mailer-
daemon|uucp|listserv|majordomo|listproc)\@';
1;

```

Foi criada uma lista chamada *aturma*. Portanto deve existir três arquivos para o funcionamento da mesma. O primeiro arquivo contém os membros da lista, este chama-se *aturma*.

```

diniz@comp.ufla.br
tony@comp.ufla.br
awdrey@comp.ufla.br
baragon@comp.ufla.br
carla@comp.ufla.br
cjdd@comp.ufla.br
castro@comp.ufla.br

```

daciomf@comp.ufla.br
tavares@comp.ufla.br
kanazawa@comp.ufla.br
erasmo@comp.ufla.br
dutra@comp.ufla.br
larochel@comp.ufla.br
camargos@comp.ufla.br
leopreto@comp.ufla.br
lbueno@comp.ufla.br
rsgomes@comp.ufla.br
rose@comp.ufla.br
vakino@comp.ufla.br
wtomaz@comp.ufla.br
popowls@comp.ufla.br
olinda@comp.ufla.br
massahud@comp.ufla.br

O segundo é o arquivo com as configurações da lista, chamado *aturma.config*.

```
#Arquivo aturma.config
admin_passwd = XXXXXX
administrivia = yes
announcements = yes
approve_passwd = XXXXX
date_info = yes
date_intro = yes
debug = no
description = Lista da Nossa Turma
digest_name = aturma
get_access = list
index_access = list
info_access = open
intro_access = list
maxlength = 1048576
message_footer << END
ODOIÁ, IEMANJÁ, SARAVÁ...
END
message_fronter << END
E AÍ, GALERINHA DO MAL...
END
```

moderate = no
mungedomain = no
precedence = bulk
purge_received = no
restrict_post = aturma
sender = owner-aturma
strip = yes
subscribe_policy = closed+confirm
unsubscribe_policy = open
welcome = yes
which_access = list
who_access = list

O terceiro arquivo contém algumas informações da lista e chama-se *aturma.info*.

[Last updated on: Wed Jun 6 8:34:36 2001]