

Proposta de um Modelo de Ferramenta para auxílio à Administração Computacional

Este exemplar corresponde à redação final da
Dissertação devidamente corrigida e defendida
por Robson Gomes de Melo e aprovada pela
Banca Examinadora.

Campinas, 10 de fevereiro de 2010.



Paulo Lício de Geus
Instituto de Computação UNICAMP -
(Orientador)

Dissertação apresentada ao Instituto de Com-
putação, UNICAMP, como requisito parcial para
a obtenção do título de Mestre em Ciência da
Computação.



**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Maria Fabiana Bezerra Müller – CRB8 / 6162

Melo, Robson Gomes de
M491p Proposta de um modelo de ferramenta para auxílio à administração
computacional/Robson Gomes de Melo-- Campinas, [S.P. : s.n.], 2009.

Orientador : Paulo Lício de Geus.

Dissertação (mestrado) - Universidade Estadual de Campinas,
Instituto de Computação.

I.Redes de computadores - Administração. I. Geus, Paulo Lício
de. II. Universidade Estadual de Campinas. Instituto de Computação.
III. Título.

Título em inglês: Proposal of a tool model to support computing administration

Palavras-chave em inglês (Keywords): 1.Computer network – Administration.

Área de concentração: Sistemas de Computação

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof. Dr. Paulo Lício de Geus (IC-UNICAMP)
Prof. Dr. Edmundo Roberto Mauro Madeira (IC - UNICAMP)
Prof. Dr. Aldri Luiz dos Santos (UFPR)

Data da defesa: 21/12/2009

Programa de Pós-Graduação: Mestrado em Ciência da Computação

TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 21 de dezembro de 2009, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. Aldri Luiz dos Santos
Departamento de Informática / UFPR



Prof. Dr. Edmundo Roberto Mauro Madeira
IC / UNICAMP



Prof. Dr. Paulo Lício de Geus
IC / UNICAMP

Proposta de um Modelo de Ferramenta para auxílio à Administração Computacional

Robson Gomes de Melo

10 de fevereiro de 2010

Banca Examinadora:

- Paulo Lício de Geus
Instituto de Computação UNICAMP - (Orientador)
- Edmundo Roberto Mauro Madeira
Instituto de Computação UNICAMP
- Aldri Luiz dos Santos
Departamento de Informática UFPR
- Christiane Marie Schweitzer
Centro de Matemática, Computação e Cognição UFABC - (Suplente)
- Célio Cardoso Guimarães
Instituto de Computação UNICAMP - (Suplente)

© Robson Gomes de Melo, 2010.
Todos os direitos reservados.

Resumo

Devido ao grande impacto gerado por sistemas e redes de computadores na vida moderna, as atividades dos administradores computacionais têm se tornado fundamentais às organizações. Entretanto, as técnicas e ferramentas utilizadas por esses profissionais não conseguiram acompanhar a evolução de uso do poder computacional.

Na tentativa de auxiliar os administradores, várias ferramentas foram desenvolvidas com o intuito de possibilitar uma maneira mais simples, rápida e eficiente de configurar e administrar um parque tecnológico. Porém, muitas delas ainda apresentam impossibilidade, principalmente pela necessidade da presença física do administrador no ambiente que se administre. Deste modo, considera-se que as ferramentas de administração não atendem às necessidades julgadas essenciais aos cenários atuais.

Neste contexto, esse trabalho apresenta um estudo das consideradas principais e mais populares ferramentas de administração computacional, analisando-as e comparando-as. Propõe também um novo modelo de ferramenta de administração, que busca solucionar as limitações encontradas nas ferramentas estudadas, como apontar novas possibilidades a esta área.

Abstract

Due to the huge impact caused by systems and computer networks nowadays, computer administrator's activities have become fundamental to their organizations. On the other hand, the techniques and tools used by these professionals didn't improve as much as the computing power of the machines we have available now.

In order to assist computer administrators, several tools were developed to allow a simpler, faster and more efficient way to configure and manage several machines. However, many of these solutions are still unable to manage and configure their systems and networks without the physical presence of the System Administrator. Therefore, these tools are not of satisfying the requirements considered essentials to the current scenarios.

In this context, this work shows a comparative analysis of the main and most popular computer management tools available. It also presents a new approach to remote administration, with the purpose of solving the limitations found in the analysed tools, and indicates the new tendencies and possibilities in this field.

Agradecimentos

Eu gostaria de agradecer primeiramente a Deus, por ter me concedido essa oportunidade e dado força para agarrá-la, a minha família por ter me apoiado nessa jornada de muita dedicação e esforço, a Daiany pela compreensão das circunstâncias nesta etapa de nossas vidas, a meu ilustre orientador prof. Paulo Lício de Geus pela aposta que fez em mim, a meus colegas de laboratório que contribuíram muito com minha formação, a meus colegas de república que ao longo desses anos se tornaram uma nova família e a meus colegas da Universidade do Estado de Mato Grosso em especial o departamento de Computação do *campus* universitário Jane Vanini em Cáceres-MT, por ter compreendido e colaborado comigo nessa etapa de minha carreira acadêmica.

Sumário

Resumo	ix
Abstract	xi
Agradecimentos	xiii
1 Introdução	1
2 Administração Computacional	5
2.1 Administração de Sistemas Computacionais	5
2.2 Administração de Redes de Computadores	8
2.3 Administração Computacional	12
2.4 Conclusão	17
3 Ferramentas de Administração Computacional	19
3.1 Classificação das Ferramentas de Administração	19
3.1.1 Ferramentas Caseiras de Administração Baseadas em <i>Script</i>	20
3.1.2 Ferramentas de Administração Baseadas em Interface Modo Texto .	25
3.1.3 Ferramentas de Administração Baseadas em Interface Gráfica . . .	33
3.2 Comparação entre as Ferramentas de Administração Computacional . . .	68
3.3 Conclusão	70
4 Proposta de um Modelo de Ferramenta	73
4.1 O Modelo de Ferramenta	73
4.2 Arquitetura do Modelo	77
4.3 Funcionamento do Modelo	80
4.4 Aspectos de Segurança	81
4.4.1 O Modelo Proposto e os Aspectos de Segurança	84
4.4.2 Segurança Envolvida no Modelo	84
4.4.3 Segurança Proporcionada pelo Modelo	86

4.5	O Protótipo	88
4.5.1	Detalhes de Implementação	88
4.5.2	Organização	96
4.6	Conclusão	103
5	Testes e Resultados	105
5.1	Ambiente de Testes	105
5.2	Serviço Escolhido	106
5.2.1	<i>O Samba</i>	107
5.2.2	A Configuração Desejada	107
5.3	Os Testes	107
5.3.1	Testes de Usabilidade	107
5.3.2	Testes de Funcionalidades	111
5.3.3	Análise de Aspectos Básicos de Segurança	114
5.4	Comparação entre o Modelo Proposto e as Ferramentas de Administração Analisadas	116
5.5	Conclusão	118
6	Considerações Finais	119
6.1	Trabalhos Futuros	122
	Bibliografia	124
A	Arquivos de Configuração Processados pela Ferramenta	129

Lista de Tabelas

2.1	Protocolos Portas e Serviços	11
2.2	Funções Exercidas pelos Administradores em 2001 - <i>Fonte: SAGE</i> [1]	13
2.3	Funções Exercidas pelos Administradores em 2002 - <i>Fonte: SAGE</i> [2]	13
3.1	Características de Funcionamento das Ferramentas em Modo Texto	31
3.2	Propriedades Básicas de Ferramentas de Modo Texto	32
3.3	Características do YaST e Linuxconf	40
3.4	Módulos de Alguns dos Serviços do IBM Tivoli	49
3.5	Ferramentas com Foco em Gerência de <i>e-business</i>	51
3.6	Funções dos Módulos do RHN/Spacewalk	53
3.7	Vantagens do <i>Webmin</i>	65
3.8	Desvantagens do <i>Webmin</i>	66
3.9	Características das Ferramentas de Acesso Remoto	67
3.10	Características das Principais Ferramentas de Administração	69
4.1	Requisitos Atendidos ou não pelas Ferramentas de Administração	76
4.2	Representação tTabular das Funções de Transição dos Estados do Autômato	93
5.1	Comparação entre o Modelo Proposto e as Ferramentas Analisadas	116
5.2	Comparação entre o Modelo Proposto e as Ferramentas Analisadas (cont.)	117

Lista de Figuras

2.1	Funcionalidade do Sistema Operacional	7
2.2	Pilha de Protocolos e Protocolo <i>TCP/IP</i>	9
2.3	Funções Exercidas pelos Administradores em 2003 - <i>Fonte: SAGE</i> [3]	14
2.4	Funções Exercidas pelos Administradores em 2004/2005 - <i>Fonte: SAGE</i> [4]	14
2.5	Funções Exercidas pelos Administradores em 2005/2006 - <i>Fonte: SAGE</i> [5]	15
3.1	System-Config-Network	34
3.2	System-Config-Http	34
3.3	Módulos do Linuxconf	36
3.4	Interface do Linuxconf	36
3.5	YaST Control Center	37
3.6	Seções do YaST	38
3.7	Arquitetura do YaST	38
3.8	<i>Remote Desktop</i> - Fonte:[6]	41
3.9	<i>Remote Desktop (cont.)</i> - Fonte:[6]	42
3.10	Funcionamento sob Arquitetura Cliente/Servidor	44
3.11	Interface do SWAT	45
3.12	Interface de Formulários do SWAT	46
3.13	Interface do Tivoli	50
3.14	Interface do Red Hat Network	54
3.15	Interface do Spacewalk	54
3.16	Arquitetura do Red Hat Network-Spacewalk	55
3.17	Funcionamento do RHN - Spacewalk no modelo Hosted	57
3.18	Funcionamento do RHN - Spacewalk no Modelo Satélite	57
3.19	Funcionamento do RHN - Spacewalk no modelo Satélite com Proxy	58
3.20	Interface do <i>Webmin</i>	59
3.21	Formulário de Configuração do Servidor WEB pelo <i>Webmin</i>	63
3.22	Esquema de Funcionamento do <i>Webmin</i>	63
4.1	Arquitetura do Modelo Proposto	77

4.2	Funcionamento do Modelo Proposto	80
4.3	Segurança Envolvida entre o Administrador e a Ferramenta	85
4.4	Segurança Envolvida entre a Ferramenta e a Máquina a ser Configurada	86
4.5	Utilização da Ferramenta para Configuração de Segurança	87
4.6	Autômato que Representa o Filtro dos Arquivos de Configuração	94
4.7	Estrutura Organizacional do Protótipo	97
4.8	Arquivo smb.conf Original	98
4.9	Arquivo smb.conf Limpo	99
4.10	Arquivo smb.conf em XML	100
4.11	Parâmetros da Sessão <i>global</i>	101
4.12	Parâmetros da Sessão <i>printers\$</i>	101
4.13	Parâmetros da Sessão <i>print\$</i>	101
4.14	Parâmetros de uma <i>Nova</i> Sessão	101
4.15	Autenticação na Ferramenta	102
4.16	Alternativas de Configuração	102
4.17	Painel de Serviços Configuráveis	102
4.18	Nível Sobre o Serviço Desejado	102
5.1	Configuração: Passo 1	108
5.2	Configuração: Passo 2	108
5.3	Configuração: Passo 3	108
5.4	Configuração: Passo 4	108
5.5	Configuração: Passo 5	109
5.6	Configuração: Passo 6	109
5.7	Configuração: Passo 7	109
5.8	Configuração: Passo 8	109
5.9	Configuração: Passo 9	110
5.10	Configuração: Passo 10	110
5.11	Captura de Trafego dos Pacotes Entre Administrador e Central	111
5.12	Captura de Trafego dos Pacotes Entre a Central e a Máquina Servidora	112
5.13	Acesso pelo <i>Mac Os X</i>	113
5.14	Acesso pelo <i>Windows XP</i>	113
5.15	Acesso pelo <i>Ubuntu 9.04</i>	113
5.16	<i>Ubuntu</i> Criação de Diretório	113

Capítulo 1

Introdução

A partir do grande crescimento da Internet, as redes de computadores como as *LAN's (Local Area Network)* e as *WAN's (Wide Area Network)*[7] passam por constantes evoluções. Os serviços disponibilizados por essa conectividade vêm sendo aprimorados a cada dia para poder garantir as funcionalidades necessárias ao bom desempenho das organizações que as utilizam.

Atualmente, os sistemas e redes de computadores, assim como seus recursos associados, têm se tornado fundamentais para o bom desempenho e sobrevivência das organizações no mundo globalizado e de uma sociedade da informação, a ponto de serem consideradas indispensáveis em tempo integral [8]. Isto significa que as tarefas de Administração, Gerência e Monitoração dessas redes e sistemas computacionais, também tem ganhado um papel significativo neste novo cenário. Entretanto, essas tarefas se tornam cada vez mais complicadas e exigem uma mão de obra qualificada e especializada, em função da complexidade imposta por tais atividades.

As redes e sistemas computacionais de hoje oferecem serviços dos mais variados tipos, para poderem atender às necessidades de uma organização. Conseqüentemente, esses serviços possuem uma variada escala de complexidade para sua implantação e manutenção, o que dificulta as atividades dos administradores, que passam a ter novas atribuições.

O administrador computacional tem como principal objetivo, proporcionar um ambiente tecnológico, estável e seguro, para que as pessoas de uma organização possam realizar suas tarefas sem se preocupar com sistemas e os meios de comunicação entre eles. Em uma linguagem mais técnica, pode-se dizer que o administrador computacional deve proporcionar um “Ambiente de Produção” funcional e com todos os serviços transparentes aos usuários deste sistema. Entretanto, a evolução dos mecanismos e ferramentas utilizadas pelos administradores para atenderem essa nova demanda não acompanham a evolução das novas tecnologias almejadas pelas organizações. Neste sentido, este trabalho

propõe um novo modelo de ferramenta de administração computacional, para auxiliar o administrador em suas tarefas habituais.

É fato, que inúmeros programas vêm sendo desenvolvidos pela comunidade e até mesmo por iniciativas particulares afim de se apresentarem como melhores alternativas de ferramentas de administração. No entanto, grande maioria desses softwares se preocupam com a configuração/administração de serviços/dispositivos específicos, não estando atentos para um conjunto amplo e variado deles.

Por outro lado, um estudo realizado entre algumas das ferramentas de administração computacional mais populares entre os administradores, a qual será analisada nos capítulos futuros deste trabalho, aponta que ainda é muito presente a utilização de ferramentas que foram desenvolvidas a mais de quinze (15) anos atrás, quando os cenários e as necessidades eram outras. O que permite concluir que a linha de evolução deste tipo de software é tênue e não acompanha a evolução dos novos serviços/dispositivos computacionais, utilizados pelas organizações para automatização de determinados processos e que aumentam exponencialmente.

Do ponto de vista da segurança não é muito diferente, como esta também é uma atividade desempenhada por administradores computacionais, não fica fora do contexto apresentado. As ferramentas que proporcionam a configuração de mecanismos de segurança também têm as mesmas deficiências encontradas nas ferramentas de configuração de serviços/dispositivos de sistema, sendo individualizadas e utilizadas em cenários específicos.

Deste modo, os objetivos específicos deste trabalho são estudar as ferramenta de administração de sistemas e redes de computadores, compará-las, avaliá-las e apresentar uma proposta de um novo modelo de ferramenta de administração, baseada em uma interface gráfica, que possa ser acessada remotamente através de um navegador de internet, capaz de realizar configurações de um conjunto variado de serviços/dispositivos computacionais em máquinas também remotas, onde a presença física do administrador no ambiente que se configura seja inviável ou até mesmo impossibilitado de acontecer, além de dispor de um repositório de arquivos de configurações para possíveis restaurações de configurações do sistema caso seja necessário, ou a propagação em massa de uma mesma configuração ou configurações específicas para ambientes remotos.

Organização

Para conseguir alcançar os propósitos apresentados, o restante desta dissertação está organizada da seguinte maneira:

Capítulo 2: aborda conceitos de administração de sistemas e administração de redes de computadores, propondo a utilização do termo Administração Computacional para se referir as atividades comuns a essas áreas, que em organizações de pequeno e médio porte

são realizadas na maioria da vezes pela mesma pessoa ou equipe.

Capítulo 3: apresenta um estudo realizado sobre algumas das mais utilizadas e populares ferramentas de administração computacional, como o Cfengine, Slack, Puppet, Linuxconf, YaST, IBM Tivoli, Webmin, entre outras, sendo essas as principais alternativas para os administradores na tentativa de auxiliá-los em suas atividades habituais. Para tanto, o capítulo apresenta uma classificação das ferramentas, as quais se dividem em Ferramentas Caseiras de Administração, desenvolvidas pelos próprios administradores para ajudá-los em seu cotidiano, Ferramentas de Administração Sem Interface Gráfica, apenas baseadas na utilização de terminais não gráficos e Ferramentas de Administração com Interface Gráfica, essa categoria subdividida em Ferramentas de Acesso Local e Ferramentas de Administração de Acesso Remoto.

Capítulo 4: apresenta a proposta de um novo modelo de ferramenta de administração computacional. Neste capítulo são descritos os aspectos relacionados à arquitetura modular com a descrição de cada módulo, o funcionamento desse modelo, bem como características julgadas de fundamental importância à ferramentas pertinentes aos cenários atuais, como a configuração e administração de um amplo conjunto de serviços ou dispositivos que tenham seu funcionamento baseado em arquivos de configuração no formato texto, flexibilidade para a construção e inserção de novos módulos responsáveis pela configuração de serviços ou dispositivos específicos, interface gráfica de acesso remoto via WEB, realização da configuração remota de serviços e dispositivos sem a necessidade de instalação de programas agentes ou aplicações clientes nas máquinas a serem configuradas remotamente, instalação remota de pacotes de softwares, utilização de um repositório de arquivos de configuração já editados pelo usuário para possível restauração de configurações ou mesmo a distribuição em massa ou individualizada para máquinas da rede, o armazenamento do estado da sessão do administrador para configurações futuras ou interrompidas por motivos externos, etc. O capítulo também trata dos aspectos relacionados a segurança, onde tem-se uma discussão sobre dois sentidos: *i)* a segurança incorporada no modelo de ferramenta proposta, *ii)* a segurança que pode ser proporcionada pelo uso do modelo de ferramenta.

Na sequência é apresentado um protótipo criado para a validação do modelo proposto, bem como os detalhes de sua implementação.

Capítulo 5: esse capítulo tem por objetivo apresentar o ambiente de testes realizados sobre o protótipo da ferramenta de administração computacional, bem como os resultados alcançados. Também é dado destaque para uma breve comparação entre as ferramentas de administração estudadas e o modelo de ferramenta proposto neste trabalho.

Capítulo 6: apresenta as considerações finais, o relato de algumas dificuldades encontradas e a sugestão de trabalhos futuros.

Capítulo 2

Administração Computacional

Neste capítulo são apresentados conceitos relacionados à “Administração Computacional”. O objetivo é ilustrar a relevância e a utilização do termo como uma alternativa às atividades referentes à administração. Para tanto, o capítulo apresenta uma análise sobre o foco da administração de sistemas, sintetizando conceitos de sistemas operacionais e administração de redes, com fundamentos de redes de computadores, apontando essas como as duas atividades imprescindíveis à profissão. O capítulo segue com a apresentação de atividades habituais à administração de sistemas e administração de redes, além de uma discussão sobre os resultados apresentados por pesquisas realizadas nesta área, justificando a utilização do termo Administração Computacional para se referir ao conjunto de atividades pertinentes aos cenários atuais. Também é colocado em destaque a complexidade envolvida na administração computacional, principalmente quando os sistemas estão geograficamente dispersos, situação em que a presença física do administrador nos vários locais que se administre se torna inviável.

2.1 Administração de Sistemas Computacionais

Quando se fala de administração de sistemas computacionais e administração de redes de computadores, nem sempre se atribui os devidos valores a essas atividades. Para tentar entender essas áreas fundamentais às organizações que pretendem se firmar em um mundo globalizado e da informação, é apresentado uma análise dicotômica sobre o que se refere cada termo, iniciando pela administração de sistemas computacionais.

Segundo Drucker [9], administrar é manter a organização coesa, fazendo-a funcionar. Ao tentar estabelecer uma analogia desta definição, pode-se dizer que para administrar um computador ou um conjunto deles, um administrador é responsável por instalar e manter os sistemas computacionais coesos e funcionando. Também, pode-se acrescentar o fato de que o mesmo deve saber operá-los com muita destreza, objetivando maior eficiência dos

recursos que esses sistemas podem lhe oferecer.

Define-se como sistemas, um conjunto de elementos interdependentes em interação, que visam atingir um objetivo comum [10]. Nesse contexto, entende-se por sistemas computacionais um conjunto de dispositivos eletrônicos (*hardware*) capaz de processar uma informação de acordo com uma sequência de instruções lógica (*software*). Esses por sua vez podem ser desde simples aplicativos até os sistemas operacionais.

Sistemas Operacionais

A administração de sistemas computacionais exige conhecimentos aprofundados relativos aos sistemas operacionais, dado que é função do administrador de sistemas instalar, configurar e manter um ambiente de produção altamente estável e seguro para seus usuários, tarefa essa que não é simples, principalmente devido ao surgimento de novos sistemas operacionais (S.O.) e as novas versões de sistemas já consolidados no mercado e na academia.

Os sistemas operacionais vêm passando por um processo intenso de transformação, devido ao aprimoramento de sistemas que antes eram considerados inadequado e sem suporte, como é o caso das inúmeras distribuições de sistemas *Linux*, que vêm ganhado cada vez mais espaços nas organizações devido a sua robustez, segurança, baixo ou nenhum custo com licenças, etc. Neste contexto, outros sistemas operacionais baseados em *UNIX* começam a ressurgir com potencial para ocupar lugar de destaque no *ranking* dos sistemas operacionais mais utilizados e se despertam para o mercado comercial, como é o caso dos sistemas da *SUN* o *Solaris* e o *FreeBSD* originário do *BSD* da Universidade de Berkeley nos Estados Unidos. Assim, também se enquadram os *MacOS* e *MacOS X* da *Apple* que eram considerados sistemas inacessíveis financeiramente mas que agora já podem ser adquirido por muitos, em função de sua popularização e pela diminuição de preço de mercado.

Além do aparecimento desses sistemas considerados “novos”, tem-se os sistemas operacionais da família *Windows* da *Microsoft*, que tem estado entre os mais vendidos no mercado de sistemas operacionais, entretanto, vêm perdendo espaço em função do surgimento desses outros sistemas supracitados.

Cenários com inúmeros tipos de sistemas operacionais são muito presente atualmente, isso causa uma heterogeneidade de sistemas nas organizações, dificultando as atividades do administrador de sistemas. Desta forma, é de fundamental importância o conhecimento preciso dos administradores sobre os princípios e fundamentos de sistemas operacionais.

Segundo Silberschartz, Stalling e Tanenbaum [11, 12, 13], existem dois modos distintos de conceituar um sistema operacional:

1. Pela perspectiva *top-down* de cima para baixo: é uma abstração do hardware, fa-

zendo o papel de intermediário entre o aplicativo (*software*) e os componentes físicos do computador (*hardware*);

2. Numa visão *bottom-up*, de baixo para cima: é um gerenciador de recursos, isto é, controla quais aplicações (*processos*) podem ser executadas, quando e que recursos (*memória, disco, processador, etc.*) podem ser utilizados.

Pode-se ilustrar a funcionalidade de um sistema operacional através da Figura 2.1 :

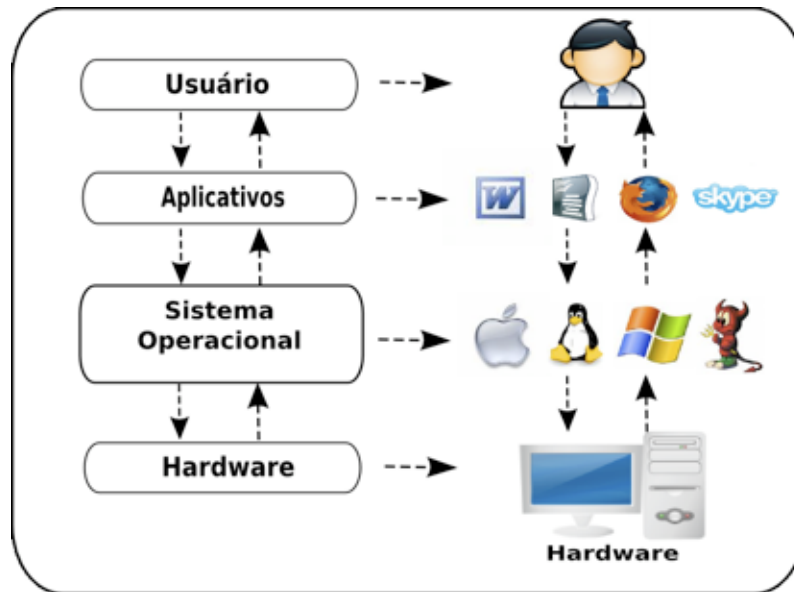


Figura 2.1: Funcionalidade do Sistema Operacional

Na ilustração apresentada pela Figura 2.1, o usuário representa as pessoas em interação com o computador por intermédio de programas aplicativos, responsáveis pelas execuções de atividades específicas, graças a abstração do *hardware* proporcionada pelo sistema operacional.

Além dos sistemas operacionais, o administrador de sistema deve estar atento ao surgimento de novas aplicações que apareçam no mercado para automatizar determinadas tarefas, assim, deve também analisar os impactos prós e contras da implantação dessas aplicações no seu ambiente de trabalho.

Para Burgess [14], a atividade de um administrador de sistemas é de natureza extremamente prática e requer um alto grau de responsabilidade, pois em suas mãos está a incumbência de manter todos os sistemas computacionais de uma organização em bom funcionamento. Entretanto, além de ser uma tarefa de teor altamente prático, um administrador deve ser um autodidata que busque sempre estar atualizado, principalmente

porque essa área se caracteriza por constantes mudanças decorrentes do desenvolvimento e aperfeiçoamento dos sistemas operacionais.

Para manter um sistema computacional configurado e funcionando corretamente, é preciso um profissional qualificado, que domine senão todos mas a maioria dos recursos e segredos de um sistema operacional, em certas situações é necessário ser criativo e conhecer bem o funcionamento dos computadores para resolver os problemas em tempo hábil e da melhor maneira possível. Isso requer planejamento, organização e metodologia para obter soluções robustas e que funcionem, desta forma se faz necessário um administrador bem preparado, o que torna cada vez mais escassa a existência destes perfil de profissional.

2.2 Administração de Redes de Computadores

Segundo Costa [15], formalmente pode-se definir administração de redes como a área da computação ligada à manutenção das funcionalidades esperada das redes de computadores. Entende-se por redes de computadores neste contexto, estruturas físicas e lógicas utilizadas para interligar computadores e equipamentos computacionais em diferentes posições geográficas.

De acordo com Tanenbaum [16], o propósito de uma rede de computadores é a possibilidade de compartilhamento de recursos e dados entre mais de um equipamento. Com este objetivo é possível transmitir informações de um ponto a outro. Isso proporciona dinamizar negócios, interagir pessoas, entreter, informar e outras possibilidades de troca de informação. Além, do compartilhamento de recursos físicos como impressoras, scanners, discos e até computadores inteiros o que passa a ser algo precioso para as organizações e pessoas, considerando-se a economia gerada.

Por proporcionarem inúmeras vantagens, as redes de computadores estão se tornando cada vez mais presente na organizações dos mais variados portes. No entanto, juntamente com as vantagens vêm também as dificuldades, principalmente para os administradores de redes. Como já mencionado, os sistemas operacionais estão se desenvolvendo e são eles os principais agentes de uma rede, também sobre os quais são implementados servidores para serviços específicos, tentando atender a uma determinada demanda cada vez maior, essa tarefa no entanto é realizada pelos administradores de redes.

Não bastando a dificuldade de lidar com a heterogeneidade de sistemas operacionais nas redes, os administradores ainda têm de conviver com a gerência dos equipamentos que dizem respeito a conectividade como os *switches*, *hubs* e *roteadores*, para citar alguns exemplos. Existem inúmeros agravantes que podem afetar a construção, operação e administração de uma rede. Problemas, desafios e emergências podem acontecer em vários setores, como nos equipamentos de conectividades, meios de transmissão, computadores e nos próprios programas executados em redes. Neste sentido é de suma importância que

um administrador conheça profundamente o funcionamento e estrutura de uma rede de computadores.

Redes de Computadores

De acordo com Kurose [7], as redes de computadores podem ser classificadas quanto à sua dimensão e posição geográfica dos seus elementos. As redes de acesso local denominadas *LAN's (Local Area Network)*, são redes em que os computadores estão fisicamente próximos em uma área limitada. As redes de acesso metropolitano são denominadas *MAN's (Metropolitan Area Network)*, por que são formadas por equipamentos que abrangem áreas semelhantes a uma cidade. As *WAN's (Wide Area Networks)* são redes de áreas afastadas que abrangem grandes dimensões geográficas como a rede mundial.

Além de seus elementos físicos e suas dimensões, uma rede também é composta por seus elementos lógicos, como os protocolos de comunicação. O protocolo de comunicação mais difundido é o protocolo *TCP/IP (Transmission Control Protocol/Internet Protocol)*[7], amplamente utilizado na internet e constituído de uma arquitetura em camadas, conforme ilustra a Figura2.2

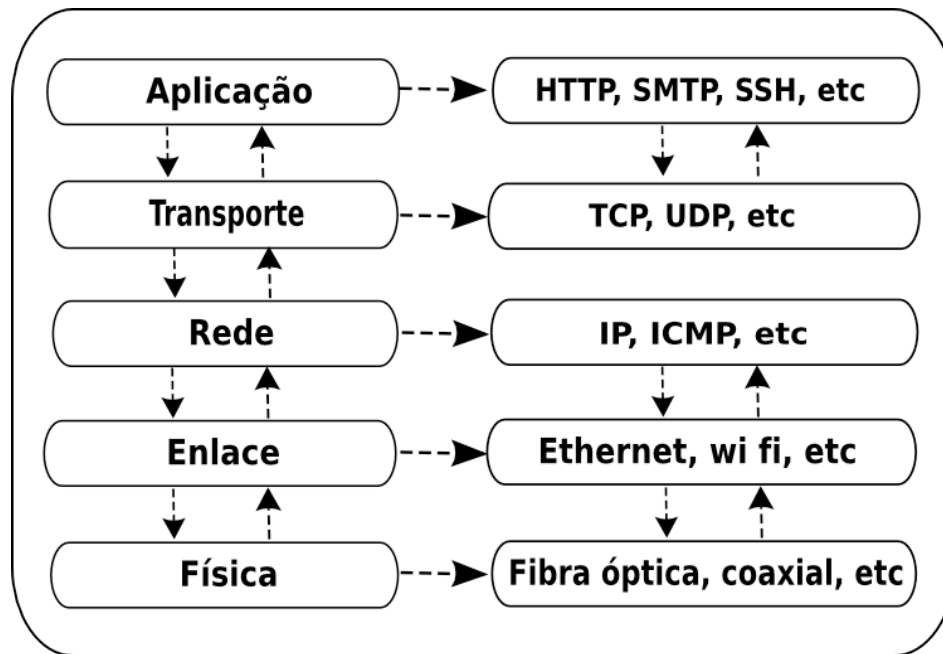


Figura 2.2: Pilha de Protocolos e Protocolo *TCP/IP*

A Figura 2.2, ilustra um pilha de protocolos e o protocolo *TCP/IP* com todas as suas camadas e respectivas tecnologias subsequentes. Cabe ao administrador de redes o bom entendimento deste protocolo assim como a funcionalidade de cada uma de suas camadas

e conseqüentemente suas principais tecnologias, isso em razão de sua ampla utilização e ser o principal protocolo de comunicação da internet.

De forma sucinta e baseado em Kurose [7], cada camada do protocolo *TCP/IP* pode ser definida da seguinte maneira:

- *Aplicação* : É a denominação dada à camada localizada no topo da pilha do protocolo *TCP/IP*, tem por objetivo realizar a troca de mensagens entre programas aplicativos em sistemas computacionais geograficamente separados ou localizado em mesma máquina;
- *Transporte*: É a camada do protocolo responsável pela comunicação lógica entre processos de aplicação em diferentes hospedeiros, tem por objetivo quebrar as mensagens da camada aplicação em segmentos e enviar para a camada de rede;
- *Rede* : É a camada responsável pela comunicação lógica entre hospedeiros, tem por objetivo encapsular os segmentos vindos da camada de transporte em datagramas, é composta por um protocolo de identificação *IP* e um protocolo de roteamento;
- *Enlace* : É a camada do protocolo que conecta hospedeiros por intermédio de um canal de comunicação, os objetivos são providenciar a transmissão dos datagramas recebidos da camada de rede, que são denominados nesta camada como quadros e detectar e possivelmente corrigir, erros que possam ocorrer no meio físico;
- *Física* : É a camada que trata das características elétricas, ópticas e mecânicas do meio de comunicação, como tipos de conectores e dispositivos físicos utilizado para estabelecer uma conexão, o objetivo é a transferência de bits de um hospedeiro a outro.

Ainda se tratando de estrutura lógica, o que mais se espera de um administrador de redes é um bom conhecimento sobre os serviços de redes ao qual ele pode proporcionar a seus usuários e sua organização. Esse profissional, deve sempre estar atento ao surgimento e evolução de serviços.

A Tabela 2.1 ilustra algumas das aplicações comumente utilizadas em redes de computadores das mais variadas proporções, bem como suas respectivas portas de comunicação.

Protocolo	Porta	Serviço
FTP	TCP 20/21	Transferência de Arquivos
SSH	TCP 22	Terminal Remoto Seguro
TELNET	TCP 23	Terminal Remoto Não Seguro
SMTP	TCP 25	Transferência de E-mail
DNS	TCP/UDP 53	Tradução direta e reversa de Nomes de Domínios em Endereços de Rede
DHCP	UDP 68	Distribuição/Configuração automática de endereços de Rede
HTTP	TCP 80	Transferência de Conteúdo Hipertexto
HTTPS	TCP 443	Transferência de Conteúdo Hipertexto Seguro
POP3	TCP 110	Leitura de E-mail
IMAP	TCP 143	Leitura de E-mail
LDAP	TCP/UDP 389	Serviço de Diretório Remoto
LDAPS	TCP 636	Serviço de Diretório Remoto Seguro

Tabela 2.1: Aplicações comuns em Redes

O conhecimento de portas de comunicação dos serviços de rede é de suma importância para o administrador, pois é exatamente por meio destas que cada serviço ou aplicativo de rede mantém uma relação estrita com o sistema operacional, o qual é responsável por um gerenciamento das mesmas através de um mecanismo de enumeração peculiar e restrito a cada serviço. Um administrador de redes pode identificar, analisar e controlar o que esta sendo trafegado/utilizado em sua rede por intermédio da monitoração das portas de comunicação em seu sistema operacional.

Em um modelo Cliente/Servidor cada serviço de rede como os mencionados na Tabela 2.1, são disponibilizados por computadores especialmente configurados para tal propósito, esses computadores são denominados Servidores e ficam aguardando o estabelecimento de conexão em uma porta específica, análogo as já descritas. Um exemplo deste cenário é um Servidor de páginas na Internet, que fica aguardando o estabelecimento de conexões na sua porta 80, depois da conexão estabelecida serve as páginas para as aplicações Clientes, como os navegadores de Internet.

Como já mencionado, configuração e manutenção de um Servidor necessita de conhecimentos aprofundados do Sistema Operacional, o qual será preparado para atuar essencialmente sob uma aplicação na rede, isso remete a uma atividade muito semelhante a administração de sistemas computacionais.

2.3 Administração Computacional

Antunes [17], define administração de sistemas como o ato de gerir de forma eficiente os recursos de um computador. Estes recursos são compartilhados em uma arquitetura que inclui *discos, memória, cpu* etc. Tal definição se aplica perfeitamente à administração de redes de computadores, na qual o ato de gerir de forma eficiente os recursos inclui desde o aproveitamento eficaz da largura de banda até otimizações dos servidores e serviços oferecidos pela rede. Desse modo, considera-se não fazer sentido desvincular administração de redes da administração de sistemas. Pois a forma mais comum de se utilizar um computador é interligando-o a uma rede principalmente quando se trata de ambientes organizacionais como empresas, universidades, etc.

As redes de computadores dessas organizações estão tomando grandes proporções, em função da necessidade constante de automatização de tarefas proporcionadas pelo uso de sistemas computacionais. Entretanto, isso cria um cenário com sistemas heterogêneos, ou seja, com diferentes sistemas operacionais, aplicações e serviços de redes, exigindo ainda mais preparo do Administrador.

Neste contexto, as tarefas de administração de redes e administração de sistemas, acabam se fundindo, ficando difícil delimitar onde termina a administração de sistema e começa a administração redes ou vice versa.

Na grande maioria das organizações principalmente nas de pequeno e médio porte a administração de sistemas e administração de redes, se reúnem em atividade única, composta por várias tarefas realizadas por uma pessoa ou equipe e que exige extremo esforço, atualização e capacitação profissional.

Segundo dados obtidos pelas pesquisas realizadas pelo *SAGE*¹ é possível uma análise do perfil deste profissional, o que demonstra que a administração envolve outras inúmeras funções além da administração de redes e da administração de sistemas.

De acordo com o relatório da pesquisa realizada pela *SAGE* em 2001 [1], dos 1974 questionários válidos respondidos: 230 eram do sexo feminino o que representa cerca de 11,65% e 1744 eram do sexo masculino representando 88,35% dos administradores entrevistados. A Tabela 2.2 apresenta as funções desempenhadas por estes administradores no referido ano.

¹Grupo de Interesse Especial da Associação USENIX. Seu objetivo é servir a comunidade de administração de sistemas - <http://www.sage.org/>

Funções	% Rep.	Funções	% Rep.
Administrações de Sistemas	87.47	Engenheiro de suporte	25.28
Segurança	47.52	Base de Dados	20.01
Administrador de Redes	46.00	Treinamento	17.73
Gerenciamento	32.32	Gerenciamento de facilidades	12.87
Programador	26.14	Vendas	1.68
Administrações WEB	25.99	Arquitetura	1.52
Help Desk	25.63	Gerente de Projeto	25.28

Tabela 2.2: Funções Exercidas pelos Administradores em 2001 - *Fonte: SAGE* [1]

O Relatório de 2002 [2], mostra que dos 9651 questionários válidos 8973 eram do sexo masculino representando 92,97% dos entrevistados e 678 que representam 7,03% eram do sexo feminino. No entanto também é apresentada uma generalização das funções dos administradores, com seque na Tabela 2.3.

Funções	% Rep.	Funções	% Rep.
Administrações de Sistemas	33.69	Gerente de Projeto	3.85
Geral	22.69	Base de Dados	3.09
Segurança	14.27	Help Desk	2.84
Administração de Rede	12.53	Gerente de Pessoal	1.39

Tabela 2.3: Funções Exercidas pelos Administradores em 2002 - *Fonte: SAGE* [2]

Em 2003 o relatório [3], publicado com 4060 entrevistados, sendo 95,4% homens e 4,6% mulheres, já demonstra as funções desempenhada pelos administradores por meio de Gráficos enfatizando ainda mais a vasta gama de funções exercidas pelos administradores.

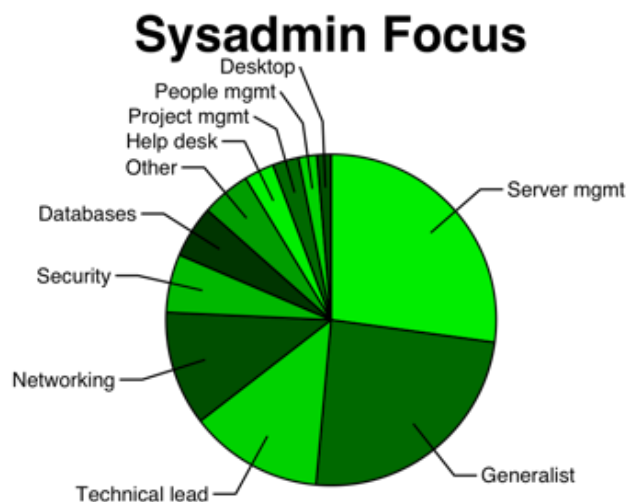


Figura 2.3: Funções Exercidas pelos Administradores em 2003 - *Fonte: SAGE* [3]

Já em 2004 o relatório [4], compreende resultados obtidos nos anos de 2004 e 2005. Neste período foram entrevistados 2974 administradores, sendo 96,2% do sexo masculino e 3,2% do sexo feminino. Entretanto o número de funções desempenhadas ainda é variada como demonstra a Figura 2.4:

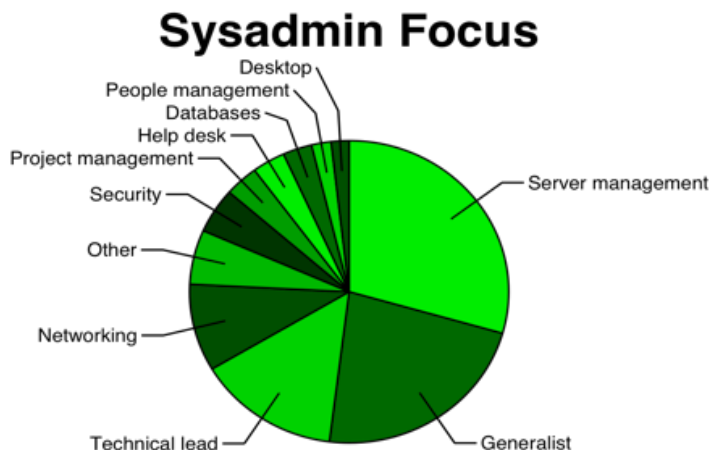


Figura 2.4: Funções Exercidas pelos Administradores em 2004/2005 - *Fonte: SAGE* [4]

O relatório [5] de 2005 disponibilizado pelo *SAGE*, compreende resultados das pesquisas feitas nos anos de 2005 a 2006. Percebe-se que houve uma queda entre o número de entrevistados, sendo 681 questionários válidos dos quais 91,6% são homens e 8,4% mulheres, no entanto a vasta gama de funções exercidas pelos administradores entrevistados continua grande como é ilustrado na Figura 2.5.

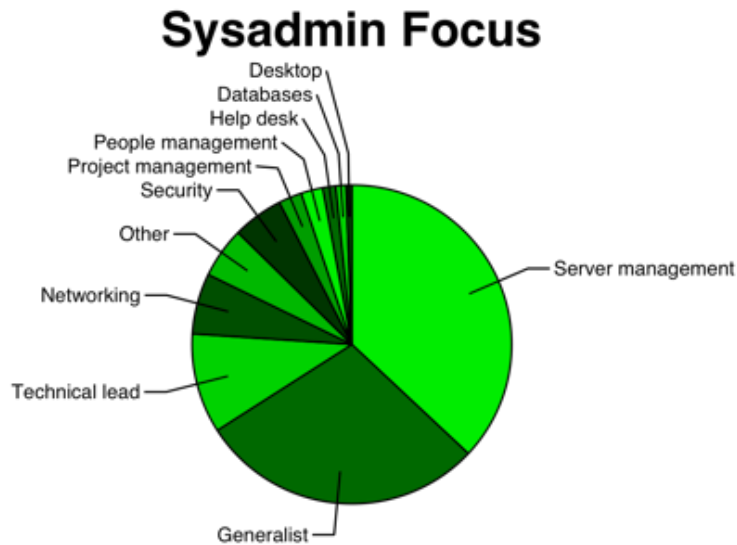


Figura 2.5: Funções Exercidas pelos Administradores em 2005/2006 - Fonte: *SAGE* [5]

Com base nos resultados apresentados pelas pesquisas realizada pelo *SAGE*², acredita-se que “Administração Computacional” se enquadra como melhor termo para designar tais atividades, visto que, as funções dos administradores passaram a se tornar as mais variadas possíveis dentro das organizações aumentando a complexidade das atividades deste profissional. Desse modo, pode-se caracterizar que a administração de redes é uma área da administração de sistemas e que o oposto também é verdadeiro, dependendo do foco que estiver sendo considerado.

Antes porém, as diretrizes do trabalho se concentra sobre a administração de redes e a administração de sistemas, que como visto tendem a realizar tarefas em comum, que muitas vezes estão apenas sendo definidas por nomenclaturas diferentes. Sendo assim, é apresentado na sequência uma pequena discussão sobre algumas das principais tarefas da administração computacional.

²<http://www.sage.org/>

Tarefas de Administração Computacional

As tarefas da administração computacional vêm se tornando um desafio para muitos profissionais, tanto para aqueles que atuam em pequenas e médias empresas quanto para os grupos que administram redes e sistemas de grandes organizações.

Para um administrador computacional o tempo em que as redes e seus sistemas ficam inoperantes devidos a falhas ou outros eventos, devem ser minimizado ao máximo. Cada segundo de indisponibilidade pode resultar em muito prejuízo para as organizações.

Em Geus [18] é possível encontrar um panorama da movimentação financeira realizada por algumas organizações no Brasil que utilizam redes e sistemas computacionais como meios para a realização de negócios, assim como os impactos causados pela indisponibilidade destes sistemas.

Para conseguir manter sistemas e redes operante no maior tempo possível é de suma importância o trabalho de planejamento e estratégia. Só assim é possível atuar nas mais diversas situações, principalmente nas de extremo risco. Essa atividade requer paciência, compreensão e experiência. É importante para um administrador computacional saber trabalhar em situações em que normalmente os recursos são limitados e restritos, neste contexto, a criatividade passa a ser de grande valia, por isso é importante o conhecimento de como os computadores funcionam ou se comportam nas mais variadas situações.

Assim como descrito por Burgess [14], a administração computacional é uma atividade muito parecida com a medicina, onde se precisa desenvolver técnicas comum as três áreas como o diagnostico, tratamento e prevenção.

Burgess [19], também coloca em destaque que aos administradores cabem as atribuições das seguintes tarefas:

- Modelagem da Arquitetura física e lógica da Rede de Computadores;
- Configuração de Equipamentos de Conectividade;
- Instalação e Configuração de *S.O.* e Periféricos;
- Configuração dos Servidores e Serviços;
- Administração de Contas de Usuários;
- Configuração das Estações de Trabalho;
- Interoperabilidade de Sistemas Operacionais e Aplicações heterogêneos;
- Monitoramento e Gerenciamento de sistemas, estações de trabalhos, servidores, serviços e da Rede;

- Prevenção, Detecção e Recuperação de Falhas;
- Suporte e Treinamento;
- Segurança ;
- Documentação das soluções existentes em seu ambiente, etc.

Esse grande número de tarefas incumbidas ao administrador torna-se ainda mais complexa quando os sistemas estão geograficamente dispersos, situação em que a presença física do administrador nos vários locais que se administre/configure se torna inviável. Uma possível solução para esta problemática seria a descentralização das tarefas, distribuindo-as entre vários administradores. Entretanto, a falta de sincronismo entre eles e a grande quantidade de recursos humano capacitado necessário são novos problemas que tornaram esse cenário não desejado [17].

Mencionado esse grande número de tarefas relativas ao administrador computacional, considera-se que profissionais que contemple este perfil são cada vez mais escassos. Na tentativa de suprir a falta de mão de obra qualificada, as ferramentas de administração que buscam auxiliar os administradores em suas atividades, automatizando muitos processos manuais, se apresentam como alternativas para possíveis soluções. Estas ferramentas, no entanto, serão o alvo do próximo capítulo deste trabalho.

2.4 Conclusão

Neste capítulo foram apresentados aspectos relacionados à administração computacional, para tanto foi ilustrado conceitos de administração de sistemas, aspectos de sistemas operacionais como também assuntos de administração de redes de computadores e seus fundamentos.

O capítulo também apresentou alguns resultados obtidos pela pesquisa realizada pelo *SAGE*, o que fortalece a idéia de utilização do termo “Administração Computacional” para se referir as atividades comuns em administração, principalmente no aspecto de administração de sistemas e administração de redes de computadores, mostrando que ambas atividades não caminham separadas e que na maioria das organizações dos mais variados portes, essas são atividades desenvolvidas pela mesma pessoa ou equipe. Finalizando o capítulo é enfatizado a complexidade envolvida na administração computacional, principalmente quando os sistemas estão geograficamente dispersos, situação em que a presença física do administrador nos vários locais se torna inviável.

Capítulo 3

Ferramentas de Administração Computacional

Este capítulo trata dos aspectos relacionados às ferramentas de administração computacional. O objetivo é apresentar as características das principais ferramentas utilizadas pelos administradores, na tentativa de auxiliá-los em suas atividades cotidianas. O capítulo se inicia com uma breve descrição das principais linguagens de *scripts*, utilizadas pelos profissionais para a construção de ferramentas caseiras, buscando a automatização de tarefas repetitivas. Seguidamente, é apresentado um estudo realizado com algumas das mais populares e importantes ferramentas de administração computacional. Neste estudo faz-se uma classificação desta ferramentas sob duas perspectivas: *(i)* ferramentas baseadas em ambientes de interface modo texto, sem parte gráfica. *(ii)* ferramentas de administração baseadas em interface gráfica, esta segunda se subdividindo em outras duas categorias, ferramentas gráficas de acesso local e ferramentas gráficas de acesso remoto. Finalizando o capítulo é apresentado uma comparação entres as principais ferramentas de cada categoria.

3.1 Classificação das Ferramentas de Administração

De acordo com Oliva [20], configurar uma máquina significa tomá-la de um estado qualquer, com o sistema recém instalado ou com arquivos de configuração parcialmente consistente, e criá-los ou editá-los, de modo a atingir o estado desejado.

Sendo assim, considera-se que uma das principais tarefas da administração computacional, em plataformas *UNIX*, é a manipulação e edição de arquivos de configuração ou manipulação de base de dados [21]. Esta tarefa por sua vez exige que o administrador tenha bons conhecimentos, sobre o sistema operacional e os serviços a qual deseja configurar.

É habitual que os administradores mais experientes procurem fazer a edição direta destes arquivos de configuração e as bases de dados, utilizando consoles e linhas de comando. Entretanto, em algumas situações isso pode ser uma tarefa muito penosa e desgastante.

Na tentativa de automatizar, acelerar e facilitar essa atividade, tanto para os administradores iniciantes quanto para os mais experientes, existem inúmeras ferramentas de administração/configuração.

Neste trabalho a fim de facilitar o estudo sobre as ferramentas de administração, faz-se uma classificação destas ferramentas da seguinte maneira:

1. Os *Scripts* desenvolvidos pelos próprios administradores como ferramentas “Caseiras” de administração;
2. As ferramentas de administração baseadas em interface a Modo Texto (sem interface gráfica);
3. As ferramentas de administração baseadas em Interface Gráfica.

Entretanto as ferramentas de interface Gráfica, podem ser classificadas em outras duas subclasses, *i)* As ferramentas de administração de acesso local, *ii)* Ferramentas de administração de acesso remoto.

Baseado nesta classificação, é apresentando um estudo sobre as mais populares ferramentas e técnicas utilizadas pelos administradores computacionais, para a realização de suas atividades profissionais.

3.1.1 Ferramentas Caseiras de Administração Baseadas em *Script*

Na hora de administrar/configurar uma coleção de máquinas, a manipulação direta dos arquivos texto, é um dos recursos muito utilizado pelos administradores, pois eles podem usar todo o potencial de manipulação de arquivos texto que o *UNIX* oferece, e assim construir ferramentas caseiras de automatização através de *scripts*, que são incomparáveis às funcionalidades de algumas ferramentas comerciais.

Os *Scripts* são arquivos textos que contem instruções em uma linguagem interpretada. Como são blocos de instruções, precisam de programas interpretadores para cada linha do arquivo, reunindo-as em um conjunto de comando que executem várias tarefas no Sistema Operacional [22].

Os programas interpretadores podem ser específicos de cada linguagem, como é o caso do *Python*, *Perl*, *Shell Script* etc. Um administrador tem a necessidade de desenvolver *scripts* como ferramentas caseiras, para automatizar tarefas repetitivas. No entanto, na administração computacional muitas vezes deve-se fazer a manipulação direta de arquivos texto, e então são incorporados aos *scripts* outras inúmeras ferramentas de fundamental importância. Como por exemplo as Expressões Regulares:

1. Expressões Regulares

Uma expressão regular segundo Silva [22], pode ser classificada como um conjunto de “regras”, que permite descrever precisamente todos os elementos de um conjunto, seja ele finito ou infinito, sem a necessidade de enumerá-los explicitamente.

As expressões regulares provém uma forma concisa e flexível de identificar cadeias de caracteres. São escritas em uma linguagem formal que podem ser interpretadas por um processador de expressão regular.

Por exemplo as expressões regulares a seguir:

a) ‘ $\wedge 0 - 9\$$ ’	b) ‘ $\wedge a - zA-Z\$$ ’
-------------------------	----------------------------

A expressão da letra *a*) identifica a cadeia de caracteres que começa com os algarismos de 0 a 9, a expressão da letra *b*) identifica a cadeia que começa com os caracteres Maiúsculos e Minúsculos de A a Z.

Outra ferramenta muito utilizada pelos administradores na construção de *scripts* é o *AWK*.

2. *AWK*

É uma linguagem de *scripting*, criada por Alfred V. Aho, Peter J. Weinberger e Brian W. Kernighan, para processamento de dados baseados em texto, sejam eles provenientes de arquivos ou de *streams* de dados. Seu nome *AWK* vem dos sobrenomes de seus criadores. Essa linguagem de *scripting* usa o tipo de dado *String* ou *arrays* associativos (*arrays* indexados por palavra chave) e expressões regulares [22].

É uma linguagem que possui blocos de códigos agrupados em comandos, o que a deixa com uma sintaxe reduzida. A função básica do *AWK* é a procura de determinado padrão em um linha (ou trecho de texto) de um arquivo. Assim, para cada casamento de padrão deve haver uma ação associada.

Por exemplo:

c) <i>AWK</i> ‘/ $\wedge [AEIOUaeiou]/$ ’ [arquivo de entrada]
--

O comando referente a letra *c*) imprime todas as linhas que começarem com as vogais em maiúsculo ou minúsculo, contidas no arquivo de entrada.

Uma outra alternativa muito utilizada em *scripts* pelos administradores é o *SED*.

3. *SED*

O *SED* é um editor para linguagens de *scripting*, significa *Stream Editor* ou editor de fluxo. É um editor muito utilizada nos sistemas *UNIX*, ao contrário dos editores convencionais, o *SED* atua em linha de comando ou integrado a *scripts* [22].

Uma das principais características é poder editar arquivos automaticamente, assim, sempre que for necessário fazer alterações sistemáticas em vários arquivos, o *SED* é uma solução eficaz, pois também aceita expressões regulares, o que lhe confere maior poder.

O *SED* trabalha com o conceito de endereço, deste modo quando se programa em *SED*, um comando se refere o que se quer fazer, enquanto um endereço refere-se onde se quer fazer, ou seja o endereço onde se quer executar o comando.

Existem três tipos de endereços:

- (a) **Endereço pelo número da linha;** É quando se quer referir a uma determinada linha.

```
sed '5 d' /etc/passwd
```

- (b) **Endereço pelo conteúdo da linha;** É quando se quer referir a linha que contém qualquer sequência de caracteres.

```
sed '/root/ d' /etc/passwd
```

- (c) **Endereço múltiplo, que abrange mais de uma linha;** É quando se quer as linhas que estão entre a linha X e Y.

```
sed '5,10 d' /etc/passwd
```

Com o comando da letra *a)* apaga-se a quinta linha do arquivo */etc/passwd*. O comando da letra *b)* apaga a linha que contém a palavra *root* no arquivo *passwd*. Com o comando da letra *c)* apaga-se desde a linha 5 até a linha 10 do arquivo *passwd*.

Embora essas ferramentas apresentadas até o momento sejam poderosíssimas, não são as únicas, há um conjunto de outras linguagens (ou ferramentas) de *scripting* como o *GREP*, *CUT*, *TR*, etc. que associadas entre si auxiliam o administrador a manter o seu sistema operante.

Essas ferramentas podem ser incorporadas juntas, em arquivos de *scripts*, que também podem ser escritos em algumas linguagens específicas para cada interpretador. Algumas das principais linguagens de *scripts* utilizadas pelos administradores são:

4. *Perl*

O *Perl* (*Practical Extraction and Report Language*) é uma linguagem de extração e elaboração de relatórios, foi criada em dezembro de 1987, por Larry Wall. O *PERL* integra os conceitos e fundamentos dos *shell script*, *AWK* e da linguagem *C*, estando disponível em praticamente todos os Sistemas Operacionais, mas é usado regularmente em sistemas *UNIX* e derivados [22, 23].

Em geral, a sintaxe de um programa escrito em *PERL* assemelha-se muito com a de um programa escrito em *C*, existem variáveis, expressões, atribuições, blocos de código delimitados, estruturas de controle e sub-rotinas. O *PERL* também possui muitas funções integradas para tarefas comuns como ordenação e acesso de arquivos em disco, entre outras.

Há um grupo de pessoas encarregado pelo desenvolvimento geral da linguagem, os “*Perl Porters*”. Desta forma garante-se pouca discrepância entre as versões, refletida na compatibilidade entre versões anteriores e as seguintes. Dado que o núcleo da linguagem é de código aberto, nenhuma companhia ou empresa pode apropriar-se da linguagem.

5. *Python*

O *Python* é uma linguagem de programação com características peculiares sendo interpretada, interativa, orientada a objetos mas, também com suporte para outros paradigmas, como a programação funcional e modular, onde os tipos de variáveis, retornos de funções e parâmetros são identificados pelo interpretador e os valores e objetos têm tipos bem definidos [24, 25].

A delimitação dos blocos de instruções é feita por indentação, não havendo delimitadores como as “{ }” da linguagem *C*, ou “*Begin*” e “*End*” da linguagem *ALGOL*.

Oferece tipos de dados de alto nível como *strings*, *dicionários*, *listas*, *classes*, entre outros. A sintaxe é fácil de ser compreendida e rápida de ser dominada, sendo considerada por muitos um dos grandes atrativos da linguagem.

Segundo Lutz [24] *Python* foi projetada para otimizar a produtividade do desenvolvedor, a qualidade do software, a portabilidade do programa e a integração dos componentes. Embora tivesse sido criada para ensino de programação de computadores, não é uma linguagem limitada ou com poucos recursos, é uma linguagem de altíssimo nível, ou seja, bem mais próxima do raciocínio humano que da arquitetura da máquina.

6. *Shell Scripts*

Um *Shell Script* ou *Script em Shell*, segundo Jargas [26] é um arquivo texto que reuni várias linhas de comandos a serem executados em sequência por um programa interpretador, no Sistema Operacional.

Interpretadores são programas feitos para servir de interface entre o usuário e seu Sistema Operacional. Através destes, o usuário executa determinadas funções do sistema por meio de linhas de comandos. Entre os interpretadores de comandos podem ser destacados:

- **Bourne shell (.sh)** Desenvolvido em 1977 por Stephen Bourne do laboratório *AT&T*. Por muito tempo foi o *shell* padrão de usuários em sistemas *UNIX*. É o *Shell* padrão do super usuário (root) nas implementações *UNIX* atuais.
- **Cshell - C Shell (.csh)** Como seu nome sugere, ele usa um modelo de sintaxe baseado na linguagem *C*. Talvez por ignorar o legado de facilidades didáticas do *.sh* está cada vez mais fora de uso no *GNU/Linux*. É mais utilizado em ambientes *BSD's* e *Xenix*.
- **Korn shell (.ksh)** Baseado no código do *Cshell*, foi o primeiro *shell* a introduzir recursos avançados. Foi desenvolvido por David Korn, do *Bell Labs*, e é um super-conjunto do *.sh*, isto é, possui todas as facilidades do *.sh* e a elas agregou muitas outras.
- **Bash - Bourne again shell (.sh)** Atualmente é o mais poderoso e mais utilizado interpretador para *Shell Script* em sistemas *GNU/Linux*. Desenvolvido para o projeto *GNU* é o padrão de várias distribuições *Linux*. É compatível com o *Bourne shell*, além de incorporar os melhores recursos do *C shell* e do *Korn Shell* e adicionar outros consecutivamente a cada nova versão.

A utilização de *Shell Script*, como linguagem de *script* para administração computacional em sistemas baseados em *UNIX*, é sem dúvida nenhuma a mais utilizada, principalmente pela grande facilidade de integração com o Sistema Operacional e pela reutilização de comando nativos do sistema. Além disso, os *Scripts* são de fácil compreensão e podem ser executados por outros programas que os inclua em seu código.

As Ferramentas caseiras baseadas em *Scripts*, de uma forma geral, facilitam a administração através da sua robustez, flexibilidade e capacidade de adequação as necessidades do administrador. Entretanto, algumas características os tornam inconvenientes em determinadas circunstâncias.

Por exemplo, para cada alteração que ocorra no cenário, o administrado deverá alterar seus *scripts* para adequá-los a nova realidade, o que não deixa a solução dinâmica. Mesmo que haja mínimas alterações os *scripts* terão que ser adaptados e aplicados as máquinas

novamente, trazendo de volta as tarefas morosas que o administrador procura automatizar quando cria um *script*.

Além do processo de adaptação dos *scripts*, não se pode esquecer a capacidade técnica necessária para a elaboração dos mesmos, pois além de um administrador é necessário que este profissional seja um bom programador, para que seus códigos não tragam brechas de segurança ao sistema, nem possa causar falhas de funcionamento a algum serviço a qual é de importância para os seus usuários e sua organização.

3.1.2 Ferramentas de Administração Baseadas em Interface Modo Texto

Como já foi descrito, uma boa parte de administradores costuma escrever *scripts* como ferramentas caseiras para poder automatizar determinadas tarefas repetitivas em seu cotidiano. Pensando nisso, foram desenvolvidas várias ferramentas de administração que utilizam um console em modo texto para serem executadas.

Essas ferramentas podem ser caracterizadas por serem executadas através de comandos, parâmetros e algumas vezes possuem uma linguagem muito peculiar para a sua execução, além de apresentarem respostas imediatas.

Entre as principais ferramentas que se enquadram neste perfil podem ser citadas:

1. Rdist

O Rdist¹ é um programa de código aberto que serve basicamente para manter arquivos idênticos entre máquinas diferentes. Foi um trabalho desenvolvido por Michael A. Cooper e apresentado para a comunidade acadêmica em 1992 [27].

Segundo Oliva [20], seu funcionamento baseia-se na criação de um arquivo de entrada para o *Rdist* na máquina servidora, chamado *distfile*, especificando quais são os arquivos de configuração que irão ser enviados, as máquinas que podem recebê-los e onde irão ser armazenados em cada máquina. Ainda é possível determinar comandos para serem executados nas máquinas logo após o recebimentos dos arquivos do servidor.

Entretanto, é necessário uma configuração prévia da máquina que vai receber as cópias dos arquivos, além de que, a mesma deve aceitar conexões do *Rdist* e para isso precisa de entradas nos arquivos *.rdist* ou *host.equiv*.

Outra desvantagem é a necessidade de alteração do arquivo de configuração no servidor sempre que for inserida uma nova máquina na rede. As novas versões desta

¹<http://www.magnicomp.com/rdist/>

ferramenta vem buscando intensamente suprir estas desvantagem com a inserção de novas possibilidade e a resolução de problemas como os apontados. O Rdist até o momento da composição deste trabalho está na versão *7.0.0-alpha*.

2. Config

John Rouillard e Richard Martin desenvolveram sobre o *Rdist o Config*, que permite administração distribuída do sistema, a solução se baseia em um diretório central de configuração, que era mantido sob controle do *CVS(Concurrent Version System)*, que realiza controle de versão e permite alteração concorrente [20].

Cada diretório controlado pelo *CVS* pode ter um *makefile*, que será acionado pela comando *Make*, assim, pode-se gerar configuração específica para determinada máquina.

A idéia é possibilitar a edição/alteração de cada arquivo de configuração localmente, depois armazenar no repositório utilizando o *CVS*, e propagar para as máquinas utilizando um *script* escrito em *Perl*, chamado *Rdist*, que utilizava técnicas do sistema *Rdist*(anteriormente mencionado), que toma como entrada um arquivo que contém informações específicas de cada máquina [20].

3. GeNUAdmin

Desenvolvido por Magnus Harlander, o *GeNUAdmin* é um sistema que permite controlar a administração de máquinas com Sistemas Operacionais baseados em *UNIX*, mas de diferentes distribuição, em um ponto central de administração. O *GeNUAdmin* armazena informações relevantes de cada sistema em um repositório central, na forma de arquivos texto [28].

Segundo Correia [21] o sistema esta escrito em *Perl* e constrói vários *scripts* em *.sh (Bourne Shell)* dinamicamente, na tentativa de auxiliar nas tarefas de configuração remota. Os pedidos de administração remota, são realizados pelos clientes, utilizando o comando *.rsh* nas máquinas da rede.

De acordo Magnus e Antunes [28, 17] a atualização da configuração de um sistema utilizando o *GeNUAdmin* segue os seguintes passos:

- (a) Construir a configuração do sistema no ambiente do *GeNUAdmin*.
- (b) Consultar a base de dados e analisar a consistência dos dados existentes com os dados relativos a configuração local do sistema.
- (c) Criar a configuração do sistema em um diretório local, sem alterar a configuração atual.

- (d) Instalar os *Targets (scripts)* e executá-los nas máquinas remotas.

A definição da configuração de um determinado serviço, pode ser dependente ou independente do Sistema Operacional em execução na máquina em questão, desde que seja baseado em *UNIX*. Este sistema está disponível Gratuitamente sob licença *GPL*.

4. MakeConfig

O *MakeConfig* é uma ferramenta de administração proposta por Alexandre Oliva [20]. A posposta desta ferramenta é utilizar os recursos mais primários de um Sistema Operacional baseado em *UNIX*, como o *AWK*, *SED*, *sh* e o *Make*.

A idéia se fundamenta principalmente em cima da ferramenta *Make*², que utiliza um arquivo de entrada chamado de *Makefile*, com regras que definem maneiras de construir ou atualizar arquivos a partir de outros.

Oliva [20] descreve o funcionamento do *MakeConfig* da seguinte maneira: No diretório onde se encontra o *Makefile*, que é o arquivo de entrada, armazena-se também arquivos preparados para configurar as máquinas, como por exemplo: *resolv.conf*, *host.equiv*, *shell's scripts* e outros inúmeros arquivos que terão o mesmo conteúdo em diversas máquinas, então basta definir regras nos *makfiles* que faça a copia destes arquivos para o diretório adequado, nas máquinas a serem configuradas.

Para atingir uma quantidade específica de máquinas, esta ferramenta necessitava de que este diretório, que contém os arquivos de configuração e o *Makfile* fosse exportado para todas as máquinas em questão. Assim em conjunto com a ferramenta utiliza-se um servidor *NFS (Network file system)*, para exportar esse diretório. E a execução do comando *Make*, ficava agendado no *CRON*³ de cada máquina, podendo ser executado inclusive no momento do *boot* da mesma.

Como apresentado, esta ferramenta traz alguns inconvenientes, como a necessidade de um servidor *NFS* para a exportação dos diretórios, uma pré configuração nas máquinas para a montagem deste diretório, a possível complexidade envolvida na elaboração dos *scripts* a serem executados, assim como os arquivos *Makefiles* executados pelo *MAKE*, além de uma manutenção de configuração dos *CRON's* de cada máquina de sua rede periodicamente. Entretanto, é importante ressaltar que na

²O *Make* é um software normalmente utilizado pra automatizar o processo de compilações de pacotes em sistemas baseados em *UNIX*

³O *CRON* é um serviço disponível em sistemas baseados em *UNIX*, com ele é possível agendar rotinas ou tarefas, de modo que o sistema execute-as periodicamente

época de sua criação foi considerada um avanço entre os modelos de ferramentas de administração.

5. Cfengine

O *Cfengine* foi criado por Mark Burgess, na faculdade de Engenharia de Oslo, na Noruega em 1993 [29, 20]. É considerado pelos administradores um dos sistemas mais maduros quando se trata de ferramenta de administração computacional.

O *Cfengine* se baseia no paradigma Cliente/Servidor, onde um servidor central possui os arquivos de configuração ideal para cada máquina da rede que receberá a configuração adequada as suas necessidades. A referência a essas máquinas podem ser por meios de grupos de máquinas identificadas no servidor por uma classe, assim como determinados comandos também podem ser agrupados por classes, desta forma, o administrador pode definir quais as classes a serem executada [30].

As máquinas cliente possuem um agente que se conecta ao servidor e verifica quais classes devem ser executadas e se possuem características que satisfaçam essas condições.

O *Cfengine* possui uma linguagem específica para a construção das classes, sua utilização e configuração não é exatamente um conjunto de *scripts*, mais sim um programa compilado e escrito em linguagem de programação *C*.

Essa ferramenta foi considerada um modelo inovador de administração, utilizando um servidor central e agentes clientes configurados nas máquinas a serem administradas, várias outras ferramentas passaram a adotar esta técnica em seu funcionamento posteriormente.

Em 2001 Beadnall e Mayhew [31], desenvolveram o *CfAdmin* o que seria uma ferramenta de interface gráfica para o *Cfengine*. O *CfAdmin* possui uma interface gráfica de acesso remoto pela WEB, e seu funcionamento se daria sob outras tecnologias envolvidas como o servidor WEB *Apache Tomcat* um banco de dados e o *Cfengine*. Mas a maneira mais comum de se utilizar o *Cfengine* é exatamente sozinho e independente de outras ferramentas.

Apesar do destaque em relação aos outros modelos de ferramenta, o *Cfengine* possui o inconveniente de uma configuração prévia do servidor e dos clientes, a necessidade de um conhecimento específico de sua linguagem de configuração, além de possuir algumas limitações na execução de comandos no lado dos clientes.

6. Slack

Slack desenvolvida por Alan Sundell, e mantido por administradores do *Google*, é uma das ferramentas utilizada pelo site de busca para administrar seus servidores [32].

Segundo o *site* do projeto⁴, *Slack* é um software livre de código aberto utilizada pelo *Google* para aplicação de *patches*, e configuração remota em seus servidores. Pouco ainda se sabe sobre esta ferramenta, a informação que se tem é que o sistema é constituído de um servidor central, que possui os arquivos de configuração necessários para determinados serviços. O servidor envia esses arquivos por meio de um servidor *Rsync* que copia os arquivos ou até mesmo diretórios para as máquinas clientes, que receberão a configuração.

Além do servidor *Rsync* a ferramenta distribui para as máquinas clientes arquivos chamados de “*role*”, que contém informações a respeito da execução de comandos no sistema, para a atualização a partir dos arquivos recebidos.

Outra importante característica do *Slack* é que se no meio de uma transferência de arquivos ou diretórios do servidor para o cliente, ocorra uma perda de conexão, o sistema é capaz de iniciar a transferência a partir de onde parou, em função do sistema *rsync* que é executado.

O inconveniente desta ferramenta é praticamente o mesmo das outras já descritas, a necessidade de configuração prévia no servidor central, como nas máquinas clientes e a possível dificuldade encontrada na elaboração dos *scripts* a serem executadas na máquinas remotas caso o serviço a ser configurado necessite.

7. Puppet

O *Puppet* é um sistema de código aberto, desenvolvido e mantido pelo *reductivelabs*⁵. O *Puppet* também é uma ferramenta que segue o paradigma Cliente/Servidor, centralizando a gerência de configuração em um servidor central [33].

Para utilizar esta ferramenta, executa-se um servidor central chamado *PuppetMaster* e os *Daemons* clientes do programa em todas as máquinas que se deseja administrar/configurar. No servidor central encontra-se toda a configuração adequada para cada estação da rede, que será propagado para as máquinas clientes em momentos determinados pelo administrador.

⁴<http://code.google.com/p/slack/>

⁵<http://reductivelabs.com/>

O *Puppet* é normalmente utilizado em uma formação estrelar, com todos os clientes conectado com um ou mais servidores centrais. Cada cliente contacta o servidor periodicamente (de 30 a 30 min. por padrão), recebe sua última configuração e garante que está em sincronia. Uma vez realizado este procedimento é possível o envio de um relatório de volta para o servidor indicando seu *status*. Existem duas características importantes desta ferramenta que devem ser colocadas em destaques:

1. O Servidor mantém um registro da configuração de cada cliente, assim se um sistema “morrer” é possível gerar uma configuração idêntica em um sistema recém instalado.
2. É possível fazer a configuração de forma idêntica em mais de uma máquina.

O *Puppet* também funciona em conjunto com uma outra ferramenta chamada *Facter*, que define variáveis específica em cada máquina, permitindo uma configuração peculiar em cada cliente. Entretanto, assim como a maioria, senão todas as ferramentas apresentadas, o inconveniente na sua utilização está na configuração prévia necessária nas máquinas clientes que rodam os *Daemons* e também no servidor onde se conecta esses *Daemons*. É importante ressaltar, a gama de complexidade exigida por esses modelos de ferramentas, onde o administrador necessita de conhecimentos aprofundados na elaboração de *scripts*, que terão suas execuções a serem agendadas em momentos específico. Outra característica peculiar a esses modelos de ferramentas, são as “linguagens” utilizadas para a construção ou configuração de seus arquivos e que também são responsáveis pela troca de informação entre os *Daemons* clientes e o servidor.

Comparação

Após a análise de algumas características, a Tabela 3.1 apresenta de forma sucinta uma comparação sobre o aspecto de funcionamento das ferramentas até aqui estudadas.

Nome/Caract.	Repositório de admin.	<i>Daemon</i> cliente	Execução de <i>scripts</i>	Lingua- gem de Config.	Ferram. auxiliar	Config. Prévia
Rdist	Não	Sim	Sim	Sim	SSH	Sim
Config	Sim (CVS)	Sim	Sim	Sim	Rdisr, CVS e GuMake	Sim
GeNUAdmin	Sim (Banco de dados de Arquivo)	Não	Sim	Sim	Rsh/NFS	Sim
MakeConfig	Sim (Di-retório)	Não	Sim	Sim	NFS e GnuMake	Sim
CfEngine	Não	Sim	Sim	Sim	Não	Sim
Slack	Sim (Di-retório)	Não	Sim	Sim	Rsync	Sim
Puppet	Sim (Di-retório)	Sim	Sim	Sim	Facte	Sim

Tabela 3.1: Características de Funcionamento das Ferramentas em Modo Texto

Além de ferramentas baseadas em interface “Modo Texto” já discutidas, constata-se a existência de inúmeras outras, que utilizam algumas das técnicas descritas, como também técnicas diferenciadas.

Como o objetivo neste momento é apenas ilustrar algumas destas ferramentas, apresenta-se a Tabela 3.2 com propriedades básicas de ferramentas mencionadas e outras que não foram descritas em função de sua baixa popularidade entre os administradores e a comunidade acadêmica, devido a pouca literatura encontrada sobre elas.

Nome / Caract.	Lin- guagem desenvol- vida	Cliente/ Servidor	Tipo de Li- cença	Usa Crip- tografia	Ano / Primeira versão	Data / Ultima versão estável
Arusha Project (ARK)	Python	Não	BSD	Sim	2001	2005-04-19
Bcfg2	Python	Sim	BSD	Sim	2004	2008-02-12 / 0.9.5.7
Quattor	Perl	Sim	EDG	Sim	2005	2007-12-12 / 1.3-2
ISconf	Python	Não	GPL	Não	1998	2006-08-13 / 4.2.8.233
LCFG	Python	Sim	GPL	Sim/Parcial	1994	2007-02-19 / 2007021901c
PIKT	C	Não	GPL	Sim	1998	2007-09-10 / 1.19.0
Radmind	C	Não	BSD	Sim	2002	2007-07-5 / 1.9.0
GeNUAdmin	Perl	Sim	GPL	Não	1994	?
Cfengine	C	Sim	GPL	Sim	1993	2008-06-11 / 2.2.7(espe- rada v3 para 01/2009)
Slack	Perl	Sim	GPL	Sim	?	2008-08-01 slack-0.15.2- 1
Puppet	Ruby	Sim	GPL	Sim	2005	2008-03-25 0.24.4

Tabela 3.2: Propriedades Básicas de Ferramentas de Modo Texto

Como visto, existem inúmeras ferramentas de administração computacional, baseadas em interface “Modo Texto”, isto é, sem interface gráfica, que buscam auxiliar os administradores em suas tarefas cotidianas.

Essas ferramentas entretanto, possuem algumas características idênticas, encontradas em quase todas descritas até então, como a necessidade de conhecimentos na elaboração de *scripts*, que serão a principal arma para a automatização de tarefas, visto que, a maioria das ferramentas descritas, se preocupam na forma de transferência destes *scripts*, ou arquivos já preparados, para as máquinas que serão configuradas. Outro ponto que merece ser colocado em destaque, é a necessidade de adaptação por parte dos administradores, em utilizar linguagens específicas para criar ou editar os arquivos de configurações das ferramentas, pois estes em sua maioria, são os responsáveis pelo estabelecimento e troca de informação entre os módulos clientes e servidor das ferramentas de administração.

Pode-se mencionar também, o lado desgastante e muitas vezes incomodo, para alguns administradores de se ter que trabalhar com ambiente a base de linhas de comandos em modo texto, cenário este, que por sinal é muito propício a erros que são causados principalmente por falhas de configuração e/ou mau edição de arquivos.

Seguindo um outro paradigma, estão as ferramentas de administração que utilizam interface gráfica, algumas das principais e mais populares serão analisadas na sequência.

3.1.3 Ferramentas de Administração Baseadas em Interface Gráfica

Ferramentas de administração baseadas em “Interface Gráficas” que cubram automação em escala, como as proporcionadas pelas ferramentas com interface em “Modo Texto”, ou ferramentas caseiras baseadas em *scripts*, são raras e caras e é justamente aí que está o desafio. Criar tal modelo de ambiente gráfico, disponibilizando as facilidades de automação que proporcione ao administrador a flexibilidade necessária para a árdua tarefa de administração/configuração em ambientes *UNIX*.

Dentre as aplicações Gráficas desenvolvidas para administração podem ser evidenciadas duas classes *i)* as ferramentas de administração de acesso local, *ii)* as ferramentas de administração de acesso remoto, como segue:

Ferramentas de Administração de Acesso Local

São consideradas ferramentas de administração de acesso local, aquelas ferramentas que o administrador utiliza para gerir as máquinas que necessitam de sua presença, ou seja, é necessário que ele esteja fisicamente na frente da máquina a qual está configurando.

Entre as ferramentas de acesso local, usadas para administração computacional, pode-se colocar em destaque o grupo de ferramentas:

1. Linux System Administration Tools

São ferramentas de administração de acesso local, incorporadas por algumas distribuições *Linux*, para auxiliar o Administrador.

Nessa classe, é possível encontrar ferramentas gráficas utilizadas para a configuração de serviços ou dispositivos específicos, como é o caso de alguns aplicativos utilizados para a configuração de dispositivos de redes, dispositivos de áudio, serviço de compartilhamento de arquivo, entre outras finalidades.

A Distribuição *Linux Red Hat*⁶ (e suas descendentes) é considerada uma das distribuições que possui o maior número de ferramentas desta natureza. Essas ferramentas são conhecidas nestas distribuições, por *system-config-tools*. Deste modo, existem inúmeras aplicações independentes utilizadas para a configuração de dispositivos ou serviços individualmente. Pode-se citar como exemplo o *system-config-network* para configuração do dispositivo de rede e o *system-config-http* para a configuração de um servidor *WEB*, como visto nas Figuras 3.1 e 3.2.

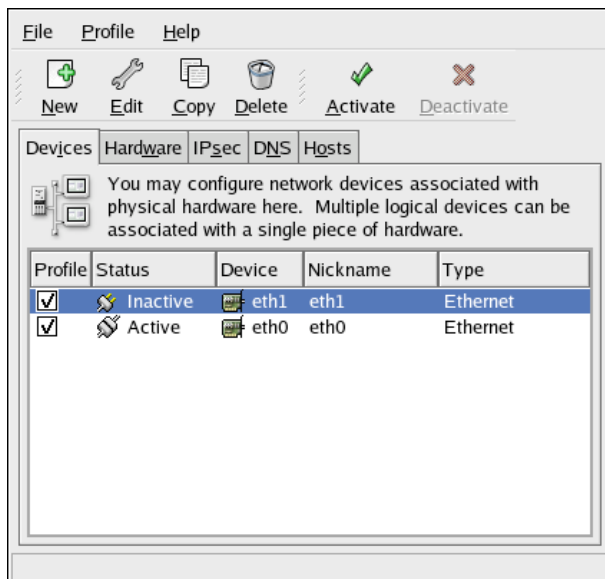


Figura 3.1: System-Config-Network

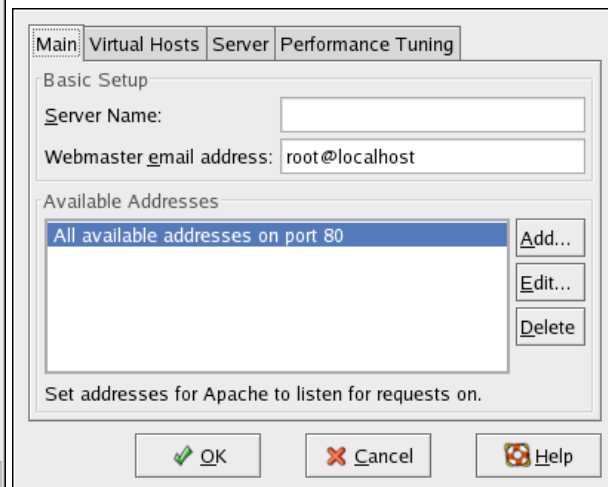


Figura 3.2: System-Config-Http

No entanto, o foco deste trabalho não é ferramentas individuais, utilizadas para configuração de dispositivos ou serviços específicos, mas sim ferramentas que englobam a configuração de vários serviços ou dispositivos em um único aplicativo.

Dentre as ferramentas mais populares nesta linha evidenciam-se:

⁶[http:// www.redhat.com](http://www.redhat.com)

2. Linuxconf

O *Linuxconf*⁷ liderado por Jacques Gélinas, seu principal desenvolvedor, foi um dos primeiros projetos de sistema visando a administração/configuração de máquinas *Linux*. Seu foco se resume na maneira rápida e fácil de se configurar e administrar um computador, o que inclui dispositivos físicos como o hardware em geral e serviços de rede como servidores *DHCP*, *DNS*, *NFS* etc. Isso por Interfaces amigáveis, para usuários que preferem configurar o sistema através de uma interface gráfica, tendo à sua disposição janelas, caixas de diálogo, botões, etc [34].

A ferramenta começou como um sistema proprietário e atualmente é de código livre sob licença *GPL*, pode ser instalada em qualquer distribuição *Linux*, entretanto, foi incorporada como ferramenta padrão de configuração e administração por algumas distribuições, como é o caso da antiga *Conectiva Linux* atualmente *Mandriva*, entre outras.

a) Arquitetura do Linuxconf.

O *Linuxconf* é um sistema escrito em *C++* que contem cerca de 80.000 linhas de códigos, foi desenvolvido de forma modular, para facilitar a criação e incorporação de novos módulos, capazes de realizar configurações de serviços específicos em sistemas *Linux* [35]. Tendo em vista esta característica, para a adição uma nova funcionalidade é necessário somente o desenvolvimento de um novo módulo para que seja executado a tarefa específica. Com isso, consegue-se uma ferramenta que pode centralizar a configuração de todo o sistema. Sob esse ponto de vista, ainda vale ressaltar que o *Linuxconf* subdivide suas configurações em tarefas de clientes, ou seja, configurações em máquinas que serão clientes de algum tipo de serviço, e tarefas de servidor, que é a configuração de um servidor em específico.

O sistema possui uma *API* (*Application Programming Interface*)⁸ central, escrita em *C++*, a qual os módulos podem ser conectados e executados, estes módulos podem ser desenvolvidos em *Shell Scripts* e serem executados por um outro módulo próprio do *Linuxconf*, chamado *Shellmod*, assim, é possível utilizar os módulos do *Linuxconf* por meio de sua interface gráfica ou incorporá-los em *scripts* próprios [34]. Desta mesma maneira é possível que se habilite ou desabilite módulos específicos dentro da ferramenta.

⁷<http://www.solucorp.qc.ca/linuxconf/>

⁸*API* é um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por programas aplicativos

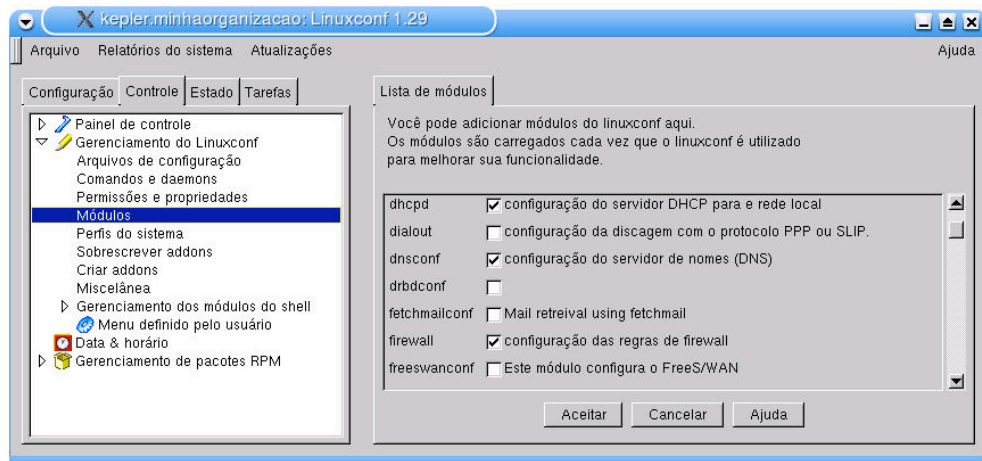


Figura 3.3: Módulos do Linuxconf

Graças a característica de modularidade, o *Linuxconf* possui três (3) tipos de interface gráfica: uma baseada em “Modo Texto”, executada por meio de linhas de comando, outra utilizando “Interface Gráfica” de acesso local, com biblioteca *Qt*⁹ e *GTK*¹⁰, e outra interface gráfica de acesso remoto via *WEB* por intermédio de navegadores de internet, todas com as mesmas funcionalidades [35]. Entretanto, a sua interface mais consistente e madura, é a interface de acesso local, por isso essa ferramenta foi enquadrada nessa categoria.

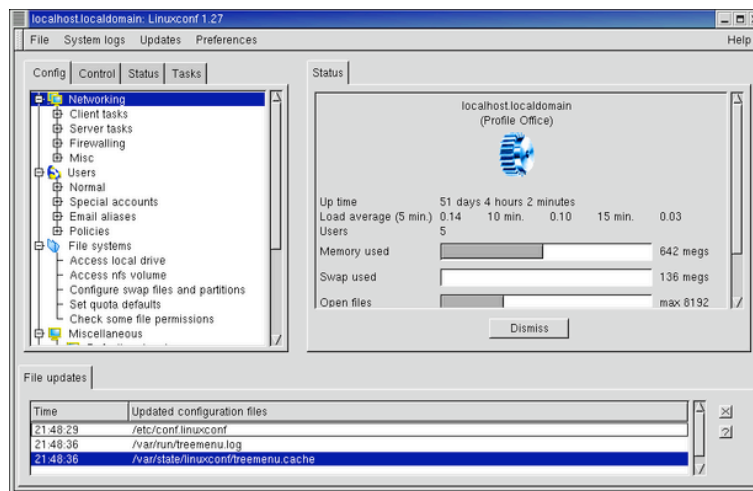


Figura 3.4: Interface do Linuxconf

O *Linuxconf* é uma ferramenta versátil e poderosa sendo utilizada por muitos ad-

⁹*Qt* é um sistema multiplataforma para o desenvolvimento de programas de interface gráfica

¹⁰*GTK* é um *toolkit* multiplataforma para a criação de interfaces gráficas

ministradores, ela traz características que a diferenciam das demais ferramentas, principalmente pela sua modularidade e facilidade de manuseio. Mas, a manutenção da ferramenta por parte de seus desenvolvedores, deixa a desejar fazendo com que a ferramenta tenha poucas atualizações e sendo considerada por muitos como um projeto descontinuado.

3. YaST

O *Yast* (*Yet another Setup Tool*) [36], é a ferramenta padrão de instalação e configuração das distribuições *OpenSUSE* e *SUSE Linux Enterprise*. A primeira distribuição *SuSE* que incluiu o *YaST* foi lançada em maio de 1996, mas seu desenvolvimento iniciou-se em janeiro de 1995. Foi escrito inicialmente em linguagem *C++* por Thoamas Fehr (um dos fundadores da *SuSE*) e Michael Andrés. O Projeto começou como uma ferramenta de código fechado, atualmente já está com seu código aberto para que novos programadores possam contribuir no seu desenvolvimento.

O *YaST* é popular por ser de fácil utilização e possuir uma interface gráfica atraente, além da capacidade de personalizar o seu sistema rapidamente durante e após a instalação.

A ferramenta pode ser usada para configurar o seu sistema inteiro. É possível fazer configurações de hardware, configurações de rede, configurações de segurança e de inúmeros serviços de sistema, para tanto basta apenas com alguns cliques do mouse a partir do *YaST Control Center*.

As tarefas desta ferramenta são agrupadas em sessões, para facilitar a sua utilização pelos administradores menos experientes. Acompanhe as Figuras 3.5 e 3.6.

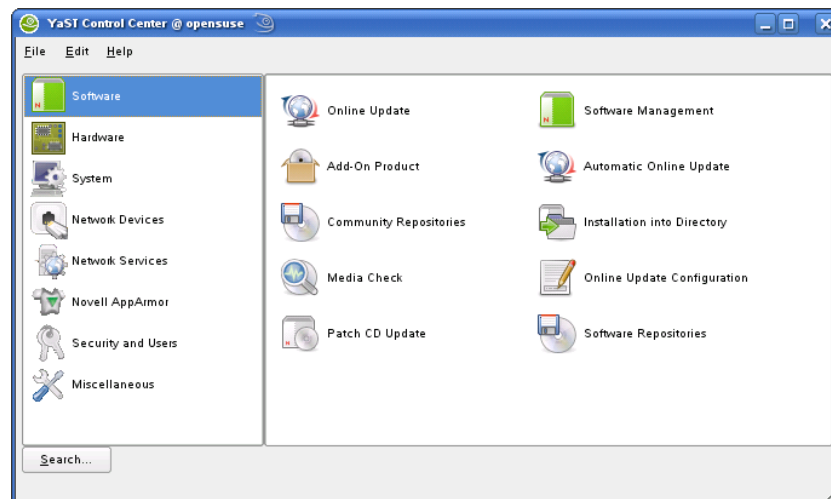


Figura 3.5: YaST Control Center

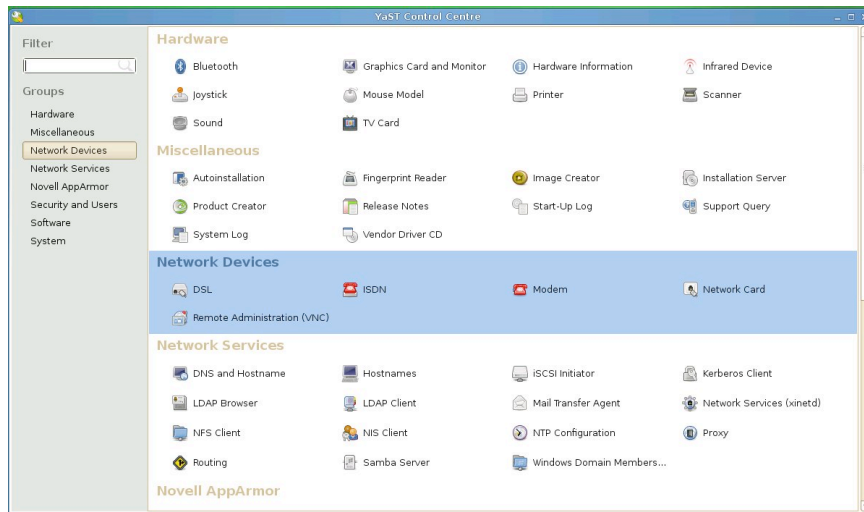


Figura 3.6: Seções do YaST

a) Arquitetura do YaST.

Segundo Novell Inc. Team [37], o *YaST* é desenvolvido de forma modular, com um número de componentes individuais e específicos para cada tarefa. O núcleo do *YaST* e os principais módulos são escritos em uma linguagem própria, chamada *YCP* (*YaST Control Language*) e/ou podem ser escritos em *Perl*. A interface gráfica do *YaST* é proporcionada por meio de uma *API* específica de um dos módulos desenvolvidos em *YCP*, esta *API* executa as funções do servidor *X11* para gerar a interface gráfica de acesso local.

A Figura 3.7, encontrada no Tutorial de Módulos do *YaST* [38], mostra um esquema da arquitetura da ferramenta, sob o ponto de vista da implementação de um módulo de configuração de um servidor *SSH*, no entanto, o mesmo procedimento estrutural é realizado para a configuração de outros serviços também.

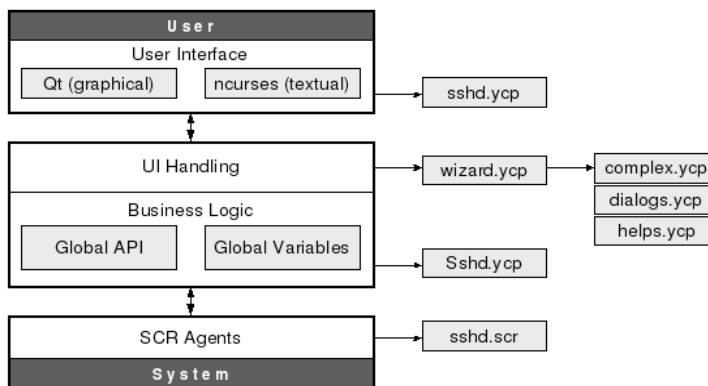


Figura 3.7: Arquitetura do YaST

Como ilustra a Figura 3.7 o *YaST* tem sua arquitetura subdivida em quatro (4) camadas:

1. Camada de interface (*User Interface*), ao qual é o modo de interação entre o usuário é a ferramenta de administração de acesso local, nesta camada localizam-se dois módulos o *QT* e *ncurses*, um responsável pela geração da interface gráfica e o outro pela interface textual.
2. Camada de *UI Handling*, tem a função de executar o tratamento de eventos realizadas pelos usuários, como a geração dos diálogos, para sequência da configuração.
3. Camada de Lógica de Negócio (*Business Logical*), responsável pelo tratamento de dados sobre a estrutura interna de acesso a camada de *Agentes SCR*. Para isso contem dois módulos: o *Global API* e *Global Variables*, um responsável por uma *API* Global funcional e o outro por Variáveis Globais, utilizadas para aplicações clientes com interface a base de linhas de comandos.
4. Camada de Agentes de Sistema, *SCR Agents*, essa camada possui pequenos programas para acessar o sistema de arquivo, base de dados, comandos do sistemas etc. Esses programas são chamados de *agentes SCR*.

O *YaST* é considerado por muitos administradores uma das maiores ferramentas de administração. Entretanto, existem algumas características que tornam a ferramenta inviável em alguns cenários e um dos principais motivos é exatamente o exclusivo suporte as distribuições *OpenSuSE*, *SuSE* e derivados. Sendo extremamente complicada a sua instalação e utilização em outras distribuições *Linux*.

Apesar do *YaST* e *Linuxconf* se apresentarem como as ferramentas mais populares entre os administradores, existem semelhanças e diferenças entre elas que podem ser fundamentais na hora de decidir qual utilizar. A Tabela 3.3 apresenta algumas características relevantes de cada uma das ferramentas que podem ajudar no momento de escolha de uma delas como a ferramenta de administração de acesso local a ser utilizada.

Características/Ferramentas	Linuxconf	YaST
Construção Modular	sim	sim
Módulos por Serviços	sim	sim
Inserção de Módulos	sim	sim
Linguagem dos Módulos	Shell Script	Perl / YCP
Linguagem do Módulos Principal	C ++	YCP (Ling. própria)
Interface Amigável	bom	ótima
Suporte	ruim	ótimo
Atualização	ruim	bom
Portabilidade de plataforma Unix	bom	ruim
Ano de Lançamento	01-10-1996	01-05-1996
Ultima Versão/Data	1.34r3 / 18-01-2005	SuSE 11 / 19-06-2008

Tabela 3.3: Características do YaST e Linuxconf

Contudo, esses modelos de ferramentas apesar de possuírem características e peculiaridades relevantes para a administração computacional, são ferramentas que só podem ser acessadas localmente, isso por não possuírem recursos de acesso remoto ou ainda não estarem maduras o suficiente para este propósito, sendo necessário a presença física do administrador no local da configuração. No entanto, essa situação se torna cada vez mais indesejada, devido as grandes proporções tomadas pelas redes e sistemas computacionais de inúmeras organizações. Sendo assim, novas aberturas aparecem para aplicações que tentam sanar essa deficiência, a qual são classificadas neste trabalho como ferramentas de acesso remoto.

Ferramentas de Administração de Acesso Remoto

Para tentar solucionar a deficiência apresentadas pelas ferramentas de administração de acesso local, em manter uma pessoa presente fisicamente configurando máquinas e sistemas computacionais, desenvolvem-se novas técnicas e ferramentas de acesso remoto para apoiar a administração computacional, isso sem a necessidade de presença física do administrador no local da máquina a ser configurada. Entre as ferramentas e mecanismos de acesso remoto de apoio à administração, pode-se citar:

1. Remote Desktop

Remote Desktop, são softwares utilizados para acessar máquina remotamente, então, a partir desta conexão é possível que o administrador configure essas máquinas utilizando alguma ferramenta de administração, ou até mesmo fazendo a configuração

de forma manual. Esses modelos de software não são o foco do trabalho, pois são apenas meios utilizadas para se conectar as máquinas remotas e não ferramentas de administração ou configuração, apresenta-se apenas as Figuras 3.8 e 3.9 como uma Tabela comparativa encontrada em [6], com os principais sistemas deste gênero e em seguida algumas características que tornam essas ferramentas “imprópria” para tal atividade, mesmo assim, algumas vezes utilizadas pelos administradores como única alternativa.
















Program 	Protocol 	License 	Client 	Server 	Java Viewer 	Built-in Encryption 	File Transfer 	Audio Support 	Multiple Sessions 	Seamless Window 	FreeBSD 	Linux 	Mac OS X 	Microsoft Windows 
Apple Remote Desktop	RFB(VNC)	Proprietary	✓	✓	X	AES-128	✓	X	✓	X	X	X	✓	X
Apple Screen Sharing (iChat)	Proprietary, RFB(VNC)	Proprietary	✓	✓	X	AES-256	X	X	✓	X	X	X	✓	X
Aqua Connect	RDP, RFB(VNC), X11	Proprietary	✓	✓	✓	SSL, TLS	✓	✓	✓	X	✓	✓	✓	✓
Bomgar	RFB(VNC), X11	Proprietary	✓	✓	X	SSL, AES-256	✓	✓	✓	X	✓	✓	✓	✓
Cendio ThinLinc	RFB(VNC)	Proprietary	✓	✓	✓	SSH	✓	✓	✓	✓	X	✓	✓	✓
Citrix XenApp	RDP, ICA	Proprietary	✓	✓	✓	SSL, TLS	✓	✓	✓	X	✓	✓	Viewer only	✓
DameWare Mini Remote Control	RDP, Proprietary	Proprietary	✓	✓	X	SSL, AES-256	✓	✓	✓	X	X	X	X	✓
EchoVNC	RFB(VNC)	GPL	✓	✓	X	✓	✓	X	✓	X	X	Viewer only	✓	✓
Ericom PowerTerm WebConnect	RDP, RFB(VNC)	Proprietary	✓	✓	X	✓	✓	✓	✓	✓	X	✓	✓	✓
Fog Creek Copilot	RFB(VNC)	Proprietary	✓	✓	X	SSL	✓	X	X	X	X	X	✓	✓
FreeNX	NX, RDP, RFB(VNC)	GPL	✓	✓	X	AES	✓	✓	✓	X	✓	✓	X	X
GoToMyPC	?	Proprietary	✓	✓	✓	AES-256	✓	✓	✓	X	X	✓	Viewer only	✓
I'm InTouch	Proprietary	Proprietary	✓	✓	✓	SSL	✓	✓	✓	X	X	Viewer only	Viewer only	✓
iTALC	RFB(VNC)	GPL	✓	✓	X	SSH	✓	✓	✓	X	X	✓	X	✓
IBM Director Remote Control	Proprietary	Proprietary	✓	✓	✓	AES, 3DES, DES	✓	X	✓	X	X	X	X	✓
jrdesktop	Proprietary	GPL	✓	✓	✓	SSL, TLS	✓	X	✓	X	✓	✓	Viewer only	✓
LogMeIn Free	Proprietary	Proprietary	✓	✓	✓	SSL	X	X	X	X	X	X	✓	✓
LogMeIn Pro	Proprietary	Proprietary	✓	✓	✓	SSL	✓	✓	✓	X	X	X	X	✓
Mac HelpMate	RFB(VNC)	Proprietary	✓	✓	✓	AES-128	X	X	✓	X	X	X	✓	✓
Netsupport Manager	Proprietary	Proprietary	✓	✓	✓	✓	✓	X	✓	X	✓	✓	✓	✓
NX Server	NX, RDP, RFB(VNC)	Proprietary	X	✓	X	AES	✓	✓	✓	X	✓	✓	X ^[1]	X
PROXY Pro	Proprietary	Proprietary	✓	✓	X	AES-256, SSL	✓	X	✓	X	X	X	X	✓

Figura 3.8: *Remote Desktop* - Fonte:[6]

RealVNC Free	RFB(VNC)	GPL	✓	✓	✓	X	X	X	✓	X	✓	✓	X	✓ ^[13]
RealVNC Personal	RFB(VNC)	Proprietary	✓	✓	✓	AES-128	✓	X	✓	X	X	X	X	✓
RealVNC Enterprise	RFB(VNC)	Proprietary	✓	✓	✓	AES-128	✓	X	✓	X	✓	✓	✓	✓
Remote Administrator	Proprietary	Proprietary	✓	✓	X	AES-256	✓	X	✓	X	X	Viewer only ^[4]	X	✓ ^[5]
Remote Desktop Connection	RDP	Proprietary	✓	✓	✓	Varies	✓	✓	✓	✓ ^[6]	X	Viewer only ^[7]	Viewer Only	✓
Remote Graphics Software	Proprietary	Proprietary	✓	✓	Client Only	✓	✓	✓	✓	X	X	✓	X	✓
ScreenConnect	Proprietary	Proprietary	✓	✓	Client Only	AES-256	✓	X	✓	X	X	✓	✓	✓
SSH with X forwarding	X11	BSD	✓	✓	X	✓	X	X	[8]	✓	✓	✓	✓ ^[9]	✓ ^[9]
Sun Secure Global Desktop Software	AIP	Proprietary	✓	✓	✓	SSL, TLS (AES-256)	✓	✓	✓	✓	✓	✓	Viewer only	Viewer only
Symantec pcAnywhere	Proprietary	Proprietary	✓	✓	✓	AES-256	✓	X	✓	X	X	✓	✓	✓
TeamViewer	Proprietary	Proprietary	✓	✓	X	AES-256	✓	X	✓	X	X	X	✓	✓
TightVNC	RFB(VNC)	GPL	✓	✓	✓	X	✓	X	✓	X	✓	✓	X	✓
Timbuktu	Proprietary	Proprietary	✓	✓	X	✓	✓	X	✓	X	X	X	✓	✓
Timbuktu ToGo	Proprietary	Proprietary	✓	✓	✓	✓	✓	✓	✓	X	X	X	✓	✓
UltraVNC	RFB(VNC)	GPL	✓	✓	✓	With plugin	✓	X	✓	Partial ^[10]	X	X	X	✓ ^[11]
visionapp Remote Desktop	RDP, ICA, VNC, SSH, Telnet	Proprietary	✓	✓	X	AES-256	X	X	✓	X	X	X	X	✓
Webex	Proprietary	Proprietary	✓	✓	✓	✓	✓	✓	✓	X	X	✓	✓	✓
WebTrain	Proprietary	Proprietary	✓	✓	N/A	SSL, TLS	✓	✓	✓	X	X	X	X	✓
XDMCP	X11	MIT	✓	✓	X	X	X	✓	Viewer only	X	✓	✓	✓	X
xpra	Bencode-based	GPL	✓	✓	X	SSH	X	X	✓	✓ ^[12]	✓	✓	X	X
xrdp	RDP	GPL	X	✓	N/A	✓	X	X	✓	X	✓	✓	X	X
X11vnc	RFB(VNC)	GPL	X	✓	✓	SSL	✓	X	✓	X	✓	✓	✓	X
AnywhereTS	RDP, ICA	Proprietary	✓	X	X	SSL, TLS	✓	✓	✓	X	X	X	X	✓
Chicken of the VNC	RFB(VNC)	GPL	✓	X	X	X	X	X	X	X	X	X	✓	X
JollysFastVNC	RFB(VNC)	Proprietary	✓	X	X	SSL, TLS, SSH	X	X	✓	X	X	X	✓	X
KRDC	RFB(VNC), RDP	GPL	✓	X	X	X	X	X	✓	X	✓	✓	X	X
NX Client	NX, RDP, RFB(VNC)	Proprietary	✓	X	✓	Varies	✓	✓	✓	X	✓	✓	✓	✓
rdesktop	RDP	GPL	✓	X	✓	✓	✓	✓	✓	✓ ^[2]	✓	✓	✓	✓
x2x	X11	BSD	✓	X	X	X ^[13]	X	X	✓	X	✓	✓	✓	✓ ^[14]
Program	Protocol	License	Client	Server	Java Viewer	Built-in Encryption	File Transfer	Audio Support	Multiple Sessions	Seamless Window	FreeBSD	Linux	Mac OS X	Microsoft Windows

Figura 3.9: *Remote Desktop (cont.)* - Fonte:[6]

Como apresentado, existem inúmeros programas a serem utilizados como uma solução para *Desktops Remotos*. Durante o período da pesquisa sobre esses aplicativos, aceita-se como consenso o quadro apresentado na *Wikipedia* [6], como sendo um retrato da situação atual desses aplicativos, que podem ser uma alternativa aos administradores, quando os mesmos necessitam de realizar configurações em máquinas remotas.

Utilizando estes aplicativos os administradores passam a ter um controle completo das máquinas remotas e isso os possibilita a utilizar inclusive aplicações instalada

nas próprias máquinas, como as ferramentas de administração de acesso local.

Entretanto, essas soluções não se mostram viável quando se necessita de agilidade e comodidade, pois estas, sempre necessitam de um *software* Cliente e um Servidor, para poder funcionar adequadamente. Isto implica em uma configuração prévia de ambas as partes, tanto na máquina servidora como na cliente, aumentando ainda mais a dificuldade na hora de realizar as configurações necessárias nas máquinas remotas.

Outra desvantagem neste tipo de cenário é a sobrecarga que estas aplicações, ou sua maioria, causam na rede, aumentando o tráfego de pacotes nos *links* de dados. Isto acontece porque estas aplicações utilizam protocolos complexos e pesados para poderem proporcionar a conectividade entre um servidor e um cliente utilizando um ambiente gráfico completo. Como alternativa às aplicações com essas características encontram-se as ferramentas baseadas na WEB.

Ferramentas de Administração Baseadas na WEB

As Ferramentas de Administração Baseadas na WEB, vêm se apresentando como uma solução aos problemas causados pela sobrecarga de fluxo de pacotes nos *links* de dados, das redes de computadores e se mostram melhores plataformas para implementação de ferramentas de administração, por esta característica. Isso pode ser visto pela grande gama de sistemas desenvolvidos voltados para WEB.

As ferramentas de administração com este perfil, possuem uma interface gráfica desenvolvida para ser acessada através de um navegador de internet e utilizam protocolos mais “leves” que os utilizados pelas ferramentas de *Desktop Remoto*.

Entre os protocolos mais utilizados pelas ferramentas de administração de acesso remoto baseadas na WEB, pode-se citar o *HTTP* (*Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertexto) e o *HTTPS* (que é o *HTTP* modo seguro, com criptografia), utilizados para o estabelecimento de conexão entre um cliente, por meio de um navegador de internet, e um servidor WEB, máquina que contém a ferramenta de administração instalada.

Por funcionarem sob uma arquitetura *Cliente/Servidor*, essas ferramentas se assemelham ao funcionamento das páginas de internet, onde parte de seu processamento pode ser realizado por tecnologias executadas do lado cliente e outras partes executadas do lado servidor, dividindo o processamento da aplicação entre o cliente e o servidor, o que isenta o servidor de realização de todo o trabalho e evita a sobrecarga de processamento do mesmo.

A Figura 3.10 ilustra um esquema de funcionamento das ferramentas de administração baseadas na WEB, sob a arquitetura *Cliente/Servidor*, com as respectivas operações realizadas por cada lado no processo.

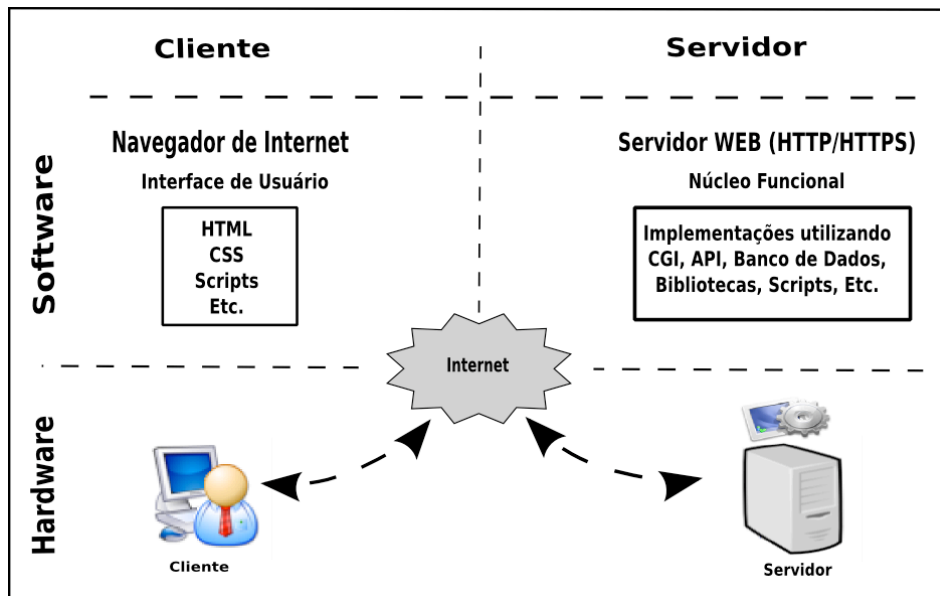


Figura 3.10: Funcionamento sob Arquitetura Cliente/Servidor

Como ilustrado na Figura 3.10, todas as operações que dizem respeito a interface gráfica, são executada do lado cliente, não exigindo processamento do servidor para esta finalidade, cabendo ao servidor somente a responsabilidade de enviar os respectivos arquivos referentes a interface para o cliente e ele os encarrega de processá-los por meio de seu navegador. Desta forma, todo processamento do servidor fica concentrado no Núcleo Funcional, que são as operações que dizem respeito as execuções de chamadas ao sistema, para a administração/configuração de um serviço ou dispositivo específicos, informados pelo administrador por parâmetros fornecidos a interface gráfica, preenchidos em campos de formulários, etc. Entre as principais ferramentas com esta características podem se citadas:

2. SWAT

SWAT (Samba WEB Administration Tools), é uma ferramenta de acesso remoto de código livre, capaz de administrar/configurar um servidor *Samba*, foi desenvolvida por Andrew Tridgell para ser utilizada através de um navegador de internet como uma solução de ferramenta de administração baseada na WEB.

O *SWAT* é baseada em uma interface gráfica com formulários, que possuem campos referentes aos parâmetros do arquivo de configuração *smb.conf*, responsável pelo funcionamento do servidor *Samba*, que devem ser preenchidos pelo administrador de acordo com as suas especificidades.

A Figura 3.11, apresenta a interface gráfica do *SWAT*.

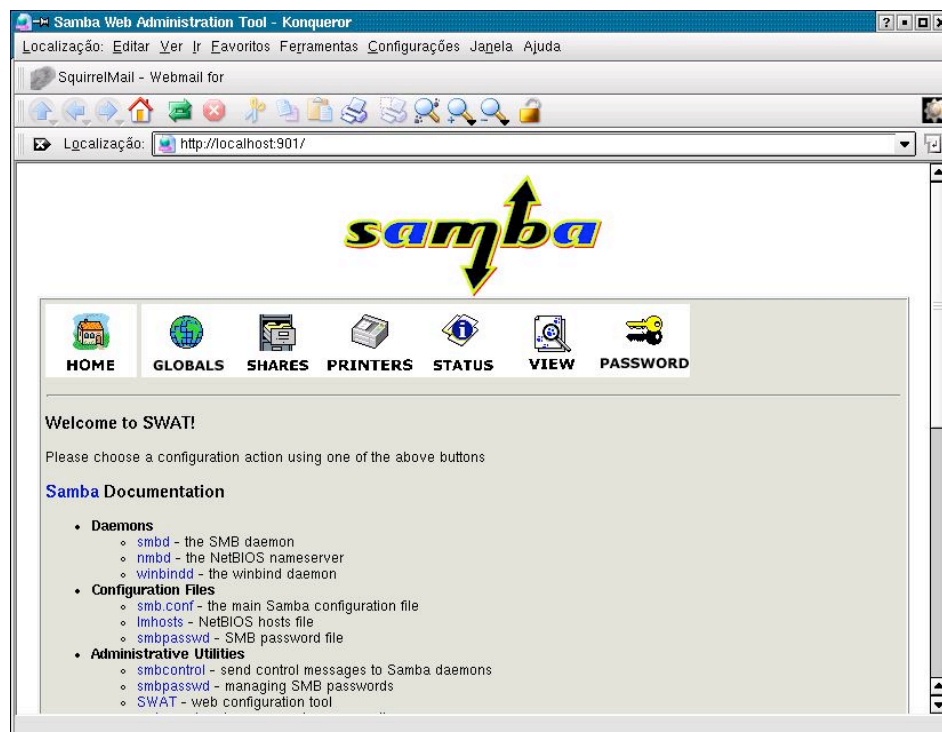


Figura 3.11: Interface do SWAT

A ferramenta traz em seus formulários dois (2) níveis de configuração, na tentativa de auxílio ao administrador.

- (a) *Básico*, nesse nível a ferramenta só apresenta ao administrador um formulário contendo os campos primordiais à configuração básica de um servidor *Samba*, assim a ferramenta tenta evitar que o administrador cause erros de configurações no serviço.
- (b) *Avançado*, neste nível a ferramenta apresenta praticamente todos os parâmetros aceitos no arquivo de configuração *smb.conf*, possibilitando ao administrador realizar uma configuração completa do *Samba*, inclusive a integração deste com outros serviços.

Outra característica que merece destaque no *SWAT*, é o seu sistema de ajuda, que

para cada campo do formulário a ferramenta disponibiliza um *link* para um pequeno texto contendo informações referentes aquele parâmetro, o que pode auxiliar o administrador em momentos de dúvida. Além disso, ainda é possível preencher o formulário com parâmetros padrões já disponível pela própria ferramenta para uma configuração simplificada e funcional. A Figura 3.12, mostra uma das interfaces da ferramenta com essas características.

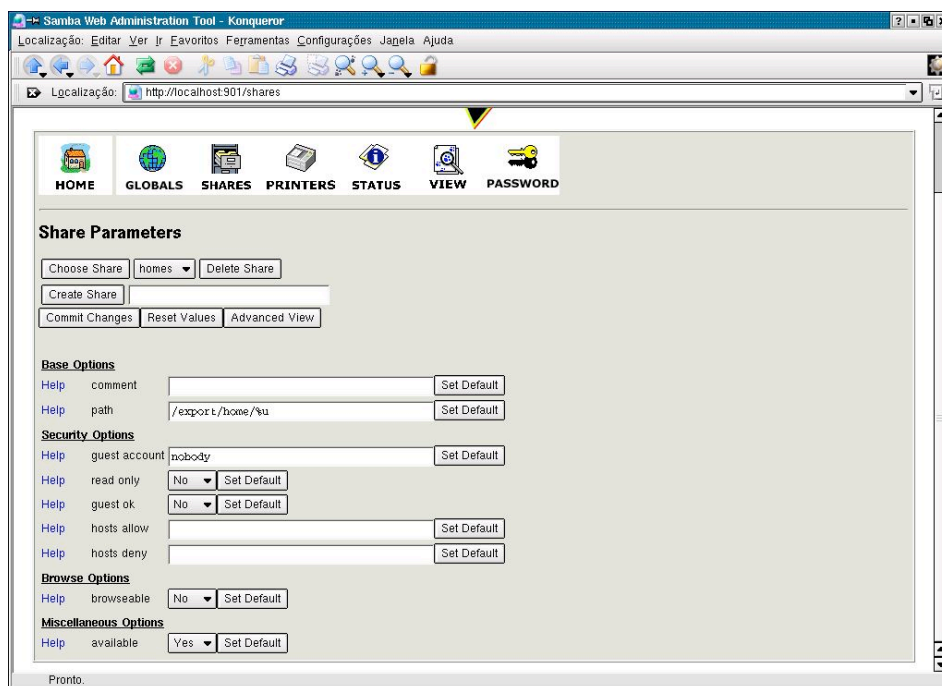


Figura 3.12: Interface de Formulários do SWAT

Apesar do *SWAT* ser uma ferramenta com características interessantes, ela não satisfaz a principal, ser uma ferramenta de administração e configuração de vários serviços e dispositivos, pois como já apresentado, o *SWAT* é uma ferramenta para a administração e configuração apenas do servidor *Samba*.

Entretanto, por suas características positivas, o *SWAT* é considerada uma ferramenta relevante quando se trata de administração/configuração remota baseada na WEB.

Além do *SWAT*, existem inúmeras outras ferramentas de acesso remoto baseados na WEB que realizam a configuração de serviços ou dispositivos de forma individual. Porém, essas ferramentas não fazem parte do escopo deste trabalho, visto que o foco são ferramentas que realizam administração/configuração de vários serviços ou dispositivos.

Entre as ferramentas de administração, baseadas em interface gráfica e de acesso remoto via WEB, é possível mencionar um conjunto principal, como será apresentado na sequência, iniciada pelo:

3. IBM Tivoli

O *Tivoli*, é um *toolkit* integrado de ferramentas proprietárias, desenvolvidas e mantidas pela *IBM*. Possui uma interface gráfica baseada na WEB, podendo ser acessada por um navegador de internet, entretanto, também possui outros tipos de interface como os de acesso local. Por ser uma ferramenta proprietária pouco se sabe sobre a sua arquitetura, mesmo assim não deve ser excluída da lista de principais ferramentas de administração computacional.

IBM Tivoli é um sistemas de gerência de plataformas da *IBM*, com o intuito de realizar administração, configuração e instalação de softwares remotamente, entre outras funcionalidades. Apesar destas definição o *Tivoli*, segundo a *IBM* ¹¹ é considerado e vendido como uma ferramenta para administrar sua infra-estrutura de *e-business*, buscando simplificar as operações de informática dentro das organizações que o adquirem.

Esta ferramenta se diferencia das outras justamente por este caráter de suporte a infra-estrutura de *e-business*, isso a torna uma ferramenta com um perfil empresarial e foco na gerência de negócio e não na configuração/administração de sistemas propriamente dito.

O *Tivoli* é desenvolvido de forma extremamente modular, possibilitando a aquisição e a utilização de alguns módulos individuais [39]. Apesar da possibilidade de utilização de módulos separados a integração de um módulo com outro é extremamente transparente, por esta característica, ainda pode ser enquadrado como uma ferramenta de administração computacional, pois existem módulos específicos para atividades mais técnicas como:

- **Gerenciamento de armazenamento:** O *software* de armazenamento, gerência e assegura a acessibilidade, disponibilidade e performance da informação armazenada.
- **Gerenciamento de sistemas:** O *software* de gerenciamento de sistema é utilizado para monitorar, controlar e otimizar os recursos de computação.
- **Gestão de ativos:** Proporciona maior eficiência na gestão de ativos, gerenciando todos os seus tipos de ativos em uma única plataforma.

¹¹<http://www-01.ibm.com/software/br/tivoli/>

- **IT Service Management:** Inovação, execução e liderança para que as empresas otimizem e controlem o negócio de *TI*.
- **Segurança:** O *software* de segurança é utilizado para proteger a confidencialidade, a integridade, a privacidade e a segurança dos sistemas.
- **Soluções de Provedores de Serviços:** Assegura que os serviços críticos estão executando aos padrões os mais elevados.

Para cada um destes serviço apresentados, existem um conjunto de módulos que o *Tivoli* disponibiliza para poder executar tal atividade.

É possível encontrar um pequeno relato sobre as vantagens e desvantagens de alguns do principais módulo do *IBM Tivoli* em *Systems Solutions* ¹².

A seguir é apresentado a Tabela 3.4 com os principais módulos que compõem cada serviço.

¹²<http://www.systemssolutions.com>

Gerenciamento de Armazenamento	Gerenciamento de sistemas	Gestão de ativos	IT Service management	Segurança	Solução de provedores de Serviço
Tivoli Continuous Data Protection	Tivoli IntelliWatch	Tivoli Capacity Process Manager	Tivoli Availability Process Manager	Tivoli Access Manager	Tivoli Netcool
Tivoli Storage Manager	Tivoli Monitoring	Tivoli Compliance Insight Manager	Tivoli Capacity Process Manager	Tivoli Directory Integrator	
Tivoli TotalStorage Productivity Center	Tivoli Configuration Manager	Tivoli Service Request Manager	Tivoli Release Process Manager	Tivoli Directory Server	
	Tivoli Provisioning Manager		Tivoli Storage Process Manager	Tivoli Identity Manager	
	Tivoli Remote Control		Tivoli Unified Process Composer	Tivoli Security Compliance Manager	

Tabela 3.4: Módulos de Alguns dos Serviços do IBM Tivoli

Apesar do *IBM Tivoli* ter se mostrado uma ferramenta bastante completa, ela possui algumas características não peculiares aos administradores computacionais. Além de ser uma ferramenta proprietária, possui um caráter de infra-estrutura de *e-business*. Isto foge do escopo da maioria dos administradores, que se preocupam exclusivamente com a administração e configuração dos recursos computacionais das organizações e não com os aspectos relacionados à gerência de negócios de *TI*. A Figura 3.13 ilustra a interface gráfica de um dos módulos do IBM Tivoli.

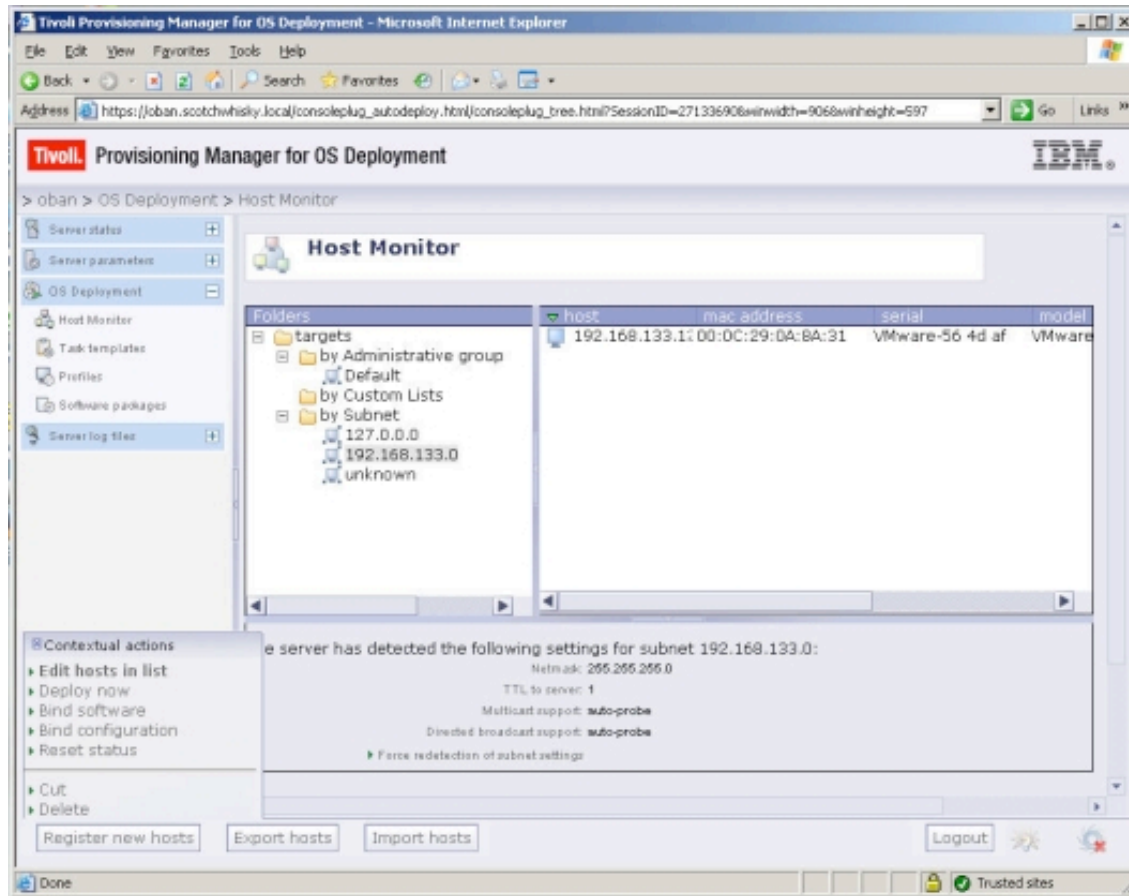


Figura 3.13: Interface do Tivoli

Existem outras ferramentas com mesmas características encontradas no *IBM Tivoli*, como o caráter de gerência de *e-business* e possibilidades de acesso remoto via WEB, é possível ilustrar a Tabela 3.5 como quadro demonstrativo de algumas destas ferramentas, que no entanto também são proprietárias.

Nome	Características
HP Open View	Uma suíte de aplicativos que permite a gestão da infra estrutura de TI. Inclui módulos operacionais responsáveis pela realizações de diversas tarefas de gestão de TI.
Suíte de gerenciamento de infra estrutura Unisys	Suíte de gerenciamento de infra estrutura de TI. Com foco em otimização de recurso. Baseados em módulos de : Governanças de serviços, Orquestração e fluxo de trabalho, Redefinição de funções, operações automáticas e utilização de recursos.
CA - Computer Associates	Conjunto de soluções específicas direcionadas a determinadas atividades como: Gerenciamento de Desempenho de Aplicativo, Gerenciamento do Banco de Dados, Governança, Gerenciamento de Operações e Infra-Estrutura, Gerenciamento de Ativos e Serviços de TI, Mainframe, Gerenciamento Financeiro, Portfólio e Projeto, Gerenciamento de Segurança, Armazenamento & Governança de Informação.

Tabela 3.5: Ferramentas com Foco em Gerência de *e-business*

Com o IBM *Tivoli* e as ferramentas mencionadas na Tabela 3.5, é possível evidenciar características semelhantes, sendo ambas consideradas ferramentas com foco em gestão de infra-estrutura de TI e *e-business*, tendo suas diretrizes pouco voltadas a administração/configuração computacional.

Outra desvantagem, é que por serem ferramentas proprietárias, não existe a possibilidade do administrador fazer alterações ou adequações em seu funcionamento de acordo com sua necessidade, ficando extremamente dependente do suporte técnico oferecido pelas empresas que fornecem a solução.

Existem algumas alternativas livres para ferramentas de administração baseadas na web, no entanto, essas, melhor se enquadram no perfil dos administradores computacionais, tendo como princípio a possibilidade de configuração de dispositivos e sistemas computacionais. Nesta linha pode-se apontar:

4. Red Hat Network (RHN) - Spacewalk

O *Red Hat Network* é uma ferramenta desenvolvida e mantida pela *Red Hat Corporation*, que teve seu código aberto para a comunidade software livre. Entretanto, como de costume nos produtos da *Red Hat*, o projeto recebeu outro nome e é concebido como *Spacewalk*. A ferramenta pode ser instalado em qualquer distribuição baseada em *Red Hat*, como *Fedora* e *CentOS* e seus derivados.

Segundo o projeto *RHN - Spacewalk*¹³, a ferramenta é considerada um gerenciador de conteúdo de software, com uma interface WEB amigável para gestão de servidores *Gnu/Linux*, ainda permite monitorar e aplicar as atualizações disponibilizadas pela *Red Hat* de forma simples e bem automatizada.

Funciona como um hospedeiro de *Kickstart*¹⁴, para facilitar a implementação de novos servidores, possibilita a verificação do *status* atual da máquina (Ligado ou Desligado), proporciona a possibilidade de gerir máquinas virtuais *XEN* e ainda oferece uma ferramenta de inventário de *software* e *hardware*.

O objetivo do *RHN-Spacewalk* é ser uma ferramenta ágil, simples e flexível. Para tanto, seu desenvolvimento e funcionamento é realizado de forma modular, baseando-se em quatro (4) principais módulos, que são apresentados na Tabela 3.6:

¹³<http://www.redhat.com/spacewalk/>

¹⁴Ao usar o *kickstart*, um administrador de sistemas pode criar um único arquivo contendo as respostas para todas as perguntas normalmente apresentadas durante um típico processo de instalação, isso possibilita a automatização de instalação do sistema

Módulos	Funções
<i>Update</i>	Permite que se administre e atualize facilmente o sistema com <i>patch</i> de segurança e pacotes de software diretamente da <i>Red Hat</i> , através de uma interface simples e pela WEB. Oferece informações completas e detalhadas de erratas, que o ajudam a compreender “como” e “por que” fazer a manutenção de suas máquinas
<i>Management</i>	Este módulo permite fazer um agrupamento de sistemas na tentativa de facilitar a administração, traz a possibilidade de agendamento de tarefas a serem executadas, além de se poder tratar e manipular permissão de sistema
<i>Provisioning</i>	Possibilita escrever <i>scripts</i> de configuração pelo <i>Kickstart</i> , ou o apontamento de um sistema existente e o <i>RHN-spacewalk</i> criará um <i>script</i> baseado neste. Permite que seu sistema regresse instantaneamente a um estado ou configuração anterior. Com ele você não precisará remover os pacotes manualmente, tentando lembrar as configurações pré-existent
<i>Monitoring</i>	Com este módulo é possível fazer o rastreamento do sistema, da rede ou da aplicação. Possibilita realizar <i>status</i> de monitoramento e relatórios de detecção

Tabela 3.6: Funções dos Módulos do RHN/Spacewalk

As Figuras 3.14 e 3.15, ilustram as interfaces gráficas dos dois (2) modelos da ferramenta, ou seja, o *RHN* e o *Spacewalk*. Como será visto, a interface gráfica de ambas as ferramentas são bem parecidas, diferenciadas apenas pelo esquema de cores e algumas funcionalidades que o *RHN* possui e que o *Spacewalk* não possui, em virtude da sua distribuição realizada pela *Red Hat Corporation*, como um produto comercial provido de suporte técnico. No entanto, a arquitetura de ambos os modelos seguem o mesmo padrão, o qual será detalhado seguidamente às ilustrações.

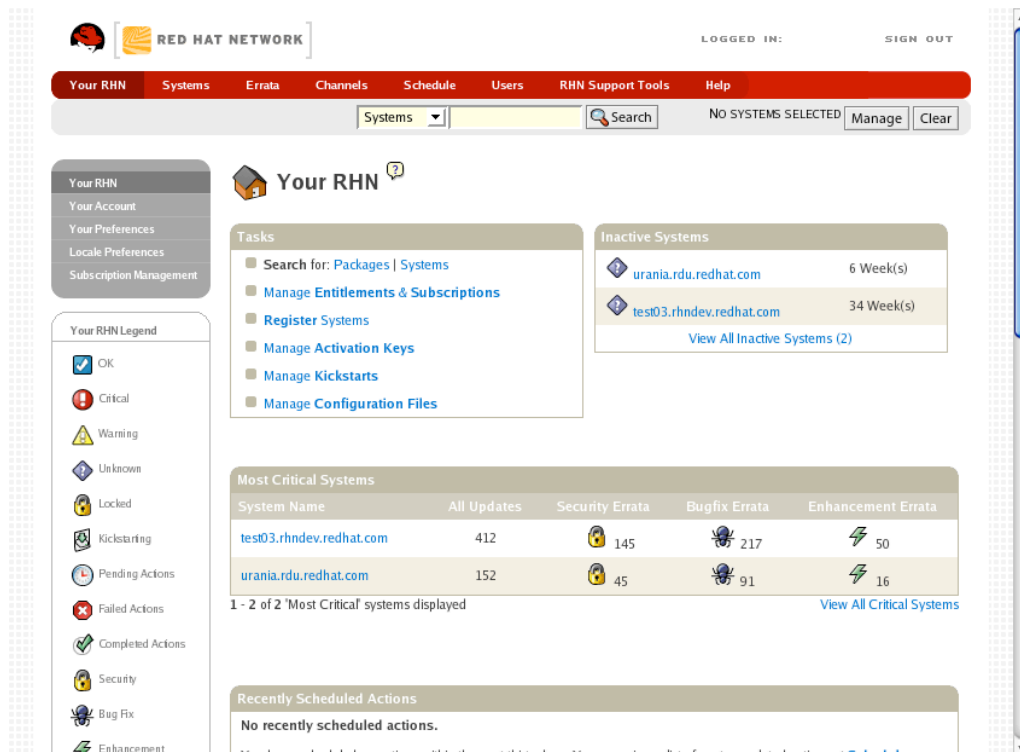


Figura 3.14: Interface do Red Hat Network

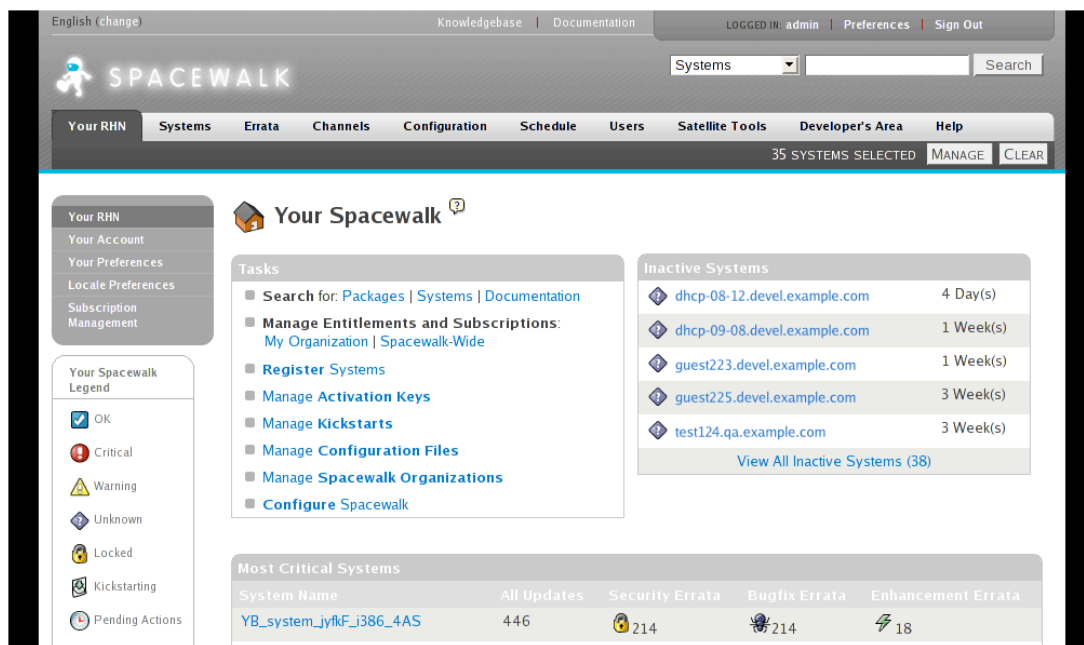


Figura 3.15: Interface do Spacewalk

a) Arquitetura do RHN-Spacewalk.

De acordo com a documentação disponibilizada para desenvolvedores no *site* do projeto¹⁵, o *RHN-spacewalk* é construído sobre uma arquitetura de três(3) camadas:

- (a) Camada de apresentação, baseadas em interface, que pode ser de: clientes de linhas de comando, clientes de *XML-RPC* e uma interface gráfica baseada na WEB;
- (b) Camada de Lógica de negocio, desenvolvida utilizando quatro (4) tipo de linguagens de programação: *Java*, *Python*, *Perl* e *PL/SQL*;
- (c) Camadas de dados, apoiada por uma base de dados *Oracle*. (Existem esforços para suporte a outras bases de dados.)

Na tentativa de esclarecer a arquitetura desta ferramenta, apresenta-se a Figura 3.16 encontrada também na documentação de desenvolvedores, ilustrando o relacionamento e funcionalidade de cada módulo da ferramenta:

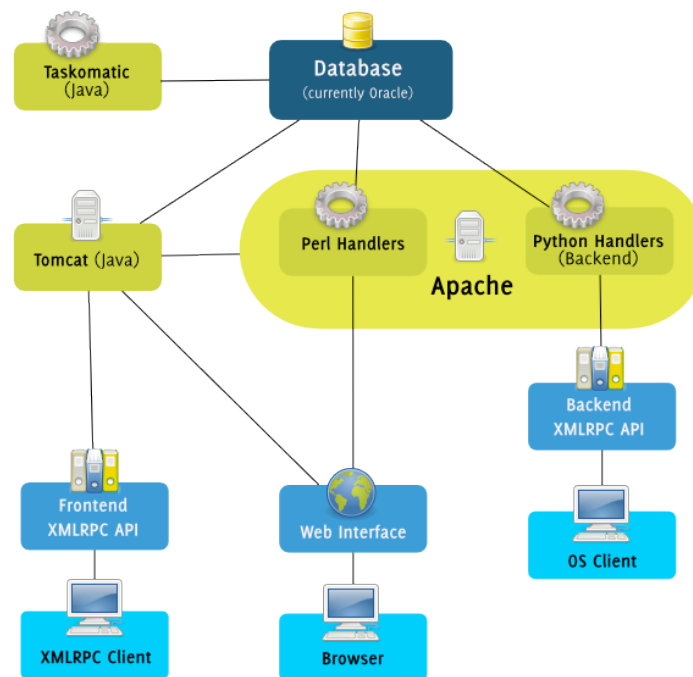


Figura 3.16: Arquitetura do Red Hat Network-Spacewalk

¹⁵<https://fedorahosted.org/spacewalk/wiki/Architecture>

A Figura 3.16, apresenta os módulos da ferramenta, os quais tem as seguintes funções:

- **Taskomatic** - Um *Daemon* responsável pela execução das tarefas as quais estão agendadas para serem executadas, seu desenvolvimento é sob tecnologias Java.
- **Database** - base de dados *Oracle* embutida para armazenar pacotes, perfis e informações de sistema.
- **Tomcat** - Servidor WEB para executar as aplicações *Java*.
- **Apache** - Servidor WEB com módulos de execução de *scripts CGI*, desenvolvidos em *Python* e *Perl*.
- **Fontend API XMLRPC** - Uma *API* para execução de chamadas vindas da interface do cliente, baseadas em *XML-RPC*.
- **Web Interface** - Uma interface gráfica de usuário baseada na WEB.
- **Backend API XMLRPC** - Uma *API* para execuções de clientes de linhas de comando.

Além da arquitetura similar entre os modelos é apresentado o modo de funcionamento de ambos, que só é diferenciado de acordo com as funcionalidades que o *RHN* possui além do *Spacwalk*, desta forma considera-se o modo de funcionamento apresentado equivalente para ambos.

b) Funcionamento do RHN-Spacewalk.

Baseado no esquema da arquitetura já apresentado, a ferramenta *RHN-Spacwalk* tem seu funcionamento sob dois (2) modelos de implementações diferentes. O modelo *Hosted* e o modelo *Satélite*.

Hosted

Neste modelo de funcionamento, cada sistema individual da organização se conecta diretamente ao *RHN-Spacewalk* pela internet, troca pacotes e informações utilizando uma interface gráfica WEB para facilitar a administração das operações. Acompanhe a Figura 3.17.

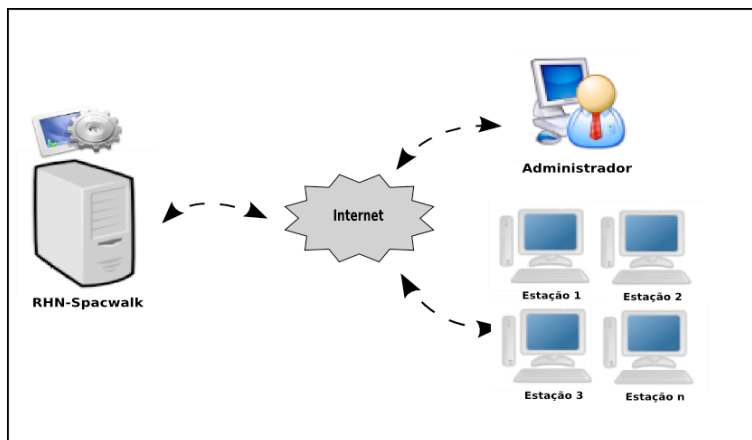


Figura 3.17: Funcionamento do RHN - Spacewalk no modelo Hosted

Satélite

Neste modelo, o servidor *Satélite* se conecta à *Red Hat* por meio da internet apenas para descarregar novos conteúdos e atualizações. É possível que o administrador coloque seus pacotes e soluções no servidor *Satélite* e posteriormente aplique essas soluções em suas máquinas sem a necessidade de uma nova conexão com a internet. Veja a Figura 3.18.

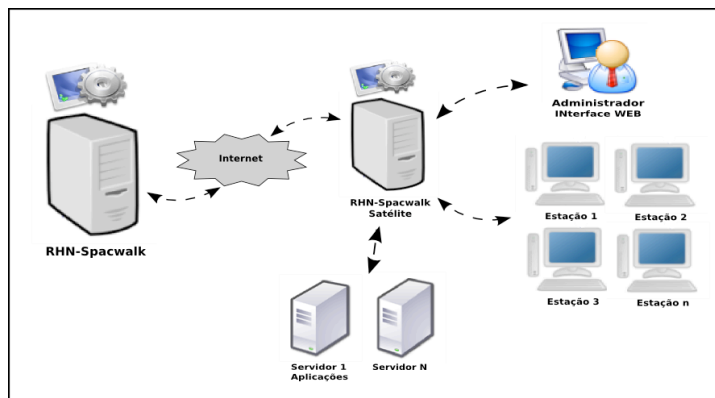


Figura 3.18: Funcionamento do RHN - Spacewalk no Modelo Satélite

Além dos modelos *Hosted* e *Satélite*, é possível a utilização do *RHN-Spacewalk* com soluções baseadas em *Proxy cachê*.

Satélite com Proxy

A solução baseada em *Proxy Cachê* serve para acelerar as transações entre o *RHN-Spacewalk Satélite* e as máquinas da rede, veja a Figura 3.19.

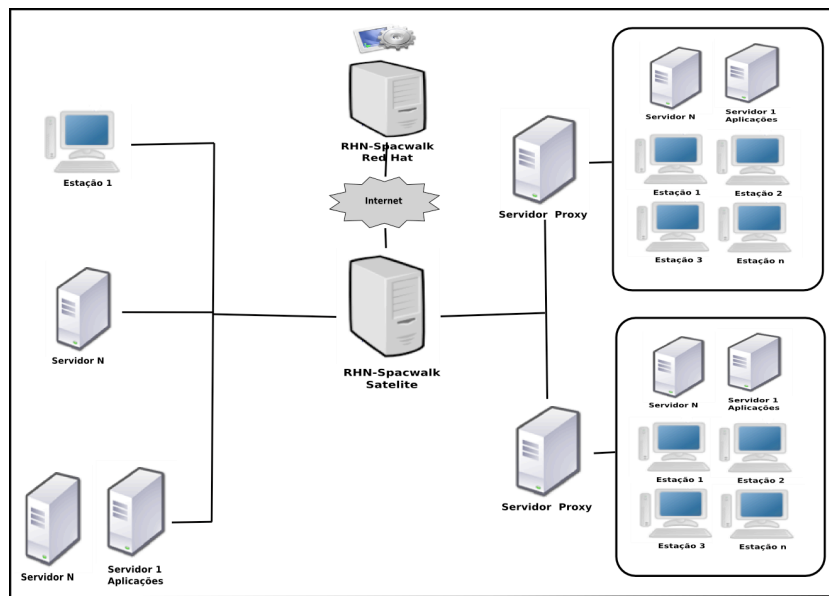


Figura 3.19: Funcionamento do RHN - Spacewalk no modelo Satélite com Proxy

Esta solução, utilizando o *Proxy* passa a ser uma alternativa para as organizações que possuem seus sistemas computacionais distribuídas em pontos geográficos distintos, como ilustrado na Figura 3.19.

Para maiores detalhes sobre modos de utilização do *RHN-Spacwalk*, recomenda-se uma leitura mais aprofundada da documentação disponibilizada no guia de referência¹⁶ da ferramenta.

Apesar do *RHN-Spacewalk* se apresentar como uma ferramenta muito completa, seu foco principal se concentra em torno da atualização de sistemas e pacotes de *software*. A ferramenta apresenta poucos recursos quando se trata de administração/configuração de sistema e serviços computacionais, se restringindo a possibilidade de agendamento de *scripts* de execução. Uma característica que pode ser colocada em destaque nesta ferramenta, é a possibilidade de criação de *cluster* entre máquinas com instalações distintas do *RHN-Spacewalk*, o que proporciona a administração do *cluster* como se estivesse em uma única máquina, centralizando todo o processo.

Com estas características apresentadas, o *RHN-Spacewalk* pode ser uma boa alternativa quando se necessita de atualizações automatizadas, no entanto, quando o foco é manipulação de configuração de serviços e dispositivos computacionais a ferramenta se apresenta insuficiente. Uma ferramenta com um perfil mais direcionado

¹⁶<https://rhn.linux.ncsu.edu/rhn/help/reference/index.jsp>

a administração/configuração computacional é o *Webmin*, o qual será estudado a seguir.

5. Webmin

O *Webmin* [40], é um software livre de administração/configuração de sistemas e redes de computadores, disponível sob a licença *BSD (Berkeley Software Distribution)*, que possibilita a configuração de sistemas com arquitetura *UNIX*. Esta ferramenta é utilizada através de uma *GUI (Graphical User Interface)* acessível por um navegador de internet, utilizando os protocolo *HTTP* ou *HTTPS* [17].

A administração/Configuração, é assistida pelo preenchimento de formulários que contem parâmetro relativos aos mencionados nos arquivos de configurações de cada serviços ou dispositivo que se deseja configurar. Assim, ao preencher um formulário está automaticamente editando o arquivo de configuração, que é alterado imediatamente após a submissão do formulário WEB.

O sistema foi desenvolvido por Jamie Cameron em 1997, logo começou a contar com o apoio da comunidade *Webmin*. Essa ferramenta vem sendo adotada por várias distribuições *Linux*, como ferramenta padrão de administração/configuração, eliminando a utilização de outras ferramentas de acesso local para a configuração de serviço/dispositivos de forma individualizada. A Figura 3.20, ilustra a interface gráfica acessível pela WEB, disponibilizada pelo *Webmin*.

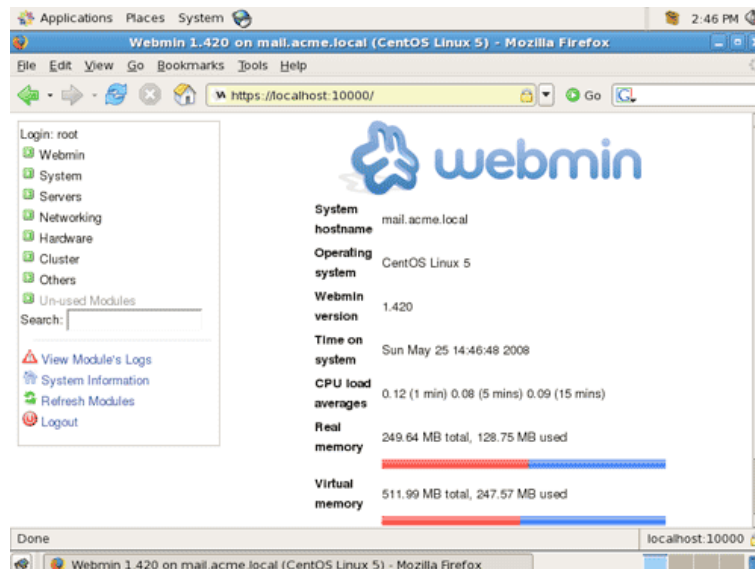


Figura 3.20: Interface do *Webmin*

Dentre as principais características do *Webmin* pode-se colocar em evidência a possibilidade de visualização e edição dos arquivos de configuração em formulários WEB, através de navegadores de internet, com uma interface mais agradável do que as linhas de comando dos consoles em modo texto. O que permite a instalação/configuração de determinados serviços com facilidade e garantia de diminuição de erros, em função de seu sistema de validação dos campos de formulários. Entretanto, é necessário salientar que as funções executadas pelo *Webmin* podem ser realizados por meio de máquinas remotas utilizando apenas um navegador de internet para acessar a ferramenta.

Em função das características apresentadas sobre o *Webmin*, considera-se até este momento ser uma das ferramenta que mais se aproxima das necessidades dos administradores, quando os quesitos são interface WEB, acesso remoto e quantidade de serviços possíveis de serem configurados pela ferramenta. Assim, a partir de então realiza-se uma análise mais detalhada sobre esta ferramenta com base nos trabalhos já realizados sobre essa tecnologia, como Antunes e Silva [17, 22].

a) Desenvolvimento

O *Webmin* é desenvolvido utilizando a linguagem de programação *Perl* e funciona por intermédio de um servidor WEB, criado pelo *script Miniserver.pl*, esse servidor no entanto funciona sob o protocolo *TCP* e utiliza a *porta 10.000* para comunicação[22]. Essa característica é o que possibilita o acesso remoto a ferramenta por meio de uma outra máquina utilizando apenas um navegador de internet.

O *Webmin* é desenvolvido de forma modularizada possibilitando a inserção e remoção de alguns módulos de configuração para serviços ou dispositivos em específico. Cada módulo também escrito em *Perl* é responsável pela configuração de um serviço individualizado. Esses módulos, são *scripts CGI (Common Gateway Interface)*, que tem como função gerar a página dinamicamente após a requisição feita pelo usuário. O *CGI* constrói essas páginas baseadas em parâmetro recebidos do formulários WEB, deste modo esses pequenos programas interpretam esses parâmetro e compõem as novas páginas depois de processá-los.

Apesar dos módulos serem desenvolvidos em *Perl*, é possível escrever novos módulos utilizando linguagens diferentes como *Shell scripts*, *PHP* e outras, graças a flexibilidade oferecida pelas funções principais escritas como *scripts CGI*[22].

b) Organização

Como apresentado, a ferramenta de administração/configuração *Webmin*, é desenvolvida em módulos esse módulos por sua vez possuem a mesma estrutura, mas funcionalidade

dades diferentes. Em uma instalação padrão do *Webmin*, os módulos e o resto do sistema se organizam sob dois (2) diretórios importantes contendo toda sua configuração. O primeiro é o `/usr/libexec/Webmin`, que pode ser considerado o diretório principal, neste encontram-se basicamente:

- `Miniserver.pl`: O próprio servidor *Webmin*, escrito em *Perl* e responsável pelo processamento dos pedidos dos clientes;
- `web-lib-funcs.pl`: Funções genéricas em *Perl* desenvolvidas para o *Webmin* e utilizada no programa;
- `*.pl`: *Scripts* desenvolvidos em *Perl* para a instalação e configuração do *Webmin*;
- `*.cgi`: Programas que processam os pedidos feitos ao servidor *Webmin* através de *CGI* escritos em *Perl*;
- `config-[PlataformaUNIX]`: Variáveis de ambiente utilizadas pelo servidor *Webmin* em cada plataforma *UNIX*.

Além destes arquivos, também existem diretórios específicos para cada módulo, que por sua vez, possuem uma estrutura bem definida e idêntica. Por exemplo, o módulo de administração/configuração de usuários, localizado no diretório `/usr/libexec/webmin/usermin/`, tem a seguinte estrutura:

- `/help/*`: Arquivos de ajuda para as opções disponibilizadas no módulo;
- `/images/*`: Diversas imagens referentes aos serviços que o módulo configura;
- `/lang/*`: Arquivos de tradução da interface gráfica para as diversas línguas suportadas pelo *Webmin*;
- `Config-[OS]`: Define para cada sistema operacional as características específicas de configuração do serviço;
- `Cron-lib.pl`: Diversas funções em *Perl* para a configuração e administração do serviço;
- `index.cgi`: Arquivo correspondente à página principal do módulo;
- `*.cgi`: *Scripts Perl* que efetuam as ações solicitadas pelo cliente ao servidor *Webmin*.

Assim, para cada serviço/dispositivo à ser configurado pelo *Webmin*, existe uma estrutura semelhante a essa.

O outro diretório de grande importância, é o `/etc/webmin/`. Dentro deste diretório existem vários outros subdiretórios para cada um dos serviços/aplicativos administrados/configurados pela ferramenta. Em cada subdiretório reside o arquivo `config` com a definição de variáveis de ambiente utilizadas pelo respectivo módulo.

c) Funcionamento

Do ponto de vista do funcionamento, o *Webmin* cria um servidor WEB através do *script miniserv.pl*, que recebe requisições na porta 10.000 sobre o protocolo *TCP*, assim, esse servidor recebe conexões vindas de um navegador de internet utilizando o protocolo *HTTP* ou *HTTPS*, isso porque pode utilizar *SSL*¹⁷. Depois da conexão estabelecida o administrador acessa o sistema como se estivesse acessando uma página de internet.

Nesta página, o administrador encontra um *menu*, que identifica as possibilidades de administração/configuração do *Webmin*, sendo:

- **Webmin:** Configurações do próprio *Webmin*, como usuários, idioma temas e outros.
- **System:** Configurações do sistema como log's, discos, sistemas de arquivos, etc.
- **Server:** Configurações de servidores como Apache, DNS, Postfix, etc.
- **Networking:** Configurações de rede como endereço, nomes, monitoramento da rede, etc.
- **Hardware:** Configurações de dispositivos como impressoras, volumes lógicos, etc.
- **Cluster:** Configuração e integração de um servidor *Webmin* com outro.
- **Outros :** Outras funções como agendamento de execução de scripts, verificação de status de serviços, etc.

Depois de selecionado o item desejado no *menu*, o administrador se depara com telas contendo formulários com botões de seleção, caixa de texto, entre outro, que são gerados pelos módulos específicos de cada serviço ou dispositivo. Veja a Figura 3.21 com um modelo de formulário gerado pelo *Webmin*, por meio dos *scripts CGI*, para a configuração de um servidor WEB.

¹⁷SSL- (Secure Sockets Layer) Protocolo criptográfico que provêem comunicação segura na Internet para serviços como navegação por páginas (HTTP) e outros.

Figura 3.21: Formulário de Configuração do Servidor WEB pelo *Webmin*

Esses formulário, são gerados por *scripts CGI*, que logo após seu preenchimento passam por um processo de validação dos campos, caso algum campo esteja preenchido inadequadamente, mensagens de erro são retornadas, mas se estiver tudo adequado, são executados os *scripts* e mensagens de sucesso são retornadas.

O esquema apresentado na Figura 3.22 ilustra o funcionamento do *Webmin*.

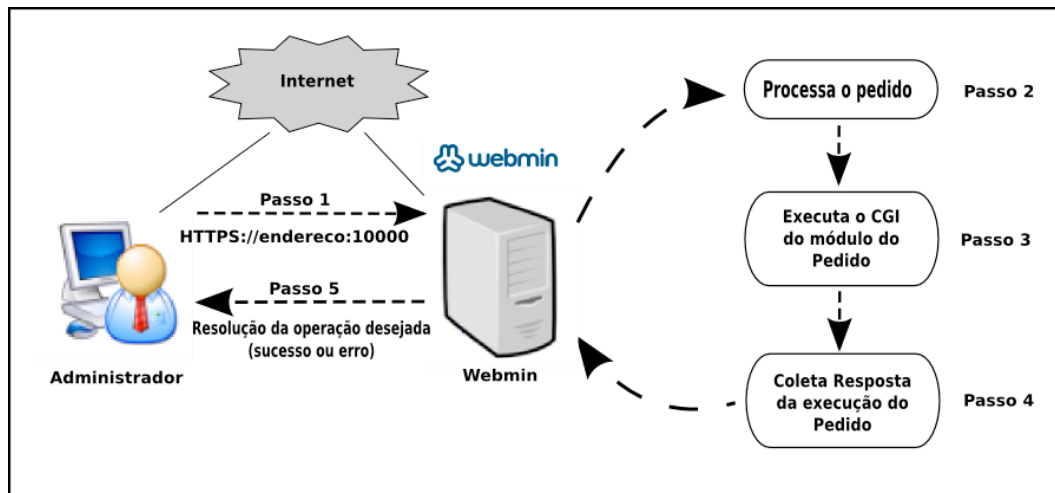


Figura 3.22: Esquema de Funcionamento do *Webmin*

Basicamente o que acontece no esquema é:

- No passo 1: O Administrador acessa a ferramenta remotamente ou mesmo localmente através de um navegador de internet pelo endereço do *Webmin* e sua porta de comunicação. Depois escolhe qual o serviço ou dispositivo que deseja configurar.
- No passo 2: O servidor *Webmin* seleciona o módulo correspondente ao serviço ou dispositivo solicitado pelo passo 1 e apresenta o formulário WEB referido.
- No passo 3: Os parâmetros dos campos preenchido no formulário são validados e com os valores recebidos dos formulários o módulo selecionado pelo passo 2 é executado.
- No passo 4: O resultado da execução do passo 3 é coletado e armazenado.
- No passo 5: É enviado para o navegador de internet do administrador os dados coletados pelo passo 4. Então, o administrador decide o que deve ser feito. Se algo de errado aconteceu ele pode retomar o processo novamente.

Vantagens e Desvantagens do *Webmin*.

Pelos aspectos apresentado, fica claro e evidente que essa ferramenta tem um potencial enorme. Pelas características relacionadas a possibilidade de acesso remoto, interface gráfica e tratamento de um número considerável de serviços e dispositivos computacionais, o *Webmin* se posiciona a frente das outras ferramentas com mesmo enfoque. Essa ferramenta sem sombra de duvida é considerado por muitos administradores um dos melhores sistemas de administração/configuração computacional. No entanto, apesar de sua alta popularidade entre os administrados ela também possui características negativas que acabam por se tornarem desvantagens em relação as outras. A seguir é apresentado as Tabelas 3.7 e 3.8 demonstrada por Silva [22], que relatam algumas das principais vantagens e desvantagens dessa ferramenta de administração/configuração computacional.

Vantagens	Devido a:
Portabilidade	Ferramenta está escrita em <i>Perl</i> .
Modularidade	As funcionalidades da ferramenta são desenvolvidas por módulos independentes entre si.
Heterogeneidade	A ferramenta pode se instalada em diversos sistemas baseados em <i>UNIX</i> e também existe uma versão para <i>Windows</i> .
Proteção contra erro de sintaxe	O código dos arquivos de configuração são apresentados em formulários mais organizados e em menor quantidade o que diminui a chance de erros
Integridade/coerência/consistência dos arquivos de configuração	Usa os mesmos arquivos de configurações das configurações manuais feitas pelos administradores garantido a consistência dos arquivos
Segurança	Possui autenticação de usuários para a utilização da ferramenta. Possibilidade de criação de perfis de usuários administrados e utilização de SSL para conexões HTTP seguras (HTTPS) entre um navegador de internet e a ferramenta.
Acesso remoto	Permite configurar servidores remotamente onde se encontra instalada a ferramenta.
Gratuidade	A ferramenta é um software livre sob licença BSD.
Centralização de configuração	É possível a criação de cluster entre servidores <i>Webmin</i> instalados remotamente para a centralização de configuração.

Tabela 3.7: Vantagens do *Webmin*

Desvantagens	Devido a:
Criação e inserção de novos módulos	Não seguem um padrão de desenvolvimento dificultando o entendimento e consequentemente o desenvolvimento de novos módulos para serviços diferentes dos já proposto pela ferramenta.
Carregamento de todos os módulos	A ferramenta instala todos o módulos, sendo desnecessário a presença de módulos que o administrador não necessita
Interface não organizada	Apesar da possibilidade de mudança de temas a interface do <i>Webmin</i> principalmente nos formulários referentes ao serviços não possuem uma padronização na sua apresentação, o que em alguns casos atrapalha o administrador
Formulários estáticos	O conteúdo de cada formulário é estático e pré-definido, assim não é possível a inserção de um novo parâmetro no arquivo de configuração, referente a uma função específica.
Administração Centralizada	Para fazer a administração de mais de uma máquina é necessário fazer a instalação de um sistema <i>Webmin</i> completo em cada máquina e criar um cluster entre eles.
Cluster	São formados por <i>Webmin's</i> instalados em máquinas distintas que se comunicam por meio de um protocolo utilizando RPC, o que dificulta o desenvolvimento de <i>Firewalls</i> , pois as comunicação sob <i>RPC</i> não possuem portas definidas de comunicação.

Tabela 3.8: Desvantagens do *Webmin*

Apesar de algumas características negativas, o *Webmin* ainda se apresenta como a ferramenta de administração/configuração mais madura atualmente, sendo amplamente utilizada entre os administradores, suportada por quase todas as distribuições *Linux* como Debian, Gentoo, Red Hat, Slackware, SuSe e suas respectivas descendentes. Isso porque além de suas vantagens já apresentada na Tabela 3.7, a ferramenta possui um caráter extremamente técnico e ao mesmo tempo bem simplificado, com foco na administração/configuração de serviços e dispositivos computacionais, diferentemente das outras ferramentas de acesso remoto via WEB examinadas neste trabalho.

Uma breve comparação, entre o *Webmin* e as outras ferramentas de administração de acesso remoto via WEB, é apresentada na sequência pela Tabela 3.9.

Caract./Ferram.	<i>Webmin</i>:	IBM Tivoli	RHN/Spacewalk
Foco da ferramenta	Administração Configuração	Administração e Gerência de Negócio	Atualização e gerenci- amento de Conteúdo de Software
Desenvolvimento mo- dular	Sim	Sim	Sim
Inserção de módulos novos	Parcial	Não	Parcial
Possibilita instalação e atualização de pacotes	Parcial	Sim	Sim
Possibilita o agenda- mento e execução de <i>scripts</i>	Sim	Parcial	Parcial
Autenticação de usuários	Sim	Sim	Sim
Centralização da con- figuração de mais de uma máquina	Parcial	Sim	Sim
Necessidade de clien- tes para configuração de máquinas remotas	Sim	Sim	Sim
Simplicidade na uti- lização	Sim	Não	Parcial
Software Livre	Sim	Não	Sim

Tabela 3.9: Características das Ferramentas de Acesso Remoto

Os tópicos considerados “Parcial” foram assim classificados por não atenderem completamente os respectivos requisitos mas, algum tipo de iniciativa já foi tomada no sentido de satisfazer essa premissa.

Além da comparação realizada entre as ferramentas de administração computacional de acesso remoto via WEB, a seção seguinte deste trabalho se destina a apresentar uma outra comparação, desta vez, entre as principais ferramentas de cada classificação aqui definida.

3.2 Comparação entres as Ferramentas de Administração Computacional

Para a realização de uma comparação entre as características das ferramentas apresentadas até este momento, utiliza-se a Tabela 3.10, como fonte demonstrativa dos pontos considerados relevantes para a administração/configuração computacional, encontrada em alguns destes softwares. Utiliza-se como base de comparação as ferramentas que foram consideradas as mais relevantes entres o mesmo perfil de classificação. São elas:

- Ferramentas de administração baseada em Interface Modo Texto: **Cfengine**;
- Ferramentas de administração baseada em Interface Gráfica de Acesso Local: **YaST**;
- Ferramentas de administração baseada em Interface Gráfica de Acesso Remoto: **Webmin**.

A Tabela 3.10, apresenta a comparação entre algumas características dessas ferramentas:

Caract./Ferram.	Cfengine	YaST	Webmin
Tipo de Licença	<i>GPL</i>	<i>GPL</i>	<i>BSD</i>
Linguagem desenvolvida	<i>C</i>	<i>C++</i>	<i>Perl</i>
Linguagem específica de configuração ou utilização	Sim	Não	Não
Necessidade de configuração prévia	Sim	Não	Parcial
Possibilita execução de <i>scripts</i>	Sim	Não	Sim
Possibilita configuração remota	Sim	Não	Parcial
Possui <i>Daemon</i> cliente e servidor para configuração remota	Sim	Não	Sim
Protocolo de comunicação	<i>RPC</i> (entre cliente e servidor)	Não	<i>HTTP/HTTP's</i> (entre navegador e ferramenta) <i>RPC no Cluster de Webmin's</i>
Possui repositório central de configuração	Não	Não	Não
Usa Criptografia	Sim	Não	Sim
Possui autenticação de usuários	Não	Sim	Sim
Construção modular	Não	Sim	Sim
Possui Módulo por serviço	Não	Sim	Sim
Possível inserir novos módulos	Não	Sim	Parcial
Complexidade de desenvolvimento de novos módulos	Pouca	Muita	Muita
Possibilidade de carregamento somente de módulos necessários	Sim	Não	Não
Interface amigável	Ruim	Ótima	Boa
Edição de arquivos de configuração	Manual	Por formulários	Por formulários
Portabilidade de plataformas Unix	Sim	Parcial	Sim
Possibilita instalação e atualização de pacotes	Sim	Sim	Sim
Possibilita o agendamento e execução de <i>scripts</i>	Sim	Não	Sim
Limitação dos serviços ou dispositivos que administra ou configura	Numero bem restrito	Numero considerável	Numero bem relevante
Ano da primeira versão	1993	1996	1997

Tabela 3.10: Características das Principais Ferramentas de Administração

Novamente vale ressaltar que os tópicos considerados “Parcial” foram assim classificados por não atenderem completamente os respectivos requisitos mas, algum tipo de iniciativa já foi tomada no sentido de satisfazer essa premissa.

Existem várias outras ferramentas de administração/configuração baseadas em ambiente gráfico, tanto de acesso local como de acesso remoto que realizam configurações de serviços isolados e/ou específicos como configuração de interface de rede, criação de usuários, gerenciamento de base de dados, etc. Entretanto, todos esses modelos de ferramentas de administração só são capazes de manipular e configurar serviços nas máquinas onde estão instaladas, não podendo configurar serviços ou aplicações em máquinas remotas à sua instalação, a não ser com a instalação de programas clientes nas máquinas remotas, o que não soluciona e nem diminui os problemas dos administradores, pois também passarão a ter que administrar não somente os aplicativos e servidores como também os clientes destes softwares instalados nas máquinas a serem configuradas.

O objetivo deste trabalho é projetar uma ferramenta de administração computacional que possa utilizar as melhores características das ferramentas estudadas, mas com três (3) grandes enfoques não encontrados em nenhuma dessas:

1. A possibilidade de acesso remoto e configuração remota sem a necessidade de utilização de *Daemons* ou agentes clientes instalados na máquinas remotas a serem configuradas por uma central de administração. Utilizando para isso apenas pacotes comuns a uma instalação típica de um sistema *Linux*.
2. A possibilidade de criação de uma ferramenta que contenha um repositório permanente de arquivos de configuração já pré-configurados pelos usuários, para que caso haja a necessidade de uma reinstalação do sistema por qualquer que seja o motivo, em poucos minutos é possível recuperá-la com a aquisição da configuração adequada, contida nos arquivos armazenados no repositório central.
3. A possibilidade de administração e configuração de qualquer serviço ou dispositivo que tenha seu funcionamento baseado em arquivo de configuração do tipo texto plano, isso através da criação de módulos específicos para tais serviços e aplicativos e posteriormente os incorporando de forma transparente na ferramenta.

3.3 Conclusão

Neste capítulo foi apresentado os aspectos relacionados as ferramentas de administração computacional, para tanto realizou-se uma classificação destas ferramentas de acordo com a sua interface de utilização e perfil de funcionamento. Foi apresentado uma breve

descrição das principais linguagens de programação baseadas em *scripts*, utilizadas pelos administradores na hora da construção de ferramentas caseiras, para automação de suas tarefas cotidianas, técnica que se vê muito presente ainda entre os administradores computacionais.

Foi relatado um estudo sobre algumas ferramentas de administração sem interface gráfica baseadas em comandos em modo texto, no entanto, apresentou-se uma breve descrição de cada ferramenta e seguidamente uma comparação entre as características e peculiares dessas ferramentas.

Também neste capítulo, foi apresentado um estudo sobre ferramentas de administração que utilizam interface gráfica, entretanto, essas ferramentas foram subdivididas em dois (2) grupos, as ferramentas de interface gráfica de acesso local e as ferramentas de interface gráfica de acesso remoto. Da forma como feito entre as ferramentas baseadas em interface em modo texto, tem-se uma breve descrição seguido de uma comparação entre as ferramentas gráficas de acesso local, o que também ocorreu com as ferramentas gráficas de acesso remoto. Porém, houve um esforço maior sobre a descrição da ferramenta de acesso remoto *Webmin* por ter se mostrado a ferramenta de administração mais madura e popular entre as estudadas.

Finalizando o capítulo, foi apresentado um comparativo entre as ferramentas de administração que mais se sobressaíram em suas categorias, desta forma, foi possível encontrar algumas características comuns e essenciais as ferramentas de administração computacional, como também, características peculiares a somente algumas delas. Neste sentido, foi possível pensar em um modelo de ferramenta que contemple todas essas características importantes assim como outras que não foram encontradas em nenhuma delas.

Esse capítulo, serviu de base para composição das outras etapas deste trabalho principalmente para a composição da proposta de um novo modelo de ferramenta de administração computacional, a qual será descrito na sequência.

Capítulo 4

Proposta de um Modelo de Ferramenta

Este capítulo apresenta a proposta de um modelo de ferramenta para a administração computacional, seguido dos aspectos relacionados à arquitetura, funcionamento, segurança e organização do modelo. Ele descreve algumas das características do modelo, assim como peculiaridades referentes a sua construção modular e funcionamento independente de cada módulo. O capítulo segue com uma análise de alguns aspectos de segurança envolvida e proporcionada pelo modelo, além da apresentação de um protótipo desenvolvido para a validação do modelo proposto, onde serão ilustrados alguns detalhes de implementação, bem como as tecnologias utilizadas na construção de cada módulo. Finalizando são apresentados a estrutura organizacional do modelo e alguns resultados de processamentos realizados pelos módulos do protótipo desenvolvido.

4.1 O Modelo de Ferramenta

A partir do estudo realizado sobre o conjunto das principais ferramentas de administração computacional, foi possível evidenciar características importantes nos cenários atuais envolvendo essa temática e que não são atendidas completamente ou parcialmente pelas ferramentas estudadas. Essa situação justifica a proposta de um novo modelo de ferramenta, a qual satisfaça as características consideradas mais relevantes. Entre elas é possível mencionar:

- *Ferramenta de Administração e Configuração de um amplo conjunto de serviços ou dispositivos, que tenham sua configuração implementada por arquivos textos.* Do ponto de vista da administração, é importante a utilização de ferramentas que sejam capazes de realizar a configuração/administração de um conjunto amplo de

serviços ou dispositivos, o que evita a sobrecarga de ferramentas instaladas nos sistemas e garante uma centralização da administração/configuração. Existem ferramentas que atendem parcialmente esse requisito, com um número vasto, porém limitado de serviços ou dispositivos a serem administrados/configurados, principalmente pela dificuldade de tratamento da diversidade dos arquivos de configuração desses serviços/dispositivos.

- *Ferramenta de Administração Computacional flexível para construção e inserção de novos módulos responsável pela configuração de serviços ou dispositivos específicos.* É cada vez mais comum a necessidade de adequações de ferramentas de administração para atuação em cenários específicos. É difícil encontrar ferramentas que atendam as necessidades específica de cada organização, neste sentido os administradores acabam construindo ferramentas caseiras para a realização de tarefas não atendidas pelas ferramentas adotadas. Como possível solução a essa problemática, existem ferramentas de administração computacional que possibilitam a criação e inserção de novos módulos para a realização de configurações específica, entretanto, o que se constatou com o estudo de ferramentas com esse perfil, é que a construção e a integração desses novos módulos são complexos e nem sempre tem seu funcionamento garantido.
- *Ferramenta de Administração baseada em interface gráfica e de acesso remoto.* Para auxiliar o trabalho exaustivo da administração e tentando evitar muitas falhas de operação de sistemas e serviços computacionais causadas por edições manuais de arquivos de configurações, as ferramentas de interface gráfica se apresentam como melhores soluções a essas situações. No entanto, ferramentas com interface gráfica que possibilitam o acesso remoto são poucas e as que satisfazem esse requisito utilizam interfaces complexas e mau elaboradas, dificultando sua utilização por parte dos administradores, que necessitam de ferramentas simples e objetivas em suas interfaces.
- *Ferramenta de Administração que realize configuração remota sem a necessidade de instalação de aplicações ou agentes clientes nas máquinas a serem configuradas.* Ferramentas que possibilitem ao administrador uma configuração em máquinas remotas, onde sua presença física se torna inviável, vêm se tornando muito pertinente. É cada vez mais presente a necessidade de configurações de servidores e estações de trabalhos em ambientes geograficamente separados da localização física do administrador. Ferramentas que tentam atender esse requisito, em sua maioria, utilizam outras aplicações clientes instaladas nas máquinas remotas, ou ainda usam programas *agentes* que ficam retornando informações periodicamente para a ferramenta centralizadora, local onde se encontra fisicamente o administrador. Essa solução

pode ser considerada inadequada em função do grande fluxo de informação trafegado nos links de dados, sobrecarregando a rede e aumentando o número de aplicações que o administrador deverá manter. Considera-se essencial, uma ferramenta que utilize apenas os recursos disponíveis por uma instalação típica de uma distribuição *Linux*.

- *Ferramenta de Administração que realize instalação remota de pacotes de software.* Assim como as ferramentas que possuem o acesso e configuração remota para evitar a presença física do administrador na máquina a qual se configura, julga-se importante o requisito de instalação de um pacote de software de forma também remota. Isso garante a manipulação necessária para a administração em máquinas geograficamente dispersas da localização física do administrador. Ferramentas que atendem esse requisito o fazem a partir da distribuição do pacote de software de um repositório local, ou por meio de aquisição dos pacotes via internet.
- *Ferramenta de Administração que utilize repositório de configuração para possível restauração ou distribuição em massa de configurações.* Devido ao grande número de máquinas a serem administradas em um parque computacional, é de suma importância a utilização de ferramentas que possibilite a configuração em massa de equipamentos que tenham seu funcionamento idêntico. Além disso, a possibilidade de armazenamento de arquivos de configurações dos serviços desejados é outra característica relevante para novos modelos de ferramentas de administração. Desta forma, é possível a utilização de um repositório central para a distribuição de configurações para as máquinas que vierem fazer parte do parque computacional, bem como, a restauração de configurações existentes, que por algum motivo passaram a não ser mais executadas de forma adequada. Com um repositório de arquivos de configuração o administrador poderá armazenar os arquivos já configurados de acordo com sua necessidade, e posteriormente distribuí-los de forma individualizada ou em *Lote*. Assim, caso uma de suas máquinas fique inoperante por qualquer que seja o motivo, é possível uma restauração imediata a partir de uma instalação típica do sistema operacional e das aplicações necessárias, e posteriormente a aquisição dos arquivos de configuração adequado ao antigo funcionamento. Entre as principais ferramentas de administração estudadas, não foi encontrado nenhum modelo que atendesse esse requisito parcial ou integralmente da forma como proposto.
- *Ferramenta de Administração que mantenha o estado da sessão do administrador.* A partir do momento em que o administrador possa estar localizado em qualquer lugar do mundo e utilizando qualquer computador para acessar a ferramenta de administração, passa a ser extremamente pertinente um modelo de ferramenta que

possa armazenar o estado do sistema quando o administrador realiza uma configuração de determinado serviço/dispositivo. Desta forma, se uma conexão entre a máquina do administrado e a máquina da ferramenta for interrompida por qualquer que seja o motivo, é possível o armazenamento daquele estado para que da próxima vez em que o administrador acessar a ferramenta ele possa encontrar o sistema no mesmo ponto em que fora interrompido. Com isso o administrador poderá iniciar a configuração de um determinado serviço estando localizado em uma máquina e finalizar essa configuração estando em um outra.

- *Segurança.* A segurança sempre deve estar presente na administração computacional, é importante para os administradores a utilização de ferramentas que garantam o mínimo de segurança em seu funcionamento e nas configurações de seus serviços. Como essa característica é de fundamental importância, foi dedicada uma seção exclusiva para o tratamento deste assunto, tendo em vista a falta de apreço por algumas das ferramentas de administração estudadas em relação a esse requisito.

De acordo com a classificação das ferramentas de administração computacional realizada no capítulo anterior, apresenta-se a Tabela 4.1, com os requisitos considerados importantes que são ou não atendidos pelas mesmas.

Requisito/Ferramentas.	Cfengine	YaST	Webmin
Configuração de um vasto conjunto de serviços ou dispositivos	Sim/Limitado	Sim/Limitado	Sim/Limitado
Construção e inserção de novos módulos	Não	Sim/Limitado	Sim/Limitado
Baseada em interface gráfica e de acesso remoto	Não	Não	Sim
Configuração remota sem a necessidade de aplicações ou agentes cliente	Não	Não	Não
Instalação remota de pacotes de software	Não	Não	Não
Possui repositório de configuração	Não	Não	Não
Mantém o estado da sessão do administrador	Não	Não	Não
Nível de Segurança	Fraco	Fraco	Fraco

Tabela 4.1: Requisitos Atendidos ou não pelas Ferramentas de Administração

No intuito de projetar uma ferramenta de administração computacional completa e consistente, descreve-se a partir de então um modelo baseado em uma proposta de arquitetura e funcionamento. Para tanto, utiliza-se como base as principais características encontradas nas ferramentas analisadas no *capítulo 3*, assim como requisitos considerados importantes, e que não são atendidos de forma parcial ou integral pelas referidas ferramentas analisadas.

4.2 Arquitetura do Modelo

O modelo aqui proposto de uma ferramenta de administração computacional, prevê seu desenvolvimento realizado e organizado em camadas ou módulos. Essas camadas/módulos são totalmente autônomas, desta forma, pretende-se conseguir uma independência entre cada funcionalidade desempenhada pela ferramenta, além da possibilidade de flexibilidade no momento do desenvolvimento, não se atendo a homogeneidade de linguagens de programação e nem de tecnologias a serem utilizadas.

Com esse tipo de arquitetura, espera-se ter pontos delimitados que podem ser utilizados no momento que for necessário adequações à ferramenta. A arquitetura do modelo, bem como suas camadas/módulos seguem de acordo com o diagrama apresentado na Figura 4.1.

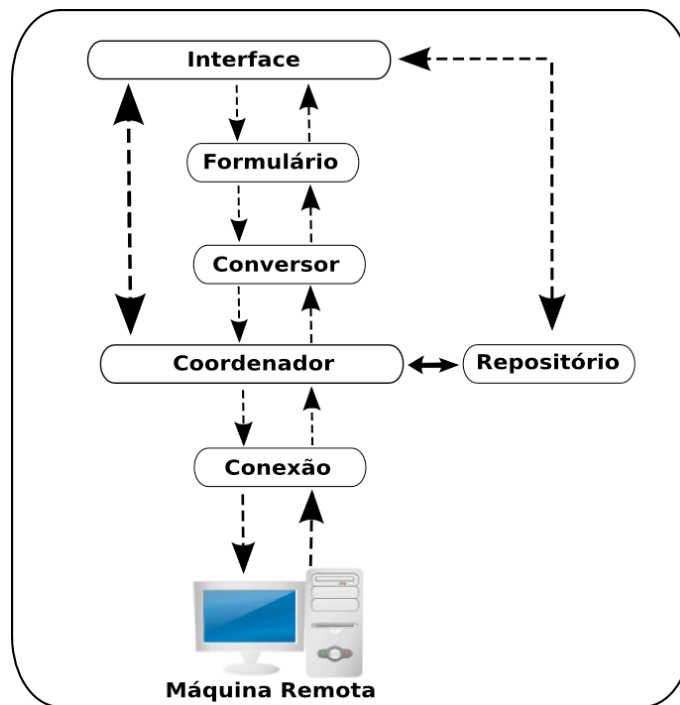


Figura 4.1: Arquitetura do Modelo Proposto

Assim como ilustra o diagrama, cada camada do modelo é responsável por funções específicas dentro do processo, e tem seu funcionamento totalmente independente. O que é relevante nessa arquitetura é a submissão de resultados de uma camada inferior para a outra superior. A ferramenta está projetada para ter seis (6) Módulos, os quais serão descritos na sequência.

Interface

Como o objetivo da ferramenta é ser utilizada através de um navegador de internet em uma máquina remota, o módulo de *interface* deve ser desenvolvido utilizando tecnologias voltadas para WEB, como as páginas de internet. Nessa camada se prevê parte da inteligência do sistema, como por exemplo:

- Definição de *Layouts* de interface: Possibilidade de definição de cores e aplicação de temas específicos a cada organização;
- Exibição pré definida de parâmetros nos formulários: Exibição de parâmetros de acordo com o nível (iniciante, intermediário e avançado) de conhecimento do administrador.
- Validação de campos de formulários: Uma tentativa de evitar erros de preenchimento dos campos do formulário, referentes aos parâmetros de configuração;
- Armazenamento do estado da sessão do administrador: Possibilidade da conexão ser interrompida e reiniciada posteriormente, encontrando o sistema de configuração no mesmo estado em que fora interrompido.

Para tanto, essa camada deverá ter suas funcionalidades independentes das outras, passando a ser executada através da interação com o usuário, fazendo chamadas às camadas inferiores, recebendo e apresentando como resultado dados já processados pelas camadas mais inferiores.

Formulário

O módulo *Formulário* é o responsável pela geração dos formulários WEB, contendo os parâmetros presentes nos arquivos de configuração de cada dispositivo/serviço a ser configurado pela ferramenta. Esses formulários são os meios por onde o administrador edita ou preenche os valores referente a configuração desejada. Esse módulo pode ser desenvolvido utilizando qualquer tecnologia, pois tem somente a função de gerar formulários WEB a partir de parâmetros recebidos das camadas *Conversor*.

Conversor

O módulo *Conversor* é responsável por um tratamento mais refinado nos arquivos de configuração de cada serviço/dispositivo a ser configurado. Essa camada recebe os arquivos de configuração pré-tratados e refina este tratamento, verificando cada parâmetro dos arquivos de configuração e repassando-os para a camada seqüente (formulário) para a geração do formulário WEB do referido serviço/dispositivo.

Coordenador

O módulo *Coordenador* é considerado o núcleo da ferramenta, é o responsável pela interconexão de todos os módulos e deve manter o sincronismo entre eles. O *coordenador* também é responsável por todas as execuções de cada camada, assim deve ter uma ligação direta com cada módulo que necessite de realizar a troca de parâmetros. No modelo proposto, esse módulo pode ser implementado por conjuntos de *scripts*, responsáveis por fazer chamadas de execução a cada camada da ferramenta.

Repositório

O módulo *Repositório* é o responsável por armazenar os arquivos de configuração já editados/configurados pelo usuário, de acordo com suas necessidades. Outra característica, é a possibilidade de restauração das configurações de uma máquina caso a mesma fique inoperante por qualquer motivo. Assim, será necessário apenas fazer uma instalação básica do sistema operacional e receber todos os arquivos de configuração referentes a antigos serviços, já configurados.

Conexão

O módulo de *Conexão* é um dos módulos mais importantes no sistema, ele é o responsável em estabelecer a conexão com os computadores remotos, que receberão os arquivos configurados e passarão a executá-los de acordo com a nova configuração estabelecida.

Esse módulo deve utilizar técnicas para o estabelecimento da conexão por meio de um canal seguro, essa conexão acontece entre a máquina com a ferramenta de administração e a(s) máquina(s) que receberá(ão) a configuração. Assim, depois da conexão estabelecida, o módulo recebe e envia os arquivos configurados ou armazenado no repositório para a máquina remota e executa um conjunto de comandos para manter o serviço desejado em funcionamento.

Maiores detalhes sobre as camadas/módulos da ferramenta, bem como características de funcionamento serão tratadas na Seção 4.5.1.

Da forma como proposto e apresentado, com a arquitetura em camadas se prevê uma flexibilidade da aplicação no seu desenvolvimento, possibilidades de heterogeneidade de tecnologias a serem incorporadas na ferramenta e um alto potencial para ampliação de seu escopo de atuação.

4.3 Funcionamento do Modelo

Depois da apresentação da arquitetura do modelo de ferramenta proposta, é apresentado a seguir o funcionamento de cada etapa do processo. Para iniciar essa explicação apresenta-se a Figura 4.2:

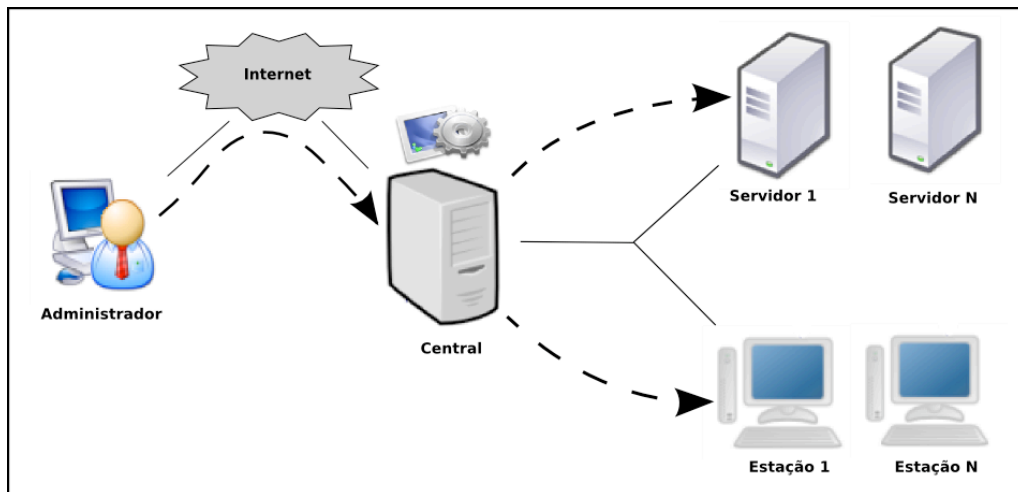


Figura 4.2: Funcionamento do Modelo Proposto

A ilustração procura exemplificar o funcionamento do modelo a partir das seguintes etapas:

1. O Administrador localizado em qualquer lugar remoto, utilizando apenas de um navegador de internet, acessa a ferramenta de administração por meio de uma página WEB;
2. Após se autenticar nessa página para ter acesso a seu conteúdo, o administrador seleciona um serviço/dispositivo a ser configurado;
3. O administrador poderá escolher em sua rede qual será a máquina servidora ou estação de trabalho, que receberá tal configuração e passará a trabalhar sob as novas perspectivas;

4. Depois de selecionar o serviço/dispositivo desejado e a máquina que receberá tal configuração, o administrador indica qual seu nível de conhecimento sobre o referido serviço, informação a qual será necessária para apresentar um formulário WEB contendo os parâmetros referentes a configuração do serviço desejado;
 - Se o administrador se considerar em nível iniciante, o sistema lhe apresentará um formulário com um conjunto bem limitado de parâmetros e completará o restante da configuração necessária com valores já pré estabelecidos, evitando que o mesmo faça uma má configuração, deixando brechas para falhas de segurança ou configurações inadequadas;
 - Se o administrador se considerar em nível intermediário, o sistema já lhe proporcionará um conjunto maior de parâmetro a serem configurados, completando apenas alguns parâmetros com valores padrão;
 - Se o administrador se considerar em nível avançado, ele terá total liberdade para o preenchimento de todos os parâmetros presente nos arquivos de configuração dos respectivos serviços/dispositivos a serem configurados.
5. Antes de propagar a nova configuração para a máquina desejada, o sistema faz uma copia do arquivo já configurado e o armazena no repositório de arquivos de configuração, para uma possível utilização futura, caso aconteça alguma coisa com a máquina e seja necessário refazer sua reconfiguração;
6. A configuração é propagada por meio de um canal seguro para a(s) máquina(s) remota(s), onde passará(ão) a executar a nova configuração.

Da forma como descrita nos passos anteriores é possível ter uma ferramenta de administração computacional capaz de ser acessada remotamente, por meio de uma página de internet e realizar configurações em outras máquina também remotas, por meio de um canal seguro e utilizando uma interface gráfica amigável.

4.4 Aspectos de Segurança

O objetivo desta seção é ilustrar aspectos de segurança envolvidos com o modelo de ferramenta proposta. Para tanto, são discutidos assuntos como as vulnerabilidades que podem ser causadas por um administração desatenta, além da utilização de ferramentas automatizadas de configurações de serviços de segurança, como uma tentativa de diminuição de falhas de operações causadas por uma configuração inadequada.

Vulnerabilidades

Segundo Geus [18], dados tempo, recursos e motivação, um intruso pode violar praticamente qualquer sistema. Mesmo que se conte com todos os recursos e procedimentos tecnológicos atualmente disponíveis para a segurança computacional, não se pode garantir 100% de proteção para quaisquer sistema.

Assim como já mencionado por Bishop [41], a principal estratégia dos profissionais de segurança é de utilizar os recursos disponíveis para tentar dificultar e minimizar as possibilidades dos incidentes acontecerem. Dentre tais recursos pode-se contar com técnicas, procedimentos e uso de ferramentas, enfim, praticamente qualquer tipo de ação que adicione bloqueios sucessivos à progressão de ataques, como por exemplo:

- Uso de roteadores (*gateways*) para ajudar a proteger as portas de comunicação com a internet;
- Uso de *Firewall* que ajudam a proteger os limites da rede;
- Uso de *VPN's* (Redes Privadas Virtuais) para transferir dados com segurança através de informações criptografadas;
- Uso de *IDS* (Sistemas de Detecção de Intruso) para alertar a tentativa de atividades maliciosas na rede;
- Uso de Servidores Virtuais na tentativa de ludibriar possíveis atacantes, etc.

Entretanto o sucesso dessas técnicas e procedimentos como de inúmeras outras dependem de outros fatores como:

- A qualificação e perícia dos funcionários responsáveis pela configuração, monitoração e manutenção destas tecnologias;
- Habilidades em consertar e atualizar serviços e dispositivos com eficiência e rapidez;
- Disponibilidade em manter vigília constantes sobre a rede e os sistemas computacionais.

Visto a evolução envolvida em sistemas e tecnologias da informação, proteger recursos computacionais nas organizações é uma tarefa bastante complexa. Devido a esta complexidade é difícil encontrar profissionais peritos para todos os seus sistemas e serviços. Assim, grande parte destes profissionais que atuam em segurança satisfazem algumas áreas e em outras muitas vezes encontram-se desatualizados ou despreparados. Isso ocorre porque cada área da segurança requer constante atenção e foco. Neste contexto, é possível se

ter em uma organização um profissional especializado em *Firewall* e incapaz de configurar e manter um *IDS*.

Em cenários como esses, existem também os problemas encontrados nos próprios sistemas e serviços, ao qual têm seu funcionamento comprometido por algumas vulnerabilidades (*bug's*) e exigem adequação à segurança, que deve ser realizada pelos administradores que em algumas situações estão despreparados ou desatentos a estes riscos.

Administração Desatenta

De acordo com Albuquerque [42], os administradores se vêem confrontados com o desafio de projetar, implementar, manter e monitorar um elevado número de serviços e mecanismos de segurança. Esses mecanismos em sua maioria possuem sintaxes de configuração heterogênea e complicada. Uma consequência dessa situação é que erros de configuração são causas frequentes de vulnerabilidades de segurança.

Como já descrito por Anderson e Albuquerque [43, 42] “A maioria das falhas de segurança se devem a erros de implementação e gerenciamento”. Resultado repetido em um estudo sobre três (3) principais serviços de grande porte na Internet [44]. Deste modo, conclui-se que os erros de configuração constituem a maior categoria dos erros de operação, sendo a causa mais frequente de falha em dois (2) de cada três (3) serviços analisados.

Administradores que não mantêm seus sistemas e serviços configurados apropriadamente proporcionam inúmeras ameaças a segurança de seus servidores e sua rede. De acordo com dados do *SANS (System Administration Network and Security Institute)*¹, a causa fundamental da vulnerabilidade na segurança computacional é “delegar pessoas não treinadas para manter a segurança e não prover nem treinamento nem tempo hábil para que o trabalho seja executado”. Isto se aplica tanto para os administradores iniciantes quanto para os mais experientes .

Alguns administradores falham por causa de sua pouca experiência e qualificação, outros por falta de atenção no momento de configuração de um serviço que exige muito rigor. A edição do arquivo de configuração de um determinado serviço requer muita atenção, qualquer falha no preenchimento de um parâmetro pode causar inúmeras vulnerabilidades. No entanto, esses são os exemplos clássicos de uma administração desatenta, os quais podem causar inúmeros danos às organizações, colocando-as em risco.

Ferramentas de Configuração de Segurança

Na tentativa de eliminar ou ao menos diminuir as vulnerabilidades causadas por uma administração desatenta ou despreparada, a utilização de ferramentas automatizadas de

¹www.sans.org

configuração se apresenta como possível solução à esses problemas. Principalmente reconhecendo a importância de um bom trabalho de administração de sistemas e lançando mão de programas importantes para o auxílio desta prática, as chamadas ferramentas de segurança [45].

Como demonstrado por Albuquerque [42], abordagens que ofereçam abstração, integração e ferramentas de suporte ao gerenciamento da configuração de mecanismos de segurança, são fundamentais para tornar o processo de configuração menos sujeito a erros e mais efetivo.

Algumas ferramentas de administração podem facilitar a configuração de serviços e dispositivos de forma segura, limitando os possíveis erros dos administradores, que são resultados de uma edição manual de arquivos de configuração. Entretanto, é importante ressaltar que não é descartada a possibilidade dessas ferramentas também causarem vulnerabilidades ao sistema. Neste sentido é de suma importância uma avaliação minuciosa das possíveis ferramentas a serem utilizadas.

4.4.1 O Modelo Proposto e os Aspectos de Segurança

O modelo de ferramenta proposto, também pode ser utilizado como mecanismo de configuração remota de serviços de segurança. Para tanto, faz uso dos mesmos princípios e procedimentos utilizados na configuração de outros serviços/dispositivos, como já demonstrado por Melo em outros trabalhos [45].

Com o modelo proposto é possível realizar a implementação e configuração de serviços de segurança, os quais tenham seu funcionamento baseado em arquivos no formato texto (*ASCII*) de forma remota, ou seja, sem a necessidade da presença física do administrador no ambiente em que se configure. Deste modo, utiliza-se de uma interface gráfica de acesso remoto via WEB e de fácil utilização.

Assim como são realizados tratamentos nos arquivos de configuração dos demais serviços e dispositivos configurados pela ferramenta, os mesmos procedimentos valem para os arquivos de configuração de serviços de segurança, os quais também passam por um processo de filtragem, conversão para o formato *XML* e posteriormente geração de formulários WEB que são incorporados pela interface gráfica. Deste modo, sob o quesito segurança o modelo de ferramenta proposto pode ser analisado sob dois aspectos: *i)* A segurança envolvida na ferramenta; *ii)* A segurança proporcionada pela ferramenta.

4.4.2 Segurança Envolvida no Modelo

Dentre as características de segurança envolvidas no modelo de ferramenta proposto, pode-se colocar em destaque duas etapas:

1. Segurança envolvida no acesso e conexão entre o administrador, utilizando uma máquina remota em qualquer lugar do mundo e a ferramenta de administração, situada dentro da rede da organização, como demonstra a Figura 4.3.

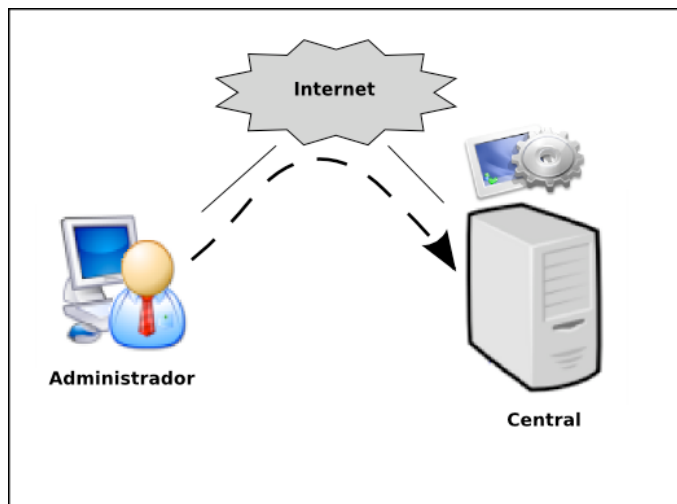


Figura 4.3: Segurança Envolvida entre o Administrador e a Ferramenta

Nesta etapa, entre os inúmeros mecanismos de segurança possíveis, se destacam:

- Um canal seguro de conexão, baseado em técnicas criptográficas como o *SSL (Secure Socket Layer)*². Esse protocolo ajuda a prevenir que intermediários entre as duas pontas da comunicação tenham acesso indevido ou falsifiquem os dados que estão sendo transmitidos;
 - Um mecanismo de controle de acesso, baseado no par usuário/senha, para permitir o estabelecimento de nível de privilégios entre os usuários e o acesso ao sistema.
 - Possibilidade de estabelecimento de filtros de pacotes, para permitir somente o estabelecimento de conexões a central de administração por pacotes que satisfaçam suas condições, como, por exemplo, portas específicas, protocolos e até endereços *IP*.
2. Segurança envolvida entre a ferramenta de administração e a máquina na rede que receberá a configuração desejada, como ilustra a Figura 4.4.

²Protocolo usado para criar páginas seguras, criptografando toda a transmissão entre um cliente e o servidor

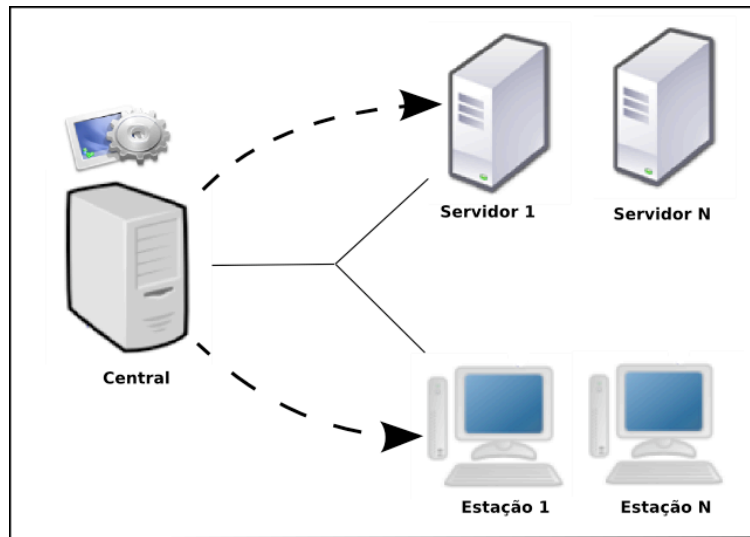


Figura 4.4: Segurança Envolvida entre a Ferramenta e a Máquina a ser Configurada

Nesta segunda etapa, as possibilidades de implementação de mecanismos de segurança também são muitas, entretanto podem se evidenciar :

- Um canal seguro de conexão entre a central de administração e as máquinas que serão configuradas. Para tanto, utiliza-se técnicas criptográficas estabelecidas pelo protocolo *SSH (Secure Shell)* que permite a conexão com outro computador da rede por um canal seguro, criptografado;
- Possibilidade de estabelecimento de filtros de pacotes para permitir a troca de mensagem entre a central e as máquinas da rede somente por endereços específicos como portas de comunicação, *IP*, etc.

É importante frisar que está sendo colocado em destaque algumas das inúmeras possibilidades de mecanismos de segurança que podem ser implementados no modelo proposto, sendo extremamente amplo o leque de possibilidades. No entanto, é assumido como premissa os aspectos de programação segura, algo que sempre deve ser a preocupação de um programador consciente de segurança e que deve ser amplamente considerado no momento de desenvolvimento, como neste momento o trabalho trata de uma proposta de modelo de ferramenta, não é dado ênfase a essas peculiaridades, mas considera-se características de fundamental importância para garantia mínima de segurança envolvida na ferramenta.

4.4.3 Segurança Proporcionada pelo Modelo

De acordo com Pinheiro [46], sistemas de segurança computacional podem ser implementados seguindo dois escopos:

1. Segurança de *host*; que são técnicas voltadas para a análise, correção e implementação de novos controles em sistemas computacionais.
2. Segurança de *rede*; que estão direcionadas para a verificação e o controle de tráfego de uma rede.

Com o modelo de ferramenta proposto, é possível configurar mecanismos de segurança que atenda os dois escopos.

Utilizando as técnicas envolvidas no modelo aqui definido, é possível a sua utilização para a configuração de serviços específicos de segurança de *host* como de *redes*, ou seja, é possível sua utilização para a configuração de outras ferramentas de segurança computacional, como sistemas de detecção de intruso, filtros de pacotes, entre outras que só são realizados até então por edição direta e manual de seus arquivos de configurações.

Com este cenário, pode-se definir o modelo de ferramenta proposto, como um *Framework* para a implementação de novos serviços de segurança, pois após a criação de um novo módulo para a configuração de um mecanismo, o mesmo poderá ser utilizado para ser implementado em qualquer outra máquina ou rede que se administre.

Outro quesito de segurança atendido pela ferramenta é a redução de falhas de operação, causada por uma administração desatenta. Com a utilização do modelo proposto a configuração dos mecanismos de segurança passam a ser realizados por formulários WEB, que possuem técnicas de validação de seus parâmetros na tentativa de redução de erros de configuração provocada pelos administradores.

A Figura 4.5, ilustra o funcionamento da ferramenta proposta, bem como a sua utilização para configurações de serviços de segurança.

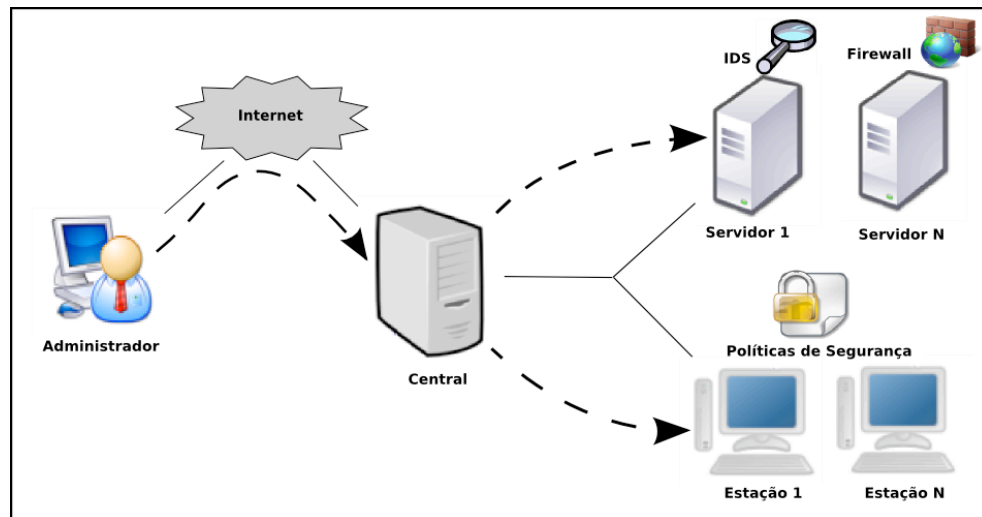


Figura 4.5: Utilização da Ferramenta para Configuração de Segurança

Com as características descritas no modelo proposto, o mesmo apresenta-se como uma alternativa aos administradores a ser utilizado para a configuração de serviços/dispositivos de forma segura, como também uma ferramenta para a configuração de outras ferramentas de segurança computacional, evidente que de forma automatizada, simplificada e remota, para os serviços que até então só eram configurados através da edição direta e manual de seus arquivos de configuração. No entanto, uma tentativa de utilizar um conjunto de ferramentas na ausência de pelo menos uma política de segurança implícita, não faz sentido e dificilmente será eficiente e eficaz.

4.5 O Protótipo

Na tentativa de validação do modelo de ferramenta proposto e em função das limitações de tempo de um programa de mestrado para implementação completa do sistema, foi desenvolvido um Protótipo.

O desenvolvimento do Protótipo foi essencial para o amadurecimento do projeto, contribuindo para a identificação das estruturas de dados necessárias, funcionalidades relevantes ao sistema, uma organização mais elaborada da proposta, além de uma prévia das dificuldades a serem enfrentadas durante a implementação da ferramenta, visto a sua complexidade devido a possível portabilidade para a administração/configuração de inúmeros serviço/dispositivos, a qual tenham suas configurações baseadas em arquivos no formato texto plano.

O protótipo foi desenvolvido considerando a modularidade prevista na arquitetura do modelo proposto. Assim, foram desenvolvidos os módulos da ferramenta de forma totalmente independente uns dos outros, sem a necessidade da homogeneidade de linguagem de programação e nem de outras tecnologias.

Cada módulo ou camada foi desenvolvido com a finalidade bem específica, apenas com o intuito de validação do modelo, sendo os esforços direcionados principalmente às funcionalidades e às estruturas, conforme o objetivo inicial do protótipo.

4.5.1 Detalhes de Implementação

Seguindo a estrutura apresentada na arquitetura do modelo proposto, os módulos foram sendo implementados e testados de acordo com suas necessidades. Os detalhes da implementação seguido de uma breve descrição das tecnologias utilizadas são apresentados na sequência:

Interface

O módulo de *Interface* foi construído utilizando as linguagens de programação *PHP*, *HTML*, *CSS* e *Java Script*. A utilização destas linguagens foi devido ao objetivo da ferramenta em ser utilizada através de um navegador de internet, possibilitando o acesso de qualquer computador remoto. Essas linguagens têm se mostrado mais eficientes e amplamente utilizadas quando se trata de desenvolvimento para WEB e este foi o principal quesito levado em consideração no momento de escolhê-las.

A complexidade envolvida no desenvolvimento deste módulo ficou por conta da definição de um *layout* gráfico que pudesse auxiliar o administrador na realização de suas tarefas, fazendo com que a forma como o administrador utilizasse o sistema não fosse mais um obstáculo, como algumas ferramentas anteriormente estudadas se mostraram. Isso remete a conclusão de que essa seria uma outra possibilidade de novos trabalhos, entretanto, relacionados a visualização de informação, tendo como objetivo desenvolver uma interface gráfica que melhor se adaptasse as necessidades dos administradores computacionais.

Outro ponto importante envolvendo a complexidade no desenvolvimento desse módulo, foi a validação dos campos dos formulários, referentes aos parâmetros dos arquivos de configuração. Esta tarefa se mostrou complexa, visto as inúmeras possibilidades de preenchimento de cada parâmetro, fazendo com que a validação pudesse ser realizada num caráter de sintaxe e não de semântica. No entanto, esse quesito serve para tentar diminuir as possíveis falhas causadas pelos administradores ao preencher o formulário com parâmetros escritos erradamente.

O módulo de interface incorpora os formulário WEB provenientes do módulo inferior e proporciona ao administrador um ambiente com a portabilidade que um navegador de internet permite e ainda pode evitar muitas falhas de operação de sistemas e serviços computacionais, devido à automatização do processo por meio de formulários que garantem sintaxe rígida de comandos. Além de proporcionar ao administrador um armazenamento do estado de seu ambiente como também uma interface agradável com descrições simplificadas de cada parâmetro à ser configurado relacionado aos respectivos serviços/dispositivos, cenário totalmente oposto dos tradicionais consoles de linhas de comandos.

Formulário

O Módulo *Formulário* é respectivamente a camada responsável por construir os formulários WEB à serem incorporados pela interface gráfica. Essa camada foi desenvolvida utilizando as linguagens de programação *PHP* e *Java Script*. A utilização do *PHP* foi devido a maior facilidade na integração deste módulo com o módulo de *Interface*, já o *Java Script* foi utilizado em trechos de códigos que são executados pelo navegador de

internet, ou seja, do lado cliente da aplicação, diminuindo o consumo de processamento da máquina servidora onde se concentra a ferramenta de administração, pois o número de administradores conectados pode influenciar o desempenho da mesma.

Este módulo recebe do módulo inferior (*conversor*) um arquivo no formato *XML*, contendo os parâmetros, delimitadores e valores, dos campos referenciados nos arquivos de configuração de cada serviço/dispositivo, que podem ser manipulados pelo administrador. No momento da geração do formulário, esse módulo consulta uma lista de parâmetros, também no formato *XML*, referente ao nível de conhecimento do administrador sobre cada serviço/dispositivo a ser configurado. Deste modo, se o administrador se considerar em um nível iniciante sobre aquele serviço, será construído e apresentado um formulário com um conjunto reduzido de parâmetros a serem editados, os quais estarão na lista de parâmetros referente ao nível iniciante, assim, o restante dos parâmetros necessários ao funcionamento do serviço/dispositivo será completado com valores pré estabelecidos, para evitar que ocorram erros por má configuração ou preenchimento de valores incorretos. Esse mesmo procedimento executado no modo iniciante, também vale para o modo intermediário, com a ressalva de que existe um conjunto maior de parâmetro a ser apresentado neste nível, no entanto, os mesmos estão presentes na lista, em *XML*. Caso exista a necessidade de alteração de parâmetro de um nível para o outro, basta a sua alteração ou inclusão na lista *XML*. Assim um parâmetro pode ser incorporado por um nível, mesmo que ele não seja indicado para aquele nível, o que proporciona maior flexibilidade à ferramenta.

Como é sabido e já descrito pela arquitetura do modelo proposto, a ferramenta possui três (3) níveis, o iniciante e intermediário já apresentados e o avançado, a qual se destina a administradores com maior experiência sobre o serviço/dispositivo a ser configurado. Para proporcionar a flexibilidade essencial e essa classe de administradores, o modelo de ferramenta possibilita a edição completa do arquivo de configuração do referido serviço/dispositivo a qual se deseja configurar, assim os administradores em nível avançado poderão fazer uma edição completa do arquivo de configuração de forma totalmente remota e com a mesma flexibilidade provida pelos editores convencionais utilizados em consoles textuais.

A partir de todo esse processamento/verificação, o módulo transforma o arquivo de configuração do referido serviço/dispositivo no formato *XML*, em um formulário WEB e repassa para o módulo de *Interface*.

Conversor

O Módulo *Conversor* é o responsável em converter os arquivos de configuração dos serviços a serem configurados, escritos no formato texto plano (*ASCII*) em arquivos semi-estruturados *XML*, que futuramente serão transformados em formulários e incorporados pela interface gráfica.

O Módulo foi escrito utilizando a linguagem de programação *Python* devido a flexibilidade que ela proporciona e a maior facilidade de integração com comandos nativos do sistema operacional.

Em seu funcionamento, esse módulo recebe os arquivos de configurações dos serviços já pré-tratados, ou seja, sem algumas características que são eliminadas nas etapas anteriores e constrói um outro arquivo no formato *XML*, definindo o que se refere à *parâmetro*, *delimitador* e *valor* de cada linha do arquivo de configuração. A definição destes elementos é considerada devido ser um padrão encontrado na maioria, senão todos, os arquivos de configuração, onde cada linha válida referente a uma configuração possui esses elementos, mesmo que os valores de parâmetros, delimitadores e valores sejam vazios [26].

No momento em que uma linha do arquivo de configuração está sendo convertida para o formato *XML*, um algoritmo identifica o delimitador da linha, e então separa o que se refere à parâmetro do que se refere à valor. Essa identificação acontece com base em uma consulta em um arquivo de delimitadores, desta forma, mesmo que o arquivo de configuração possua diversos formatos, basta inserir o tipo de delimitador que ele utiliza no arquivo de consulta, que o mesmo será tratado pelo módulo.

A maior complexidade envolvida no desenvolvimento desse módulo, está justamente em povoar o arquivo de delimitadores, pois cada arquivo de configuração de um determinado serviço/dispositivo possui um tipo de delimitador diferente, ao qual deve ser inserido no arquivo de consulta. Entretanto, caso exista a necessidade da ferramenta realizar a configuração de um serviço ou dispositivo novo, basta alimentar o arquivo de delimitadores com o respectivo formato encontrado no arquivo de configuração desse serviço/dispositivo. Para uma maior clareza sobre o formato dos arquivos de configuração aconselha-se um consulta no segundo arquivo do Apêndice A ou na Figura 4.9 a qual apresenta cada linha do arquivo contendo somente um parâmetro, delimitador e valor.

Coordenador

Como proposto na arquitetura do modelo, esse módulo é o núcleo da ferramenta, gerenciando todo o processo executado por ela, para tanto, o mesmo possui um conjunto de *scripts* responsável por realizar inúmeras atividades, entre elas pode-se dar destaque para:

1. Um *script* coordenador de execuções, que possui ligação direta com todos os outros módulos da ferramenta, responsável por iniciar as execuções dos respectivos módulos do sistema, bem como, fazer a troca de dados específicos para as camadas que necessitem;
2. Um conjunto de *scripts* responsável em manipular o módulo de *Repositório* onde estão os arquivos de configurações já configurados e prontos para serem enviados

para as máquinas adequadas. É graça a esse conjunto de *scripts* que o modelo possibilita a recuperação e propagação de configurações específicas para determinada máquina, ou conjunto delas;

3. Tratamento prévio dos arquivos de configuração recebidos do módulo de *Conexão* da ferramenta. Essa é uma das principais funções deste módulo, é por meio dessa etapa que acontece o primeiro tratamento nos arquivos de configuração. Nesse momento são removidos do arquivo elementos considerados irrelevantes ao funcionamento do serviço/dispositivo, que no entanto dificultam o processo de conversão para o formato *XML* e conseqüentemente a geração dos formulários. Entre os elementos que são removidos do arquivo de configuração original estão os comentários e linhas em branco.

Para ilustrar e formalizar o *script* que realiza esse processo de filtragem, foi construído um *Autômato Finito Determinístico* [47], o qual ilustra cada etapa do filtro.

O autômato finito é representado por uma **5-upla** $M = \{K, \Sigma, \delta, q_i, q_f\}$ onde:

- $K = \{q_i, q_n, q_d, q_e, q_f\}$, que é o conjunto dos estados possíveis do autômato, sendo q_i = estado inicial, q_n = estado neutro, q_d = estado de descarte, q_e = estado de escrita e q_f = estado final.
- Σ = Alfabeto
- δ = Função de transição (representada pela Tabela 4.2)
- q_i = estado inicial
- q_f = estado final

A linguagem executada pelo autômato é definida por:

$$L = \{ \Sigma \mid \Sigma \subset \text{Cmt, Alf, Smb, Cb, Ql, Eof} \}$$

Sendo uma cadeia de símbolos definidas sobre um alfabeto e os símbolos dessa cadeia são representados por:

- Cmd = conjunto de caracteres $\{ \text{" \#", " ;", " } \}$ seguido de Alf, Smb, Cb, Ql;
- Alf = conjunto de caracteres $\{ \{ \text{"A-Z"} \}, \{ \text{"a-z"} \}, \{ \text{"0-9"} \} \}$;
- Smb = conjunto de caracteres $\{ <, >, [,], (,), \{, \} \}$;

- Cb = conjunto de caracteres que representa o *Espaço em Branco* ou *Tab* $\{ \backslash 0, \backslash t \}$;
- Ql = caracter que representa a *Quebra de Linha* $\{ \backslash n \}$;
- Eof = caracter que representa *Fim de Arquivo*.

Deste modo, a linguagem a ser aceita por esse autômato, é sobre um alfabeto, tal que esse alfabeto contenha uma cadeia de símbolos definidos. No entanto, neste trabalho não será apresentado a composição da linguagem, sendo que somente a ilustração do autômato por sua máquina de estados e a tabela de transição já são suficientes para demonstrar os estágios de funcionamento de uma das etapas do módulo *coordenador*, a etapa de filtragem dos arquivos de configuração.

A *Tabela de Transição* entre os estados, assim como a *Máquina de Estado* que representa o funcionamento do autômato finito determinístico são apresentados pela Tabela 4.2 e Figura 4.6.

δ	$\delta(q, Ql)$	$\delta(q, Cmt)$	$\delta(q, Alf \mid Smb)$	$\delta(q, Cb)$	$\delta(q, Eof)$
qi	qi	qd	qe	qn	qf
qd	qi	qd	qd	qd	qf
qn	qi	qd	qe	qn	qf
qe	qi	qd	qe	qn	qf
qf	—	—	—	—	—

Tabela 4.2: Representação tTabular das Funções de Transição dos Estados do Autômato

A primeira linha da tabela representa a relação entre o estado atual da máquina e a entrada de um novo elemento. A primeira coluna representa os estados que podem receber a entrada de um novo elemento. Se a máquina estiver no estado *qi*(inicial) e entrar um elemento *Ql*, ela permanece no mesmo estado *qi*, mas, se entrar um elemento *Cmt* automaticamente ela vai para o estado *qd* (descarte). Ainda estando em *qi* e se entrar um elemento *Alf* ou *Smb*, vai para o estado *qe*(escrita), no entanto, se entrar um elemento *Cb* vai para o estado *qn*(neutro) e finalmente caso a entrada seja um elemento *Eof* vai para o estado *qf*(final) e finaliza o processamento da linguagem.

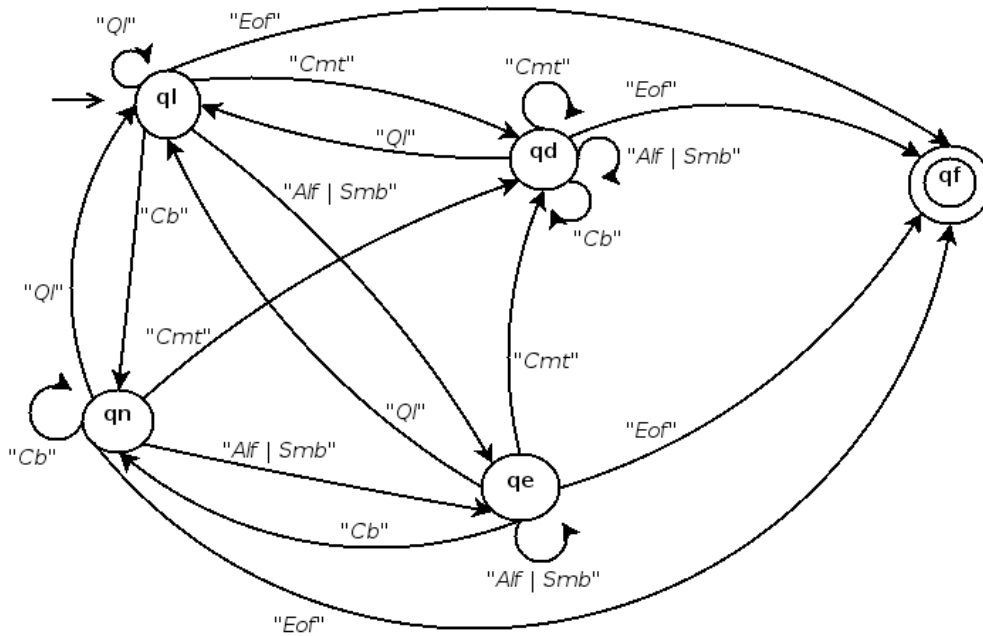


Figura 4.6: Autômato que Representa o Filtro dos Arquivos de Configuração

Note que esses mesmos procedimentos descritos valem para os outros estados da tabela de transição, que são executados pela máquina de estado apresentada.

Para a construção do *scripts* que executa a filtragem dos arquivos de configuração, formalizado pelo autômato finito determinístico apresentado, como também os outros *scripts* de fundamental importância desse módulo, foram utilizadas mais de uma linguagem de programação como *Python*, pelas suas características já comentadas e *Shell Script* pela simplicidade na construção de sequência de procedimentos a serem executadas no sistema operacional.

Desta forma, esse módulo é a peça fundamental nessa engrenagem. A complexidade envolvida em seu desenvolvimento é em manter o sincronismo com os outros módulos, repassando parâmetros e resultados de execução de um para o outro, além da construção do *script* de filtragem dos arquivos de configuração.

Repositório

O desenvolvimento deste módulo foi baseado em um *Sistema de Controle de Versão* que é integrado com a ferramenta para poder armazenar os arquivos já configurados, possibilitando a distribuição de um ou mais arquivos de configuração para uma ou mais máquinas que necessitem de mesma configuração ou para restauração de determinada configuração que devido a algum motivo deixou de funcionar a contento. Esse módulo se integra por meio de um conjunto de rotinas escritas em *Shell Scripts* que são executado pelo módulo

Coordenador e acionados por chamadas do usuário a partir da interface gráfica.

Um das dificuldades encontradas no desenvolvimento desse módulo foi a seleção do melhor sistema de controle de versão, visto que este seria um sistema separado integrado a ferramenta. Devido a existência de inúmeras possibilidades, tanto comerciais quanto livre, optou-se pela ferramenta livre *GIT*, pois a mesma possui uma característica peculiar em manter a integridade dos dados armazenados por meio de *hash* criptográfico, o que não ocorre com os outros sistemas de controle de versão. Entretanto, em função da complexidade envolvida nestes sistemas, coloca-se em destaque que esses podem ser objetos para novas pesquisas, devido à importância que essas ferramentas de controle de versão estão tomando no cenário de desenvolvimento de software, para maiores detalhes sobre essas ferramentas aconselha-se fortemente uma consulta mais aprofundada em [48] e [49].

Conexão

O Módulo de *Conexão* é um dos módulos mais importantes da ferramenta. Para conseguir atingir o objetivo de estabelecer conexões com máquinas remotas dentro de uma rede de computadores e por intermédio de um canal seguro, criptografado, esse módulo pode utilizar-se de duas alternativas:

1. A primeira faz utilização da conexão por meio de chaves criptográficas trocadas anteriormente entre a ferramenta de administração e as máquinas remotas;
2. A segunda faz utilização de uma aplicação separada chamada *SSHPASS*, que é capaz de passar por meio de parâmetro, o usuário e a senha de conexão pelo protocolo *SSH*, entre um cliente e servidor de forma também criptografada, além da possibilidade de indicar um arquivo de senhas (com permissões adequadas) como parâmetro para ser lido no momento do estabelecimento da conexão.

No entanto, as duas alternativas se mostram seguras, pois ambas utilizam um canal criptografado para a troca de informações. Em função da necessidade da construção de *scripts* para o estabelecimento da conexão entre a central e as máquinas remotas, utilizou-se a primeira opção, assim este módulo é constituído de *scripts* que são invocados pelo módulo *Coordenador*, o qual recebe chamadas da interface gráfica, selecionando a máquina remota a receber a conexão e conseqüentemente o arquivo configurado.

Em seu funcionamento, o módulo de *Conexão* executa um conjunto de *scripts* que estabelece a conexão com a máquina remota desejada e verifica se o arquivo de configuração referente ao serviço/dispositivo a ser configurado existe, se existir então faz uma cópia desse arquivo original e traz para máquina onde está a ferramenta de administração, assim o arquivo passará por todos os processos dos módulos já descritos. Caso o arquivo de configuração não seja encontrado a ferramenta alerta o administrador para que ele tome

as providencias cabíveis, como a instalação do pacote remotamente utilizando a própria ferramenta.

A dificuldade no desenvolvimento desse módulo, fica por conta da escolha da melhor forma de conexão por um canal seguro. No entanto, a maneira escolhida pode não ser a mais eficiente, mas foi a que se melhor encaixou no modo de funcionamento da ferramenta, antes porém, é preciso ressaltar que outras técnicas e procedimentos podem ser utilizadas, visto que o modelo proposto de ferramenta proporciona a flexibilidade necessária para a alteração de um módulo sem a necessidade de manutenção dos outros, situação essa que é válida para qualquer módulo.

4.5.2 Organização

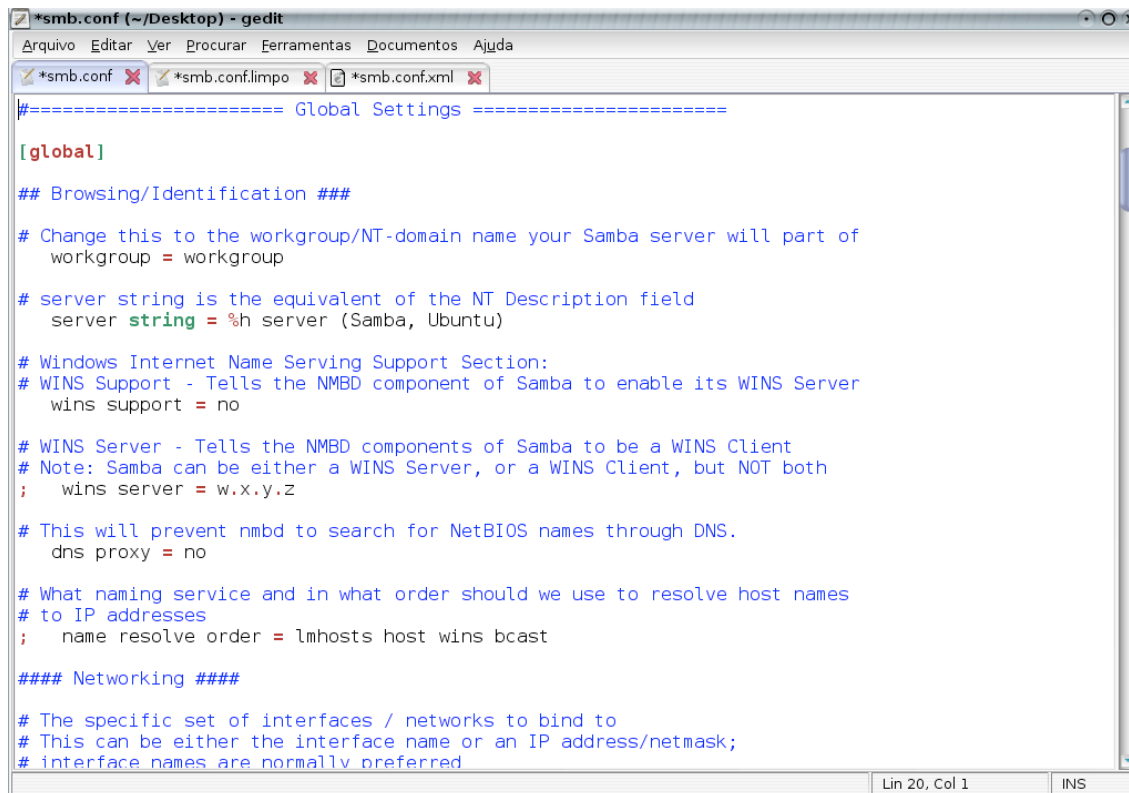
A ferramenta proposta, apesar de sua construção modular possui um sistema simples de organização, separando resultados de operações dos módulos em diretórios específicos e bem definidos, o que facilita o trabalho de manutenção da mesma.

Sua organização modular também auxilia no processo de adequação da ferramenta para a configuração de novos serviços, tornando-a uma espécie de *Framework* para configuração de serviços, aplicações ou dispositivos que tenham sua configuração baseadas em arquivos texto(*ASCII*). Desta forma o protótipo desenvolvido se apresenta semelhante ao modelo proposto, com sua estrutura organizacional e tecnologias envolvidas no seu desenvolvimento ilustrada pela Figura 4.7.

Módulo *Coordenador*:

Recebe um arquivo de configuração do módulo *Conexão* em seu formato original e aplica o primeiro tratamento. Na sequência é apresentado as Figuras 4.8 e 4.9, uma com um trecho de um arquivo de configuração original e outra com o resultado após o processamento do módulo.

Arquivo original: ***smb.conf***



```
##### Global Settings #####

[global]

## Browsing/Identification ##

# Change this to the workgroup/NT-domain name your Samba server will part of
workgroup = workgroup

# server string is the equivalent of the NT Description field
server string = %h server (Samba, Ubuntu)

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS Server
wins support = no

# WINS Server - Tells the NMBD components of Samba to be a WINS Client
# Note: Samba can be either a WINS Server, or a WINS Client, but NOT both
; wins server = w.x.y.z

# This will prevent nmbd to search for NetBIOS names through DNS.
dns proxy = no

# What naming service and in what order should we use to resolve host names
# to IP addresses
; name resolve order = lmhosts host wins bcst

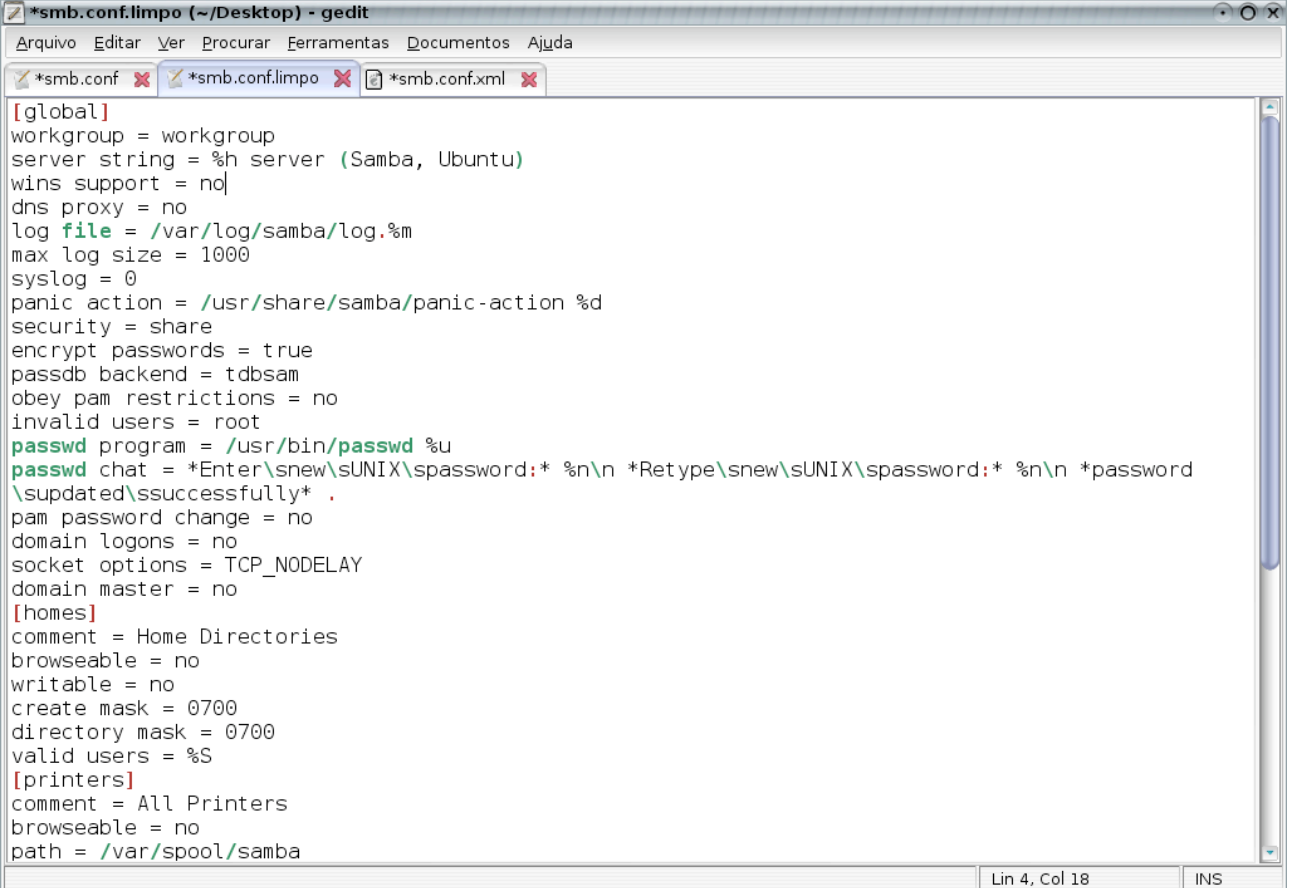
#### Networking ####

# The specific set of interfaces / networks to bind to
# This can be either the interface name or an IP address/netmask;
# interface names are normally preferred
```

Figura 4.8: Arquivo smb.conf Original

Nesta etapa o módulo *Coordenador* aplica o primeiro tratamento no arquivo de configuração do referente serviço, eliminando elementos desnecessários para o funcionamento do mesmo.

Arquivo processado: *smb.conf.limpo*



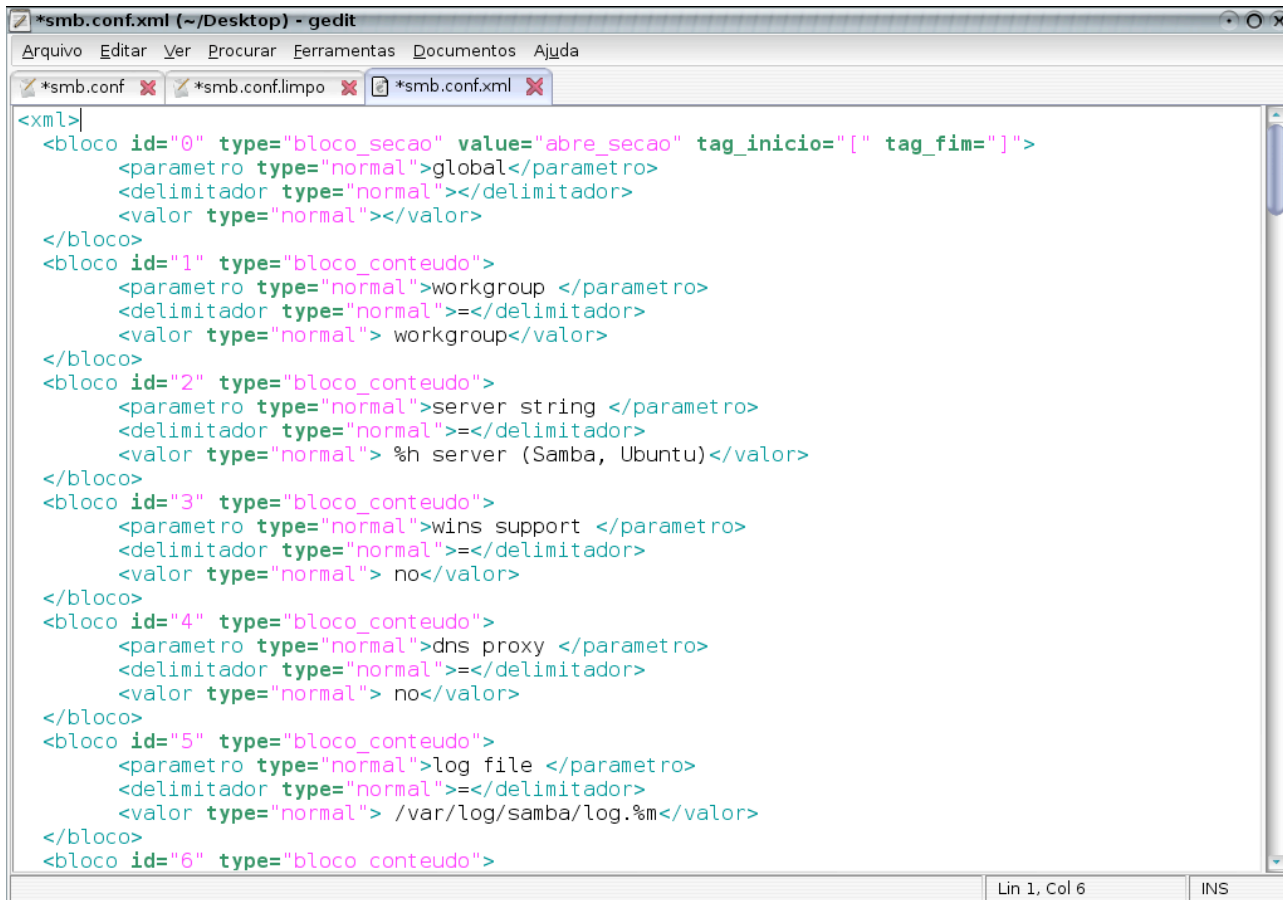
```
[global]
workgroup = workgroup
server string = %h server (Samba, Ubuntu)
wins support = no
dns proxy = no
log file = /var/log/samba/log.%m
max log size = 1000
syslog = 0
panic action = /usr/share/samba/panic-action %d
security = share
encrypt passwords = true
passdb backend = tdbsam
obey pam restrictions = no
invalid users = root
passwd program = /usr/bin/passwd %u
passwd chat = *Enter\snew\sUNIX\spassword:* %n\n *Retye\snew\sUNIX\spassword:* %n\n *password
\supdated\ssuccessfully* .
pam password change = no
domain logons = no
socket options = TCP_NODELAY
domain master = no
[homes]
comment = Home Directories
browseable = no
writable = no
create mask = 0700
directory mask = 0700
valid users = %S
[printers]
comment = All Printers
browseable = no
path = /var/spool/samba
```

Figura 4.9: Arquivo smb.conf Limpo

Módulo *Conversor*:

Recebe o arquivo de configuração processado pelo módulo *Coordenador*, e o converte para o formato *XML*:

Arquivo gerado: *smb.conf.xml*



```
<xml>
  <bloco id="0" type="bloco_secao" value="abre_secao" tag_inicio="[" tag_fim="]">
    <parametro type="normal">global</parametro>
    <delimitador type="normal"></delimitador>
    <valor type="normal"></valor>
  </bloco>
  <bloco id="1" type="bloco_conteudo">
    <parametro type="normal">workgroup </parametro>
    <delimitador type="normal">=</delimitador>
    <valor type="normal"> workgroup</valor>
  </bloco>
  <bloco id="2" type="bloco_conteudo">
    <parametro type="normal">server string </parametro>
    <delimitador type="normal">=</delimitador>
    <valor type="normal"> %h server (Samba, Ubuntu)</valor>
  </bloco>
  <bloco id="3" type="bloco_conteudo">
    <parametro type="normal">wins support </parametro>
    <delimitador type="normal">=</delimitador>
    <valor type="normal"> no</valor>
  </bloco>
  <bloco id="4" type="bloco_conteudo">
    <parametro type="normal">dns proxy </parametro>
    <delimitador type="normal">=</delimitador>
    <valor type="normal"> no</valor>
  </bloco>
  <bloco id="5" type="bloco_conteudo">
    <parametro type="normal">log file </parametro>
    <delimitador type="normal">=</delimitador>
    <valor type="normal"> /var/log/samba/log.%m</valor>
  </bloco>
  <bloco id="6" type="bloco_conteudo">
```

Figura 4.10: Arquivo smb.conf em XML

Módulo *Formulario:*

Recebe o arquivo no formato *XML* do módulo *Conversor* e cria formulários Web.

Arquivo gerado: *smb.con.php*

global	printers	print\$	Novo
workgroup	WORKGROUP		
server string	%h server (Samba, Ubuntu)		
dns proxy	no		
log file	/var/log/samba/log.%m		
max log size	1000		
syslog	0		
panic action	/usr/share/samba/panic-ac		
encrypt passwords	true		
passdb backend	tdbsam		
obey pam restrictions	yes		
invalid users	root		
unix password sync	yes		
passwd program	/usr/bin/passwd %u		
passwd chat	*Enter\snew\s*\spassword:		
pam password change	yes		
map to guest	bad user		
socket options	TCP_NODELAY		
usershare allow guests	yes		
gravar			

Figura 4.11: Parâmetros da Sessão *global*

global	printers	print\$	Novo
comment	All Printers		
browseable	no		
path	/var/spool/samba		
printable	yes		
guest ok	no		
read only	yes		
create mask	0700		
gravar			

Figura 4.12: Parâmetros da Sessão *printers\$*

global	printers	print\$	Novo
comment	Printer Drivers		
path	/var/lib/samba/printers		
browseable	yes		
read only	yes		
guest ok	no		
gravar			

Figura 4.13: Parâmetros da Sessão *print\$*

global	printers	print\$	Novo
Nome			
Comentário			
Caminho			
writable			
public			
create mode			
directory mode			
criar			

Figura 4.14: Parâmetros de uma *Nova* Sessão

Módulo *Interface*:

Recebe os arquivos do módulo *Formulário* e os incorpora na interface gráfica. Na sequência é apresentado exemplos de telas da interface criados pelo respectivo módulo:



Figura 4.15: Autenticação na Ferramenta



Figura 4.16: Alternativas de Configuração



Figura 4.17: Painel de Serviços Configuráveis

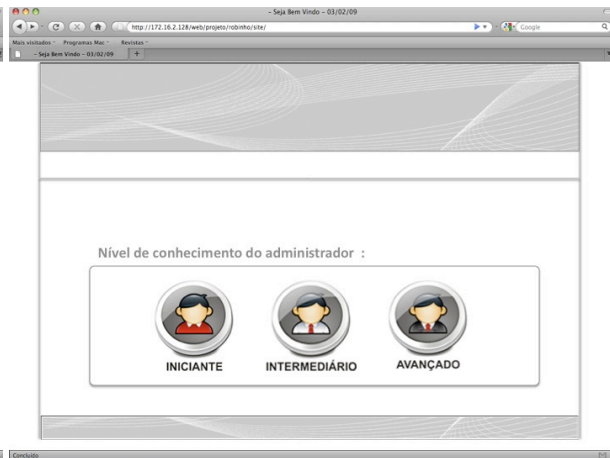


Figura 4.18: Nível Sobre o Serviço Desejado

Maiores detalhes como a visualização completa de todos arquivos gerados pelos respectivos módulos podem ser encontrados no Apêndice A.

4.6 Conclusão

Neste capítulo foi apresentado a proposta de um novo modelo de ferramenta para auxílio à administração computacional.

O modelo descrito tem como objetivo solucionar alguns dos maiores problemas que estão sendo enfrentados por administradores em cenários atuais, um deles é a necessidade da presença física do administrador no ambiente que se configura/administra. Para tanto, foi descrito características de funcionamento, arquitetura e segurança.

Foi apresentado a criação de um protótipo da ferramenta com o objetivo de validação do modelo, ilustrou-se alguns detalhes de implementação, assim como as tecnologias utilizadas para o seu desenvolvimento. Detalhes do funcionamento de cada módulo definido na ferramenta proposta também foram apresentados, assim como foi descrito a estrutura organizacional do protótipo desenvolvido, bem como os resultados de processamentos de cada módulo, alcançando o objetivo inicial de realizar a configuração/administração de máquinas e até redes inteiras de forma totalmente remota e utilizando uma interface gráfica para facilitar a administração, o que afasta dos administradores os ambientes hostis dos consoles e linhas de comandos, cenário esses muito propício a erros.

Capítulo 5

Testes e Resultados

Esse capítulo tem por objetivo apresentar o ambiente de testes realizados sobre o protótipo do modelo de ferramenta de administração computacional proposta, bem como, a configuração do serviço desejado, os vários cenários de testes montados e seus respectivos resultados, além da apresentação de uma Tabela com uma breve comparação entre o modelo proposto de ferramenta de administração computacional e os outros sistemas desta natureza analisados neste trabalho. Os testes realizados foram classificados em duas (2) categorias, Testes de Usabilidade, Testes de Funcionalidades. Também, foi realizado uma análise de aspectos básicos de segurança, cada um destes será discutido e terá seus resultados apresentados. É importante ressaltar que todos os teste foram realizados em ambientes reais, com computadores configurados especificamente para este propósito.

5.1 Ambiente de Testes

Para a realização dos testes efetuados com o intuito de validar o protótipo apresentado no *capítulo 4*, foi criado um ambiente controlado sobre um infra estrutura já existente e provida pelo Laboratório de Administração e Segurança de Sistemas (LAS), do Instituto de Computação (IC), da Universidade Estadual de Campinas (UNICAMP).

Nesse ambiente foram utilizadas os seguintes recursos:

- Um computador o qual será o servidor à ser configurado remotamente e proverá o serviço para as outras máquinas da rede. Sua configuração é:
 - Processador *Intel(R) Pentium(R) 4 CPU 3.00GHz*;
 - Memória *RAM de 2 Gb*;
 - Disco Rígido de *40 Gb*;
 - Sistema Operacional Instalado *Ubuntu kernel 2.6.24-16-generic*.

- Um computador com uma máquina virtual criado pelo pacote *VMware*¹, o qual será a central de configurações com o protótipo da ferramenta proposta instalada. Sua configuração é:
 - Processador *Intel(R) Core(TM) 2 Duo CPU 2.20GHz*;
 - Memória *RAM de 254 Mb*;
 - Disco Rígido de *8 Gb*;
 - Sistema Operacional Instalado *Debian kernel 2.6.18-5-686*.
- Um *MacBook* a qual será a máquina que o administrador utilizará para acessar a ferramenta de administração computacional e configurar o servidor do serviço desejado. Sua configuração é:
 - Processador *Intel(R) Core(TM) 2 Duo CPU 2.20GHz*;
 - Memória *RAM de 1 Gb*;
 - Disco Rígido de *160 Gb*;
 - Sistema Operacional Instalado *Mac Os X Leopard*.
- Um computador o qual será uma máquina da rede e cliente do serviço provido pelo servidor à ser configurado remotamente. Sua configuração é:
 - Processador *Intel(R) Pentium(4) 4 CPU 2.4GHz*;
 - Memória *RAM de 256 Gb*;
 - Disco Rígido de *40 Gb*;
 - Sistema Operacional Instalado *Windows XP*.

Todos esses computadores estão conectados em rede sob uma topologia de *Lan*, demonstrando que os testes realizados foram sobre um ambiente real e não simulado do ponto de vista de equipamentos.

5.2 Serviço Escolhido

Como o objetivo do desenvolvimento do protótipo era apenas a validação do modelo proposto, os testes realizados levaram em consideração apenas a configuração de um único serviço, pois o sucesso da configuração deste, já demonstraria a possibilidade de êxito em outros, visto que o foco da ferramenta são serviços ou dispositivos que tenham sua configuração baseada em arquivos texto. Assim foi escolhido o servidor para compartilhamento de arquivos e diretórios *Samba*.

¹<http://www.vmware.com>

5.2.1 O Samba

O *Samba* é um software livre desenvolvido para *Linux* (e outros sistemas baseados em *UNIX*) que serve como compartilhamento de recursos em uma rede heterogênea, formada por máquinas com sistemas operacionais diferentes. Com ele é possível utilizar um servidor para compartilhamento de arquivos, impressoras e outras funcionalidades, como se estivesse em uma rede totalmente homogênea entre os sistemas, proporcionando uma interoperabilidade entre as diferentes plataformas. Para maiores detalhes das possibilidades com o *Samba* recomenda-se uma consulta mais detalhada em seu *site* oficial ².

Devido as funcionalidades proporcionadas por este software, ele foi escolhido como sistema a ser configurado pelo protótipo, na tentativa de validação do modelo de ferramenta de administração computacional idealizada.

5.2.2 A Configuração Desejada

A interoperabilidade entre computadores com sistemas operacionais diferentes é algo cada vez mais desejado nas organizações atuais. A tarefa de manter a compatibilidade e principalmente a disponibilização de documentos entre essas plataformas é um dos problemas enfrentados por administradores computacionais desses ambientes, visto que algumas arquiteturas por princípio não oferecem essa possibilidade.

Assim, o cenário desejado é a configuração do *Samba* como um simples servidor de arquivo, que possa oferecer o compartilhamento de documentos entre plataformas de sistemas diferentes e totalmente transparente aos usuários de um ambiente semelhante a este.

5.3 Os Testes

Para a realização dos testes sobre o protótipo desenvolvido e no intuito de validar o modelo de ferramenta de administração computacional proposta, foi feita uma classificação dos testes a qual se subdividem em duas (2) categorias, Testes de Usabilidade, Testes de Funcionalidade. Também foi realizado uma análise de aspectos básicos de segurança.

5.3.1 Testes de Usabilidade

Os Testes de Usabilidade procuram demonstrar a simplicidade e facilidades em utilizar a interface gráfica da ferramenta proposta. Esse modelo de testes possibilita identificar um sequência simples, bem definida e contínua de passos necessários para a configuração de um serviço ou dispositivo utilizando o protótipo do modelo proposto.

²<http://www.samba.org>

Foi identificado que para configurar um serviço ou dispositivo o administrador deverá executar uma sequência de 10 (dez) passos, a qual serão ilustrados pelas Figuras a seguir.

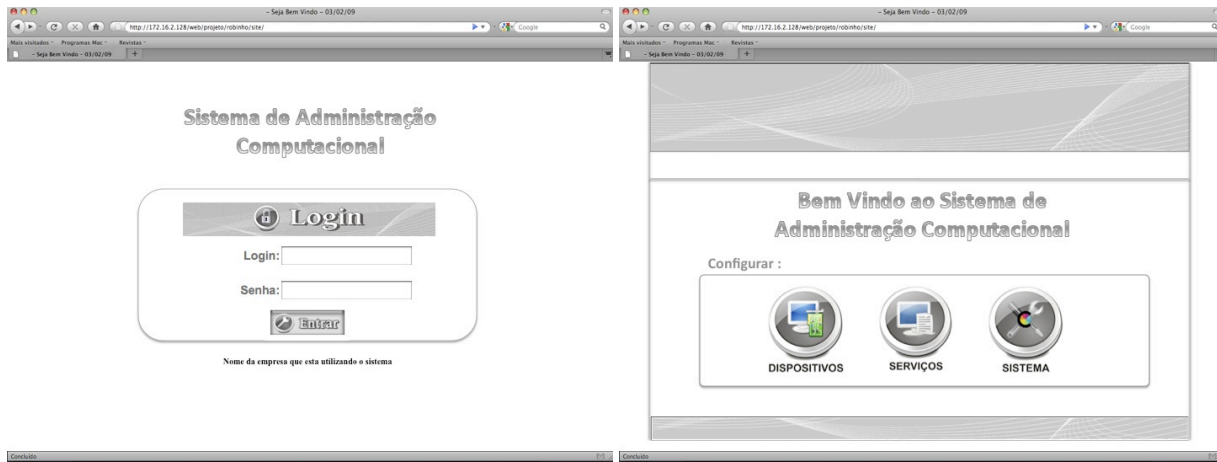


Figura 5.1: Configuração: Passo 1

Figura 5.2: Configuração: Passo 2

No *Passo 1* o administrador acessa a ferramenta pelo seu endereço WEB e realiza o processo de autenticação por intermédio de seu par, usuário e senha.

No *Passo 2* tem-se uma tela de abertura onde o administrador pode escolher o que deseja configurar: um dispositivo, um serviço ou a própria ferramenta, editando temas ou fazendo a distribuição de chaves assimétricas para o estabelecimento de conexão com máquinas a serem configuradas pela mesma.

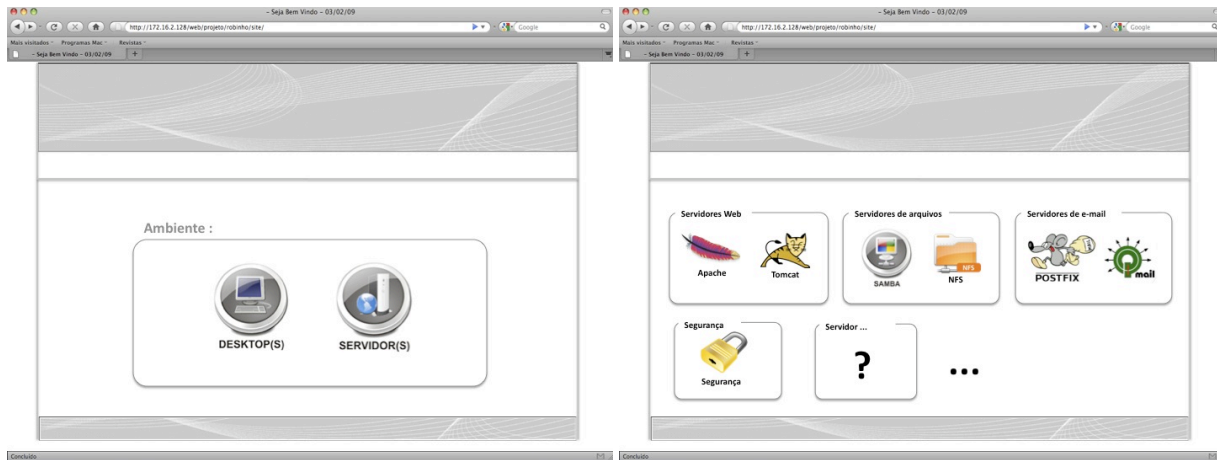


Figura 5.3: Configuração: Passo 3

Figura 5.4: Configuração: Passo 4

No *Passo 3* o administrador indica se fará uma configuração em um servidor ou uma estação de trabalho, o que pode definir um conjunto de serviços à serem configurados de acordo com sua especificidade.

No *Passo 4* é apresentado um painel com os possíveis serviços configuráveis pela ferramenta proposta, a partir de então o administrador escolhe o serviço desejado. vale ressaltar que apenas o samba esta implementado nesta fase do protótipo.

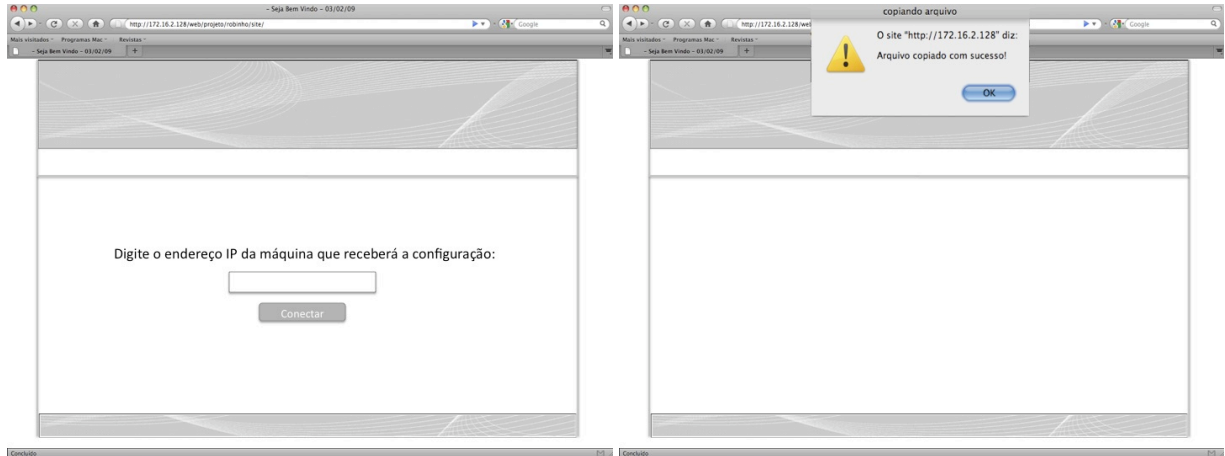


Figura 5.5: Configuração: Passo 5

Figura 5.6: Configuração: Passo 6

No *Passo 5* o administrador deverá indicar o endereço da máquina que receberá a configuração desejada e passará a funcionar de acordo com esta especificação.

No *Passo 6* é apresentado a confirmação positiva ou negativa sobre a obtenção do arquivo de configuração do respectivo serviço/dispositivo a ser configurado, o que indica se é necessário ou não realizar a instalação do pacote referente ao serviço a ser configurado na máquina remota.

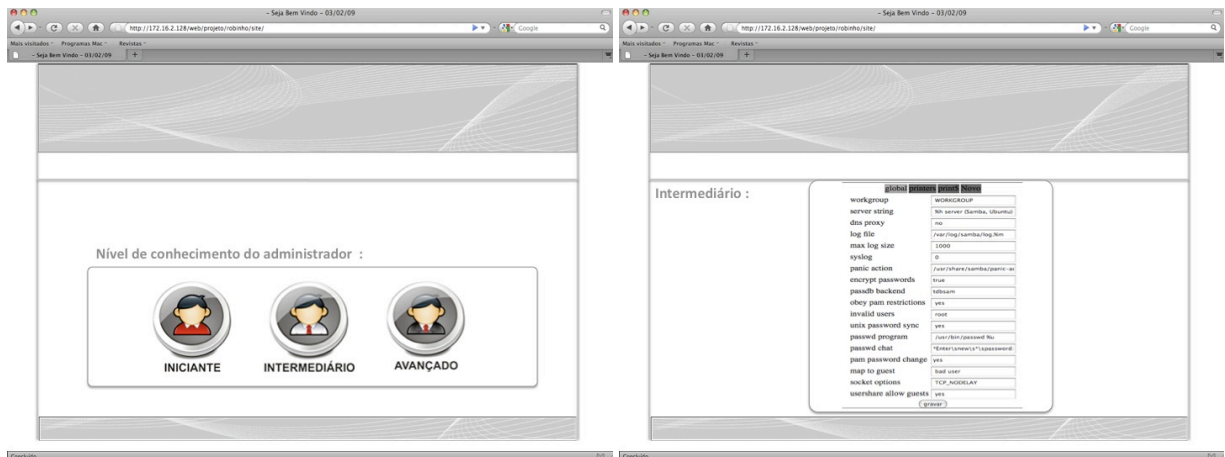


Figura 5.7: Configuração: Passo 7

Figura 5.8: Configuração: Passo 8

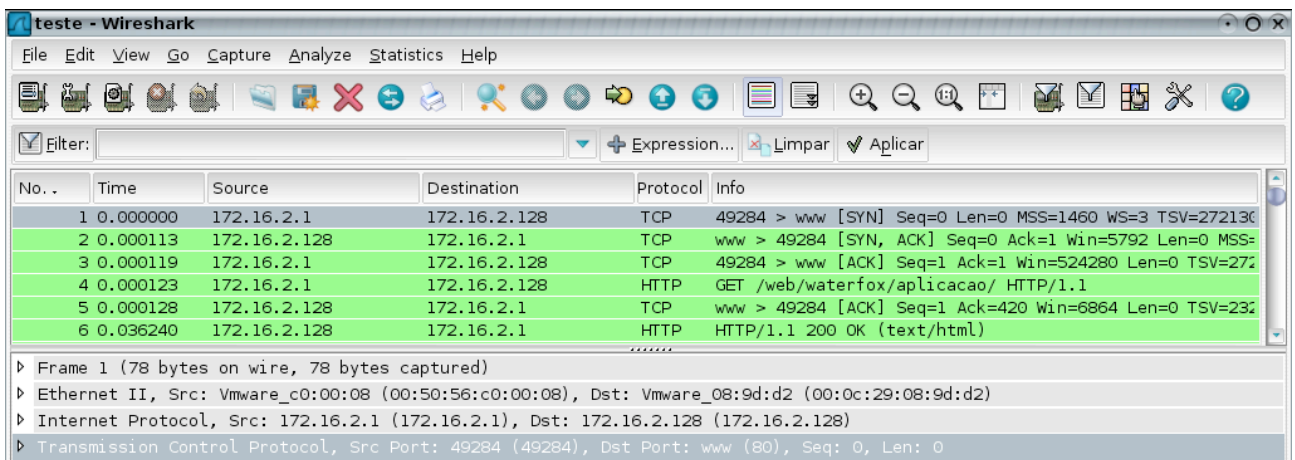
No *Passo 7* o administrador indica o nível de conhecimento em relação ao serviço

5.3.2 Testes de Funcionalidades

Os Testes de Funcionalidades visam demonstrar se as principais funções a qual o sistema se propões a realizar, realmente são executadas e de maneira eficiente.

Como já mencionado em capítulos anteriores foi desenvolvido um protótipo de ferramenta de administração computacional com o objetivo de validação do modelo proposto. Neste sentido, os testes de Funcionalidades executados recaem sobre os quesitos apresentados na sequência. Vale ressaltar que para a realização desses testes foi utilizado a ferramenta *sniffer Wireshark*³ de captura de tráfego de pacote na rede, o que ilustra a troca de mensagens entre as máquinas.

1. *Acesso Remoto do Administrador à Ferramenta pela WEB.* Esse quesito foi testado com a utilização de um computador, descrito no item 3 da seção Ambiente de Teste, a qual foi utilizado um navegador de internet para acessar o protótipo da ferramenta de administração computacional, instalado em outro computador (descrito no item 2 da mesma seção) por intermédio de uma URL, utilizando para isso o protocolo HTTP. A Figura 5.11 ilustra a captura dos pacotes trocados entre a máquina do administrador e a ferramenta de administração.



No. .	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.2.1	172.16.2.128	TCP	49284 > www [SYN] Seq=0 Len=0 MSS=1460 WS=3 TSV=272130
2	0.000113	172.16.2.128	172.16.2.1	TCP	www > 49284 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=
3	0.000119	172.16.2.1	172.16.2.128	TCP	49284 > www [ACK] Seq=1 Ack=1 Win=524280 Len=0 TSV=272
4	0.000123	172.16.2.1	172.16.2.128	HTTP	GET /web/waterfox/aplicacao/ HTTP/1.1
5	0.000128	172.16.2.128	172.16.2.1	TCP	www > 49284 [ACK] Seq=1 Ack=420 Win=6864 Len=0 TSV=232
6	0.036240	172.16.2.128	172.16.2.1	HTTP	HTTP/1.1 200 OK (text/html)

▶ Frame 1 (78 bytes on wire, 78 bytes captured)
 ▶ Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_08:9d:d2 (00:0c:29:08:9d:d2)
 ▶ Internet Protocol, Src: 172.16.2.1 (172.16.2.1), Dst: 172.16.2.128 (172.16.2.128)
 ▶ Transmission Control Protocol, Src Port: 49284 (49284), Dst Port: www (80), Seq: 0, Len: 0

Figura 5.11: Captura de Trafego dos Pacotes Entre Administrador e Central

Como ilustrado, observa-se que a conexão entre a máquina do administrador (endereço *ip 172.16.2.1*) e a central de administração (endereço *ip 172.16.2.128*) utilizando o protocolo HTTP, foi estabelecido, pois o software *wireshark* conseguiu capturar o estabelecimento desta conexão.

2. *Acesso entre a Ferramenta proposta e a máquina a ser configurada.* Para este teste foi realizado a configuração remota de um servidor de arquivos *Samba*. A Figura 5.12

³www.wireshark.org

ilustra a captura do estabelecimento de conexão entre a ferramenta de administração e a máquina remota que receberá a configuração e passará a atuar como o servidor.

The image shows a Wireshark packet capture window titled 'teste - Wireshark'. The main pane displays a list of captured packets. The filter is set to 'Filter:'. The packet list shows the following details:

No.	Time	Source	Destination	Protocol	Info
20	12.272069	172.16.2.128	10.1.1.222	TCP	33871 > ssh [SYN] Seq=0 Len=0 MSS=1460 TSV=2331143 TSE
21	12.276515	10.1.1.222	172.16.2.128	TCP	ssh > 33871 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS
22	12.276571	172.16.2.128	10.1.1.222	TCP	33871 > ssh [ACK] Seq=1 Ack=1 Win=5840 Len=0
23	12.280413	10.1.1.222	172.16.2.128	SSH	Server Protocol: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
24	12.284453	172.16.2.128	10.1.1.222	TCP	33871 > ssh [ACK] Seq=1 Ack=41 Win=5840 Len=0
25	12.284774	172.16.2.128	10.1.1.222	SSH	Client Protocol: SSH-2.0-OpenSSH_4.3p2 Debian-9
26	12.284981	10.1.1.222	172.16.2.128	TCP	ssh > 33871 [ACK] Seq=41 Ack=32 Win=64240 Len=0
27	12.288365	10.1.1.222	172.16.2.128	SSHv2	Server: Key Exchange Init
28	12.330852	172.16.2.128	10.1.1.222	TCP	33871 > ssh [ACK] Seq=32 Ack=825 Win=7056 Len=0
29	32.441349	172.16.2.128	10.1.1.222	SSHv2	Client: Key Exchange Init
30	32.442811	10.1.1.222	172.16.2.128	TCP	ssh > 33871 [ACK] Seq=825 Ack=744 Win=64240 Len=0
31	32.449638	172.16.2.128	10.1.1.222	SSHv2	Client: Diffie-Hellman GEX Request
32	32.449778	10.1.1.222	172.16.2.128	TCP	ssh > 33871 [ACK] Seq=825 Ack=768 Win=64240 Len=0

The packet details pane for packet 20 shows the following structure:

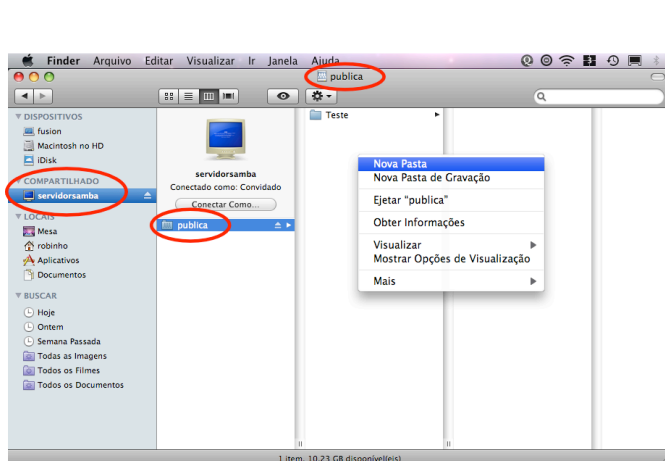
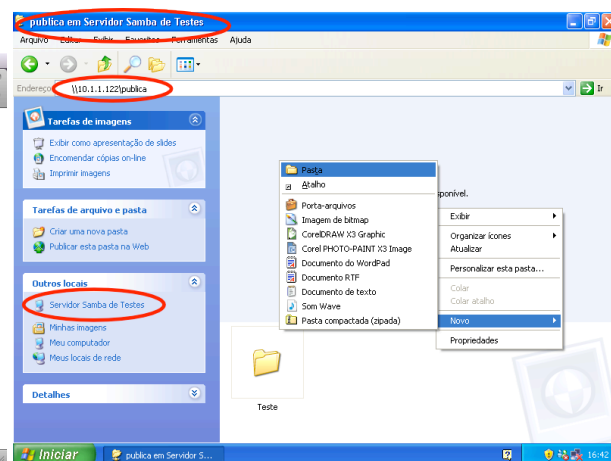
- Frame 20 (74 bytes on wire, 74 bytes captured)
- Ethernet II, Src: Vmware_08:9d:d2 (00:0c:29:08:9d:d2), Dst: Vmware_f1:11:b5 (00:50:56:f1:11:b5)
- Internet Protocol, Src: 172.16.2.128 (172.16.2.128), Dst: 10.1.1.222 (10.1.1.222)
- Transmission Control Protocol, Src Port: 33871 (33871), Dst Port: ssh (22), Seq: 0, Len: 0

Figura 5.12: Captura de Trafego dos Pacotes Entre a Central e a Máquina Servidora

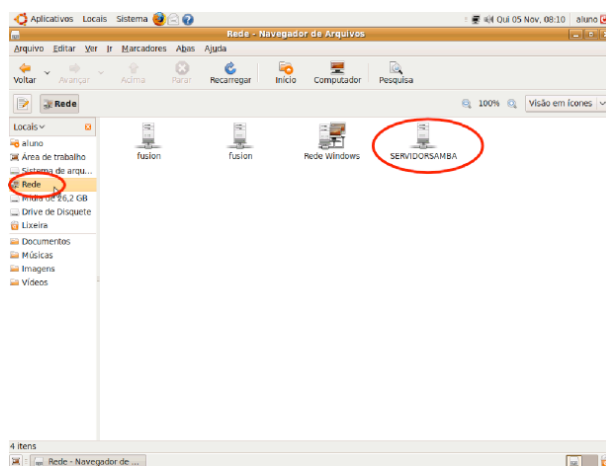
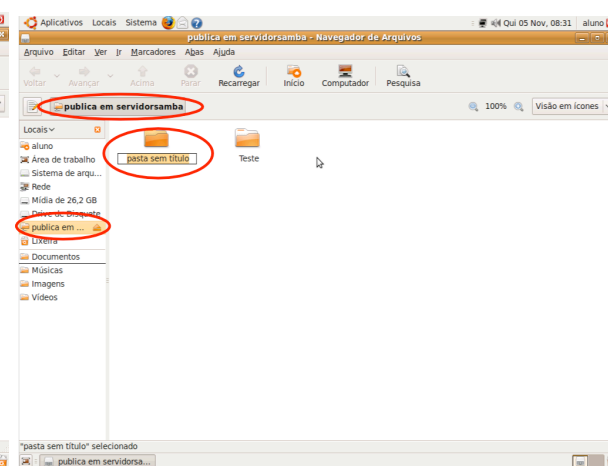
Assim como a conexão entre o administrador e a ferramenta, também foi satisfatório o teste que verifica a conexão entre a ferramenta (endereço *ip* 172.16.2.128) e o servidor que receberá a configuração do serviço desejado (endereço *ip* 10.1.1.222).

É necessário ressaltar que como o módulo da ferramenta responsável em estabelecer a conexão com as máquinas remotas a serem configuradas faz uso do protocolo *ssh*, o tráfego capturado indica a comunicação sobre o mesmo.

3. *Funcionamento do Serviço configurado.* Para o teste de verificação desta funcionalidade utilizou-se três computadores (descrito no item 1, 3 e 4 da seção Ambiente de Teste), cada um com um sistema operacional diferente que tentam compartilhar arquivos entre si na mesma rede. Os testes e resultados dessa funcionalidade podem ser vistos pelas figuras na sequência.

Figura 5.13: Acesso pelo *Mac Os X*Figura 5.14: Acesso pelo *Windows XP*

A Figura 5.13 ilustra um computador com o sistema operacional *Mac OS X* acessando um diretório público, compartilhado pelo servidor *Samba*, que foi configurado utilizando o protótipo da ferramenta proposta. O mesmo procedimento também foi possível de ser realizado com a utilização de um outro computador com o *Windows XP* ilustrado pela Figura 5.14.

Figura 5.15: Acesso pelo *Ubuntu 9.04*Figura 5.16: *Ubuntu* Criação de Diretório

Assim como computadores utilizando sistemas operacionais de famílias diferentes como *Mac OS X* e *Windows XP* puderam acessar o servidor *Samba*, máquinas com sistema operacional *Linux* como o *Ubuntu 9.04* também são capazes se conectar ao compartilhamento criado no Servidor, como ilustra as Figuras 5.15 e 5.16.

Vale ressaltar que além da permissão de acesso todos os modelos de sistemas operacionais ilustrados também possuem permissão de escrita, podendo armazenar arquivos e

diretórios no servidor.

Resultados dos Testes de Funcionalidade

Os resultado obtidos pelos testes de funcionalidade, demonstraram que a configuração remota de um servidor *Samba* para compartilhamento de arquivos entre sistemas operacionais de diferentes plataformas, como proposto e idealizado no cenário referente a seção *Configuração Desejada*, foi possível de ser realizada.

O teste que verificou a possibilidade do acesso remoto do administrador à ferramenta de administração, garantiu que a conexão por intermédio do protocolo *HTTP* aconteceu. Assim o teste que verificou a conexão entre a ferramenta de administração e máquina que será configurada também pôde ser realizado, garantindo e validando o modelo de administração remota entre o administrador e o ambiente que se configura.

Os testes que verificaram o quesito de funcionamento do serviço configurado pela ferramenta de administração proposta, foi satisfatório, pois foi possível realizar o compartilhamento de arquivos entre sistemas operacionais diferenciados, como previsto no cenário desejado. As figuras apresentadas ilustram os sistemas operacionais de diferentes arquiteturas compartilhando arquivos em uma rede de computadores heterogênea com simplicidade e facilidade, proporcionada pela configuração de um servidor de arquivos utilizando o modelo de ferramenta de administração computacional.

Deste modo, verifica-se que foi possível a utilização de uma ferramenta central de administração para a configuração de serviços em outras máquinas remotas, sem a necessidade da presença física do administrador no ambiente que se configura.

5.3.3 Análise de Aspectos Básicos de Segurança

O cenário de testes de segurança poderiam ser extremamente exaustivo visto o enfoque dado sobre o assunto, mas, como foi desenvolvido somente um protótipo com o intuito de validação do modelo proposto, não foi possível a realização de todos os testes desejados sobre os aspectos de segurança.

Sendo assim, limita-se a uma análise de aspectos básicos de segurança sob dois aspectos: 1) Controle de acesso baseado na autenticação do par *Login/Senha*, para permitir ou negar o acesso de usuários não autorizados a ferramenta. 2) A existência de um canal seguro para a troca de informações entre a central de administração e as máquinas na rede que serão configurada a partir dessa central. Este segundo item tem por objetivo evitar que terceiros possam interceptar a conexão, obter os dados e retransmití-los com alterações de configuração de cada serviço, causando possível brechas para futuro ataques. Para a realização dessa análise utilizou-se uma ferramenta de captura de tráfego na rede

Wireshark, para verificar a possibilidade de obtenção dos dados trocados entre a central de administração e a máquina a ser configurada.

Os resultados obtidos pela análise de aspectos básicos de segurança verificaram o funcionamento dos (2) dois aspectos considerados.

Sobre o primeiro quesito, os resultados mostraram que não é possível um acesso de um usuário não autorizado ao sistema visto que para isso o mesmo deverá ser previamente cadastrado, para receber seu respectivo par, *Login e Senha*, que ira identificá-lo no sistema e autorizará seu acesso com devidos privilégios. Esse procedimento no entanto é a primeira etapa para a utilização da ferramenta de administração. A ilustração que representa esse cenário pode ser vista pela Figura 5.1 na seção *Testes de Usabilidade*.

O resultado relacionado ao segundo quesito, pôde ser concluído graças a uma análise do tráfego de informações capturado entre a conexão das máquinas. Verificou-se que toda a segurança provida para a criação de um canal seguro entre as máquinas dessa conexão foi proporcionado pelo protocolo *SSH*. Como o módulo da ferramenta responsável em estabelecer a conexão com as máquinas remotas a serem configuradas faz uso desta tecnologia, toda segurança envolvida neste procedimento vêm desta aplicação, que devido a sua ampla utilização pelos administradores e as exaustivas tentativas fracassadas de descoberta de vulnerabilidades, garantem a criação de um canal criptografado e seguro. Mesmo que um intruso consiga capturar o tráfego de informações entre as máquinas, ele só obterá dados criptografados. A Figura 5.12 ilustra essa situação.

Deste modo, conclui-se que os quesitos de segurança observados pela simples análise puderam ser atendidos, mesmo que todas as possibilidades não foram exaustivamente testadas devido a completude da implementação da ferramenta, os resultados apresentados sobre o protótipo demonstram a viabilidade do modelo.

Os resultados apresentados utilizando o protótipo do modelo de ferramenta de administração computacional demonstraram a validação do modelo, bem como a possibilidade de utilização do mesmo para a configuração/administração segura de qualquer serviço ou dispositivo que tenha sua configuração baseada em um arquivo de formato texto (*ASCII*).

5.4 Comparação entre o Modelo Proposto e as Ferramentas de Administração Analisadas

Para o estabelecimento de uma comparação entre as funcionalidades e características encontradas nas principais ferramentas de administração computacional analisadas no capítulo 3. São apresentadas as Tabelas 5.1 e 5.2, que demonstram os aspectos existentes nas ferramentas que se destacam em sua categoria, sendo comparada com a ferramenta de administração computacional proposta neste trabalho.

Quesito	Cfengine	YaST	Webmin	Modelo Proposto
Configuração de um vasto conjunto de serviço/dispositivo	Sim/Limitado	Sim/Limitado	Sim/Limitado	Sim/Ilimitado
Construção modular	Não	Sim	Sim	Sim
Possibilita inserção de novos módulos	Não	Sim	Sim	Sim
Baseado em Interface Gráfica	Não	Sim	Sim	Sim
Modo de edição dos arquivos de configuração	Manual	Formulário	Formulário	Formulário
Método de geração dos formulários	Não existe	<i>Templates</i>	<i>Templates</i>	Automático
Possui algum tipo de validação de parâmetros na interface	Não	Não	Não	Sim
Possui controle de acesso ao sistema	Sim	Sim	Sim	Sim
Possibilita o acesso remoto via WEB	Não	Não	Sim	Sim
Possibilita configuração em máquinas remotas sem a necessidade de agentes	Não	Não	Não	Sim

Tabela 5.1: Comparação entre o Modelo Proposto e as Ferramentas Analisadas

Quesito	Cfengine	YaST	Webmin	Modelo Proposto
Possibilita instalação remota de software	Sim	Não	Não	Sim
Protocolos de comunicação	RPC (entre cliente e servidor)	Não	HTTP/HTTPS (entre navegador e ferramenta) RPC (no Cluste Webmin)	HTTP/HTTPS (entre navegador e ferramenta) SSH (entre ferramenta e máquinas remotas)
Possui repositório de arquivos de configuração editados pelo usuário	Não	Não	Não	Sim
Possibilita voltar a uma configuração anterior se necessário	Não	Não	Não	Sim
Possibilita propagar uma configuração para mais de uma máquinas	Sim	Não	Não	Sim
Mantém estado da sessão do administrador para continuação de configuração posteriormente	Não	Não	Não	Sim
Portabilidade de plataformas UNIX	Alta	Pouca	Alta	Alta
Estabelece níveis entre usuários para facilitar a configuração de um serviço/dispositivo	Não	Não	Não	Sim
Tratamento em arquivos de configuração para remoção de comentários e diminuição de tamanho	Não	Sim	Não	Sim

Tabela 5.2: Comparação entre o Modelo Proposto e as Ferramentas Analisadas (cont.)

Como apresentado nas tabelas anteriormente, existem vários aspectos que não são encontrados nas ferramentas de administração computacional existentes atualmente. Isso demonstra que o modelo proposto se apresenta como uma evolução às ferramentas desta natureza, proporcionando maior comodidade, agilidade e eficiência nas atividades de administração computacional.

O modelo proposto neste trabalho apresenta soluções inovadores para problemas antigos encontrados nas ferramentas de administração/configuração, como a configuração de máquinas remotas a uma central de configuração e sem a necessidade da presença física do administrador no ambiente que se configura, a existência de um repositório de arquivos de configuração já editados para a distribuição futura para uma ou mais máquinas da rede, entre outras características pertinentes aos cenários atuais não atendidas pelas ferramentas existentes.

5.5 Conclusão

Neste capítulo foram relatados os testes e resultados realizados sobre o protótipo do modelo de ferramenta de administração computacional proposto. Foi apresentado o ambiente de testes montado com a configuração das respectivas máquina utilizadas, o servidor de compartilhamento de arquivos configurado através da ferramenta proposta, o cenário idealizado para o funcionamento do serviço pretendido, assim como os resultados obtidos, além da apresentação sucinta de uma tabela comparativa dos principais aspectos encontrados no modelo de ferramenta computacional proposto e as outras ferramentas de administração analisadas. Para a realização dos testes e apresentação dos resultados deste capítulo fez-se um classificação dos mesmos em duas (2) categorias Testes de Usabilidade, Testes de Funcionalidades. Também, foi realizada uma análise de aspectos básicos de segurança, todos esses, levando em consideração o protótipo desenvolvido como um método de validação do modelo proposto.

Capítulo 6

Considerações Finais

A expansão das redes de computadores e com elas a evolução de sistemas computacionais, juntos trouxeram enormes contribuições para as organizações de todos os setores da sociedade. No entanto, manter essas redes e sistemas computacionais sempre operantes, em tempo integral, passou a ser uma tarefa complexa e com exigência de um alto grau de esforço e dedicação.

Administrar redes e sistemas de computadores sempre foi um grande desafio para os profissionais da computação. A necessidade de proporcionar um ambiente de produção estável e seguro, impulsiona os administradores a se manterem em constante aperfeiçoamento e capacitação.

Neste contexto, verificou-se que o administrador computacional deve reunir habilidades e conhecimentos sobre várias áreas da computação. Entretanto, o foco deste trabalho se resumiu a administração de redes de computadores como também conhecimentos sobre administração de sistemas computacionais. Essas habilidades e tarefas já são exigidas e executadas pelos mesmos profissionais, principalmente em organizações de pequeno e médio porte, onde as tarefas de administração de redes se fundem com as de administração de sistemas. Deste modo, é proposto a utilização do termo “Administração Computacional”, como uma nova alternativa de perfil profissional para os dias atuais, ao qual deve reunir inúmeras habilidades além do conhecimento de redes e sistemas operacionais.

Devido a essa enorme variedade de atribuições a esse novo perfil de profissional, as técnicas e ferramentas de administração computacional devem ser algo extremamente presente no cotidiano dos administradores. Com a mudança de paradigma é praticamente inconcebível a tarefa de administração sem a utilização de alguma ferramenta específica para este fim. Cenários onde somente a utilização de consoles e linhas de comandos, apesar da flexibilidade proporcionada, já não condizem com a exigência de uma configuração cada vez mais ágil e inerente a falhas causadas por uma configuração desatenta.

Deste modo, na tentativa de auxiliar os administradores, tanto os iniciantes quanto

os mais experientes, existem inúmeras ferramentas de administração/configuração de serviços ou dispositivos. Neste trabalho constatou-se que muito já foi feito nesse sentido, assim foi apresentado uma classificação e comparação entre as principais e mais populares ferramentas deste perfil.

Foi possível colocar em destaque, desde o potencial proporcionado pela construção de *scripts* como Ferramentas Caseiras de administração, para a realização de tarefas repetitivas, o que é uma prática muito comum entre os administradores, como também foi possível apontar a utilização de outras categorias de ferramentas de administração como as Ferramentas baseadas em Interfaces “Modo Texto”, sem interface gráficas e Ferramentas de administração com interface Gráfica, subdivididas em Ferramentas de Acesso Local e Ferramentas de Acesso Remoto.

O trabalho apresentou uma comparação entre as principais ferramentas de cada uma das categorias anteriormente apontadas. Tendo em vista a vantagem da utilização de Ferramentas de Administração de Interface Gráfica de Acesso Remoto, pela possibilidade de não necessitar da presença física do administrador, foi realizado um esforço maior sobre o a ferramenta *Webmin*, a qual se mostrou a mais madura entre as ferramentas com o mesmo perfil estudada.

A partir do estudo realizado entre as ferramentas de administração, apresentou-se uma proposta de ferramenta que contemple as principais características positivas das ferramentas estudadas, aliada a novos elementos não encontrados em nenhuma dessas ferramentas, mas que são julgados essenciais aos cenários vividos atualmente pelos administradores computacionais.

Dentre as principais características do modelo de ferramenta de administração proposto pode-se colocar em destaque:

1. A possibilidade de acesso remoto e configuração remota, eliminando a necessidade da presença física do administrador no ambiente que se configura. Esse quesito é parcialmente atendido por alguma ferramentas de administração como é o caso do *Webmin*, que possibilita o acesso remota a ferramenta, mas não a configuração em outra máquina da rede.
2. A não utilização de *Daemons* ou *Agentes* clientes nas máquinas remotas a serem configuradas, mas apenas pacotes comuns a uma instalação típica de um sistema *Linux*. A grande maioria das ferramentas de administração só possibilitam a configuração remota a través de programas agentes que ficam enviando informações pela redes a respeito das máquinas a serem configuradas.
3. A possibilidade de criação de um repositório permanente de arquivos de configuração, já pré-editados pelo usuário da ferramenta (*administrador*), para que caso haja a necessidade de restauração do serviço, por qualquer que seja o motivo, basta

apenas recuperar a configuração adequada dos arquivos de configuração armazenadas no repositório central, além de proporcionar a possibilidade de uma distribuição em massa de uma configuração para outras máquinas da rede.

4. A possibilidade de administração/configuração de qualquer serviço ou dispositivo, que tenha seu funcionamento baseado em um arquivo de configuração no formato texto (*ASCII*). Para isso é necessário a criação de módulos específicos para cada serviço/dispositivo e os incorporando a ferramenta.
5. Permanência de um estado de configuração do administrador, para que caso seja necessário o encerramento inesperado, ou não, de uma configuração ele possa voltar exatamente no estado em que parou.

Além destas características apontadas, existem várias outras julgadas de fundamental importância para os administradores nos cenários atuais, que foram apresentadas e descritas durante a dissertação.

No intuito de projetar esse novo modelo de ferramenta de administração computacional, foi apresentado uma proposta de ferramenta baseada na arquitetura, funcionamento e organização. No entanto, na tentativa de validação do modelo proposto e devido as limitações de tempo de um programa de mestrado foi desenvolvido um protótipo da ferramenta. O desenvolvimento do protótipo foi essencial para o amadurecimento do projeto, contribuindo pra a identificação das estruturas de dados necessárias, funcionalidades relevantes, uma organização mais elaborada da proposta e uma prévia das dificuldades a serem enfrentadas durante a implementação completa do projeto.

Por se tratar de uma aplicação experimental pouco foi feito com relação a sua completude, sendo os esforços focada principalmente na sua estrutura, conforme o objetivo inicial do desenvolvimento do protótipo.

O protótipo foi desenvolvido seguindo uma estrutura modularizada a qual cada módulo é totalmente independente do outro, além de possuir a característica de heterogeneidade de tecnologias utilizadas em sua construção, possibilitando o desenvolvimento com a utilização de várias linguagens de programação e outras recursos, como descrito durante o trabalho.

Após a apresentação do modelo proposto foi realizado uma análise de segurança sobre dois aspectos da ferramenta *i)* Segurança incorporado no modelo de ferramenta de administração, seriam as técnicas de segurança envolvidas na ferramenta como canais de comunicação seguros a base de criptografia, sistemas de autenticação de usuários, etc. *ii)* Segurança proporcionada pelo uso do modelo de ferramenta administração, neste sentido demonstrou-se que assim como ferramentas par a administração/configuração de dispositivos e serviços de rede, o modelo pode ser utilizado para a configuração de serviços de segurança como já demonstrado por Melo[45], em outros trabalhos, além de evitar muitas

outras falhas causadas por uma administração desatenta no momento de edição manual dos arquivos de configurações, cenário esse constante entre os administradores.

Com essas características apresentadas, a ferramenta proposta pode se tornar uma espécie de *Framework* para configuração de serviços de segurança, onde uma única configuração pode ser propagada para várias máquinas de uma rede, o que sem dúvida é um motivo de grande valia aos administradores, que podem realizar implementações de segurança sem a necessidade de sua presença física no local da máquina que se configura, como também, a possibilidade de restauração de uma configuração anterior caso aconteça algum tipo de inoperabilidade do sistema por qualquer motivo que seja, além outras inúmeras possibilidades.

Do ponto de vistas dos testes e resultado obtidos com a utilização do protótipo desenvolvido do modelo de ferramenta de administração computacional, e a partir de um cenário real de testes, levando em consideração a configuração remota de um servidor de compartilhamento de arquivos e diretório *Samba*, foi possível realizar uma classificação dos teste a qual se dividem em duas (2) categorias, Testes de Usabilidade, Testes de Funcionalidades e uma análise de aspectos básicos de segurança.

Os resultados obtidos e apresentados nos respectivos capítulos desta dissertação foram considerados satisfatórios, pois foi possível a realização da configuração do servidor da forma como esperado, totalmente remoto e utilizando o protótipo desenvolvido, o que garante a veracidade do funcionamento do modelo proposto.

Um modelo de ferramenta de administração computacional com essas características apresentadas, pode ser considerado um projeto audacioso e difícil de se concretizar, no entanto, como mostrado neste trabalho com uma arquitetura sólida, modularizada e com tecnologias adequadas, representa a possibilidade de revolução no modo como os administradores configuram e administram suas máquinas, redes e sistemas computacionais, principalmente pela possibilidade de não estarem presentes fisicamente no ambiente em que se administre.

6.1 Trabalhos Futuros

Considera-se que muito ainda há de ser feito para o desenvolvimento completo da ferramenta de administração computacional idealizada neste trabalho. Neste contexto, elencam-se como trabalhos futuros a implementação de novos módulos para a configuração de outros serviços específicos. Como foi implementado mecanismo para configuração do Servidor *Samba*, sendo essa uma possibilidade para novos trabalhos.

Como a ferramenta de administração proposta trata de serviços e dispositivos que tenham sua configuração implementado por arquivos no formato *ASCII (texto plano)*, verificou-se durante o trabalho uma enorme variação no formato e padrão dos arquivos de

configuração, deste modo, uma outra possibilidade de novos trabalhos seria a construção de um interpretador de arquivos de configuração, ao qual se passaria como entrada um arquivo de configuração em qualquer formato e ele seria capaz de retornar como saída um arquivo formatado em um padrão estruturado, o que facilitaria a construção de um módulo genérico para a criação de interfaces gráficas para esses serviços/dispositivos.

Um outro possível trabalho a ser desenvolvido trata exatamente do estudo e construção de um *Layout* gráfico adequado a administração computacional, com técnicas de visualização de informação que possa auxiliar o processo de configuração/administração de serviços/dispositivos computacionais.

Por ter a característica de armazenar arquivos de configuração, já editados pelos administradores, para a distribuição em massa ou de forma individualizada para as máquinas da rede, aponta-se como novas possibilidades de trabalhos um estudo mais detalhado sobre ferramentas de controle de versão e repositórios, ao qual podem contribuir de forma concisa para o aperfeiçoamento desta característica da ferramenta.

Além dessas possibilidades apontadas, inúmeras outras são possíveis e passíveis de novos trabalhos, pois como apresentado, o modelo proposto pode ter sua aplicação destinada a configuração/administração de qualquer serviço/dispositivo que tenha seu funcionamento definido por um arquivo de configuração, isso não limita a ferramenta à serviços de redes de computadores, sendo flexível suficiente às necessidades e criatividade dos profissionais da administração computacional.

Referências Bibliográficas

- [1] Rob Kolstad. 2001 sage salary survey. website, <http://www.sage.org/salsurv>, 2001.
- [2] Rob Kolstad SAGE/SANS/BigAdmin. Annual salary survey. webpage, <http://www.sage.org/salsurv/>, 2002.
- [3] SAGE. Annual sage salary survey for 2003. webpage, <http://www.sage.org/salsurv/>, agosto 2004.
- [4] SAGE. Sage annual salary survey 2004-2005. webpage, <http://www.sage.org/salsurv>, Outubro 2005.
- [5] SAGE-USENIX. Sage annual salary survey 2005-2006. webpage, <http://www.sage.org/salsurv/>, Outubro 2006.
- [6] Wikipedia. Comparison of remote desktop software – wikipedia, the free encyclopedia. website – <http://en.wikipedia.org>, Maio 2009.
- [7] James F. Kurose. *Redes de Computadores e a Internet: uma abordagem top-down*. Pearson Addison Wesley, 3 edition, 2005/2006.
- [8] Rede Nacional de Ensino e Pesquisa RNP. Introdução a gerência de redes tcp/ip. *Boletim bimestral sobre tecnologia de redes produzido e publicado pela RNP*, 1(3), agosto 1997.
- [9] Fredinand P. Drucker. *A Profissão de Administrador*. Pioneira Thompson Learning, 1998.
- [10] Ballesteros Alvarez. *Organização de Sistemas e Métodos*, volume 1. McGrawl Hill, 1990.
- [11] Avi Silberschartz. *Operating System Concepts*. Wiley, 7 edition, 2005.
- [12] William Stalling. *Operating systems: internals and design principles*. Pearson Prentice Hall, 5 edition, 2004.

- [13] Andrew Tanenbaum. *Sistemas Operacionais Modernos*. Rio de Janeiro. LTC, 1999.
- [14] Mark Burgess. Theoretical system administration. *LISA XIV*, Dezembro 2000.
- [15] Daniel Gouveia Costa. *Administração de Redes com Scripts: Bash Script, Python e VBScript*. Brasport, 1 edition, 2007.
- [16] Andrew Tanenbaum. *Computer Networks*, volume 4. Prentice Hall PTR, 2002.
- [17] Mário J.G. Antunes. Administração centralizada de grupos de sistemas unix. Master's thesis, Departamento de Ciência de Computadores; Faculdade de Ciência da Universidade do Porto; Portugal, Outubro 2001.
- [18] Emilio Tissato Nakamura and Paulo Lício de Geus. *Segurança de Redes em Ambientes Cooperativos*, volume 2. Novatec, 2007.
- [19] Mark Burgess. *Principles of Network and System Administration*. John Wiley e Sons, 1 edition, 2000.
- [20] Alexandre Oliva and Paulo Lício de Geus. Makeconfig: Configuração diferenciada de máquinas de redes unix heterogênes. *WAIS*, 1996.
- [21] Mário Antunes, Fernando Silva, and Manuel Eduardo Correia. Metawebmin: Administração centralizada de grupos de sistemas unix. *Actas da 4a. Conferência sobre Redes de Computadores, Tecnologias e Aplicações (CRC'2001)*, Covilhã, Novembro 2001.
- [22] Patricia Sofia Mourão Nunes da Silva. Análise de solução open source para administração de redes. Technical report, Instituto Politecnico de Leiria, Escola Superior de Tecnologia e Gestão, Departamento de Engenharia Informatica: Leiria, Portugal, Setembro 2006.
- [23] The Perl Foundation. Perl foundation. Website - <http://www.perl.org>, Abril 2008.
- [24] Mark Lutz. *Python Guia de Bolso*. Guia de Bolso. Alta Books, 3 edition, 2006.
- [25] Python Software Foundation. Python foundation. Website - <http://www.python.org>, Maio 2008.
- [26] Aurélio Marinho Jargas. *Shell Script Profissional*, volume 1. Novatec, 2008.
- [27] Michael A. Cooper. Overhauling rdist for the '90s. *USENIX System Administration Conference (LISA VI)*; California, USA, Outubro 1992.

- [28] Harlander Magnus. Central system administration in a heterogeneous unix environment: Genuadmin. *USENIX System Administration Conference (LISA)*, 1994.
- [29] Mark Burgess. Cfengine: host configuration manager. *USENIX Computing system*, 1995.
- [30] Mark Burgess. Cfengine: a site configuration engine. *USENIX Computing systems*, 8(3), 1998.
- [31] Charles Beadnall ; Andrew Mayhew. Cfadmin: A user interface for cfengine. *Proceedings of the 15th System Administration Conference (USENIX/LISA)*, 2001.
- [32] Redney Gedda. Google gets slack with software updates. website – Computerworld – <http://www.computerworld.com.au/article/206619/>, Outubro 2006.
- [33] Juliet Kemp. Marionete: Manipulando maquinas com o puppet. *Linux Magazine*, 42, Maio 2008.
- [34] Equipe Conectiva. *Entendendo o Conectiva Linux*. Conectiva S.A., 2003.
- [35] Jacques Gélinas. Linuxconf documentacion. website - <http://www.solucorp.qc.ca/linuxconf/>, Janeiro 2005.
- [36] Dee-Ann LeBlanc. Linux system administration tools: Introduces linuxconf, webmin, yast and coas. *Linux Jornal*, Maio 2002.
- [37] YaST Novell Inc. Team. *YaST Documentation*. Novell Inc., 2006.
- [38] Inc. Novell. Simple yast module. website, http://en.opensuse.org/YaST/Tutorials/Simple_YaST_Module, Fevereiro 2008.
- [39] Edison T. L. Melo, Elvis M. Vieira, and Solange T. Sari. Implantação de um sistema de gerenciamento distribuído utilizando tivoli. *II Workshop RNP2*, 2000.
- [40] Jamie Cameron. *Managing Linux systems with Webmin*. Pearson Education, 2004.
- [41] Matt Bishop. *Computer Security: Art and Science*. Addison Wesley Professional, 1 edition, Janeiro 2003.
- [42] João Porto de Albuquerque, Holger Isenberg, Heiko Krumm, and Paulo Lício de Geus. Gerenciamento baseado em modelos da configuração de sistemas de segurança em redes de larga escala. *V Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, 5, 2005.

- [43] Ross J. Anderson. Why cryptosystems fail. *Communications of the ACM*, 37(11):32–40, Novembro 1994.
- [44] David Oppenheimer, Archana Ganapathi, and David A. Patterson. Why do internet services fail, and what can be done about it? *Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems*, 4, 2003.
- [45] Robson Gomes de Melo and Paulo Lício de Geus. Proposta de um modelo de ferramenta de administração e segurança computacional. *Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, 2009.
- [46] Carlos Augusto Campana Pinheiro. Ferramentas de segurança. *Boletim bimestral sobre tecnologia de redes RNP - Rede Nacional de Ensino e Pesquisa*, 1(6), Novembro 1997.
- [47] Harry R. Lewis and Chistor H. Papadimitriou. *Elementos de Teoria da Computação*. Bookman, 2 edition, 2000.
- [48] GitWiki. Gitsvncomparsion. webpage – <http://git.or.cz/>, Outubro 2009.
- [49] David A. Wheeler. Comments on open source software / free software (oss/fs) software configuration management (scm) / revision-control systems. <http://www.dwheeler.com/essays/scm.html>, Maio 2005.

Apêndice A

Arquivos de Configuração Processados pela Ferramenta

Arquivo de configuração original, referente a configuração do servidor *Samba*.

Arquivo: *smb.conf*

```
# Sample configuration file for the Samba suite for Debian GNU/Linux.
#
#
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options most of which
# are not shown in this example
#
# Any line which starts with a ; (semi-colon) or a # (hash)
# is a comment and is ignored. In this example we will use a #
# for commentary and a ; for parts of the config file that you
# may wish to enable
#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.
#

#===== Global Settings =====
```

```
[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba server will part of
workgroup = WORKGROUP

# server string is the equivalent of the NT Description field
server string = %h servidor (Samba, Ubuntu)

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS Server
wins support = no

# WINS Server - Tells the NMBD components of Samba to be a WINS Client
# Note: Samba can be either a WINS Server, or a WINS Client, but NOT both
; wins server = w.x.y.z

# This will prevent nmbd to search for NetBIOS names through DNS.
dns proxy = no

# What naming service and in what order should we use to resolve host names
# to IP addresses
; name resolve order = lmhosts host wins bcast

#### Networking ####

# The specific set of interfaces / networks to bind to
# This can be either the interface name or an IP address/netmask;
# interface names are normally preferred
; interfaces = 127.0.0.0/8 eth0

# Only bind to the named interfaces and/or networks; you must use the
# 'interfaces' option above to use this.
# It is recommended that you enable this feature if your Samba machine is
# not protected by a firewall or is a firewall itself. However, this
# option cannot handle dynamic or non-broadcast interfaces correctly.
; bind interfaces only = true
```

Debugging/Accounting

```
# This tells Samba to use a separate log file for each machine
# that connects
    log file = /var/log/samba/log.%m

# Put a capping on the size of the log files (in Kb).
    max log size = 1000

# If you want Samba to only log through syslog then set the following
# parameter to 'yes'.
;    syslog only = no

# We want Samba to log a minimum amount of information to syslog. Everything
# should go to /var/log/samba/log.{smbd,nmbd} instead. If you want to log
# through syslog you should set the following parameter to something higher.
    syslog = 0

# Do something sensible when Samba crashes: mail the admin a backtrace
    panic action = /usr/share/samba/panic-action %d
```

Authentication

```
# "security = user" is always a good idea. This will require a Unix account
# in this server for every user accessing the server. See
# /usr/share/doc/samba-doc/htmldocs/Samba3-HOWTO/ServerType.html
# in the samba-doc package for details.
    security = share

# You may wish to use password encryption. See the section on
# 'encrypt passwords' in the smb.conf(5) manpage before enabling.
    encrypt passwords = true

# If you are using encrypted passwords, Samba will need to know what
# password database type you are using.
    passdb backend = tdbsam
```

```

obey pam restrictions = no

;   guest account = nobody
    invalid users = root

# This boolean parameter controls whether Samba attempts to sync the Unix
# password with the SMB password when the encrypted SMB password in the
# passwd is changed.
;   unix password sync = no

# For Unix password sync to work on a Debian GNU/Linux system, the following
# parameters must be set (thanks to Ian Kahan <kahan@informatik.tu-muenchen.de> for
# sending the correct chat script for the passwd program in Debian Sarge).
    passwd program = /usr/bin/passwd %u
    passwd chat = *Enter\snew\sUNIX\spassword:*
                %n\n *Retype\snew\sUNIX\spassword:*
                %n\n *password\supdated\ssuccessfully* .

# This boolean controls whether PAM will be used for password changes
# when requested by an SMB client instead of the program listed in
# 'passwd program'. The default is 'no'.
    pam password change = no

##### Domains #####

# Is this machine able to authenticate users. Both PDC and BDC
# must have this setting enabled. If you are the BDC you must
# change the 'domain master' setting to no
#
    domain logons = no
#
# The following setting only takes effect if 'domain logons' is set
# It specifies the location of the user's profile directory
# from the client point of view)
# The following required a [profiles] share to be setup on the
# samba server (see below)
;   logon path = \\%N\profiles\%U

```



```

# Another common choice is storing the profile in the user's home directory
;   logon path = \\%N%\%U\profile

# The following setting only takes effect if 'domain logons' is set
# It specifies the location of a user's home directory (from the client
# point of view)
;   logon drive = H:
;   logon home = \\%N%\%U

# The following setting only takes effect if 'domain logons' is set
# It specifies the script to run during logon. The script must be stored
# in the [netlogon] share
# NOTE: Must be store in 'DOS' file format convention
;   logon script = logon.cmd

# This allows Unix users to be created on the domain controller via the SAMR
# RPC pipe. The example command creates a user account with a disabled Unix
# password; please adapt to your needs
; add user script = /usr/sbin/adduser --quiet --disabled-password --gecos "" %u

##### Printing #####

# If you want to automatically load your printer list rather
# than setting them up individually then you'll need this
;   load printers = yes

# lpr(ng) printing. You may wish to override the location of the
# printcap file
;   printing = bsd
;   printcap name = /etc/printcap

# CUPS printing. See also the cupsaddsmb(8) manpage in the
# cupsys-client package.
;   printing = cups
;   printcap name = cups

# When using [print$], root is implicitly a 'printer admin', but you can
# also give this right to other users to add drivers and set printer

```

```
# properties
; printer admin = @ntadmin

##### Misc #####

# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name
# of the machine that is connecting
; include = /home/samba/etc/smb.conf.%m

# Most people will find that this option gives better performance.
# See smb.conf(5) and /usr/share/doc/samba-doc/htmldocs/Samba3-HOWTO/speed.html
# for details
# You may want to add the following on a Linux system:
#         SO_RCVBUF=8192 SO_SNDBUF=8192
#         socket options = TCP_NODELAY

# The following parameter is useful only if you have the linpopup package
# installed. The samba maintainer and the linpopup maintainer are
# working to ease installation and configuration of linpopup and samba.
; message command = /bin/sh -c '/usr/bin/linpopup "%f" "%m" %s; rm %s' &

# Domain Master specifies Samba to be the Domain Master Browser. If this
# machine will be configured as a BDC (a secondary logon server), you
# must set this to 'no'; otherwise, the default behavior is recommended.
#         domain master = no

# Some defaults for winbind (make sure you're not using the ranges
# for something else.)
; idmap uid = 10000-20000
; idmap gid = 10000-20000
; template shell = /bin/bash
;
; The following was the default behaviour in sarge
; but samba upstream reverted the default because it might induce
; performance issues in large organizations
; See #368251 for some of the consequences of *not* having
```

```

; this setting and smb.conf(5) for all details
;
;   winbind enum groups = yes
;   winbind enum users = yes

#===== Share Definitions =====

[homes]
    comment = Home Directories
    browseable = no

# By default, the home directories are exported read-only. Change next
# parameter to 'yes' if you want to be able to write to them.
    writable = no

# File creation mask is set to 0700 for security reasons. If you want to
# create files with group=rw permissions, set next parameter to 0775.
    create mask = 0700

# Directory creation mask is set to 0700 for security reasons. If you want to
# create dirs. with group=rw permissions, set next parameter to 0775.
    directory mask = 0700

# Restrict access to home directories
# to the one of the authenticated user
# This might need tweaking when using external authentication schemes
    valid users = %S

# Un-comment the following and create the netlogon directory for Domain Logons
# (you need to configure Samba to act as a domain controller too.)
;[netlogon]
;   comment = Network Logon Service
;   path = /home/samba/netlogon
;   guest ok = yes
;   writable = no
;   share modes = no

# Un-comment the following and create the profiles directory to store

```

```
# users profiles (see the "logon path" option above)
# (you need to configure Samba to act as a domain controller too.)
# The path below should be writable by all users so that their
# profile directory may be created the first time they log on
;[profiles]
;   comment = Users profiles
;   path = /home/samba/profiles
;   guest ok = no
;   browseable = no
;   create mask = 0600
;   directory mask = 0700

[printers]
    comment = All Printers
    browseable = no
    path = /var/spool/samba
    printable = yes
    public = no
    writable = no
    create mode = 0700

# Windows clients look for this share name as a source of downloadable
# printer drivers
[print$]
    comment = Printer Drivers
    path = /var/lib/samba/printers
    browseable = yes
    read only = yes
    guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# Replace 'ntadmin' with the name of the group your admin users are
# members of.
;   write list = root, @ntadmin

# A sample share for sharing your CD-ROM with others.
;[cdrom]
;   comment = Samba server's CD-ROM
;   writable = no
```

```
;   locking = no
;   path = /cdrom
;   public = yes

# The next two parameters show how to auto-mount a CD-ROM when the
# cdrom share is accessed. For this to work /etc/fstab must contain
# an entry like this:
#
#       /dev/scd0    /cdrom  iso9660 defaults,noauto,ro,user    0 0
#
# The CD-ROM gets unmounted automatically after the connection to the
#
# If you don't want to use auto-mounting/unmounting make sure the CD
# is mounted on /cdrom
#
;   preexec = /bin/mount /cdrom
;   postexec = /bin/umount /cdrom
```

Arquivo de configuração do servidor *Samba*, após o processamento do módulo *coordenador*.

Arquivo: *smb.conf.limpo*

```
[global]
workgroup = WORKGROUP
server string = %h server (Samba, Ubuntu)
wins support = no
dns proxy = no
log file = /var/log/samba/log.%m
max log size = 1000
syslog = 0
panic action = /usr/share/samba/panic-action %d
security = share
encrypt passwords = true
passdb backend = tdbsam
obey pam restrictions = no
invalid users = root
passwd program = /usr/bin/passwd %u
passwd chat = *Enter\snew\sUNIX\spassword:*
%n\n *Retype\snew\sUNIX\spassword:*
%n\n *password\supdated\ssuccessfully* .
pam password change = no
domain logons = no
socket options = TCP_NODELAY
domain master = no
[homes]
comment = Home Directories
browseable = no
writable = no
create mask = 0700
directory mask = 0700
valid users = %S
[printers]
comment = All Printers
browseable = no
path = /var/spool/samba
```

```
printable = yes
public = no
writable = no
create mode = 0700
[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no
```

Arquivo de configuração do servidor *Samba*, após o processamento do módulo *Conversor*.

Arquivo: *smb.conf.xml*

```
<xml>
  <bloco id="0" type="bloco_secao" value="abre_secao" tag_inicio="[" tag_fim=""]">
<parametro type="normal">global</parametro>
<delimitador type="normal"></delimitador>
<valor type="normal"></valor>
  </bloco>
  <bloco id="1" type="bloco_conteudo">
<parametro type="normal">workgroup </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> WORKGROUP</valor>
  </bloco>
  <bloco id="2" type="bloco_conteudo">
<parametro type="normal">server string </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> %h server (Samba, Ubuntu)</valor>
  </bloco>
  <bloco id="3" type="bloco_conteudo">
<parametro type="normal">wins support </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> no</valor>
  </bloco>
  <bloco id="4" type="bloco_conteudo">
<parametro type="normal">dns proxy </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> no</valor>
  </bloco>
  <bloco id="5" type="bloco_conteudo">
<parametro type="normal">log file </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> /var/log/samba/log.%m</valor>
  </bloco>
  <bloco id="6" type="bloco_conteudo">
<parametro type="normal">max log size </parametro>
```



```

<delimitador type="normal">=</delimitador>
<valor type="normal"> 1000</valor>
</bloco>
<bloco id="7" type="bloco_conteudo">
<parametro type="normal">syslog </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> 0</valor>
</bloco>
<bloco id="8" type="bloco_conteudo">
<parametro type="normal">panic action </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> /usr/share/samba/panic-action %d</valor>
</bloco>
<bloco id="9" type="bloco_conteudo">
<parametro type="normal">security </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> share</valor>
</bloco>
<bloco id="10" type="bloco_conteudo">
<parametro type="normal">encrypt passwords </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> true</valor>
</bloco>
<bloco id="11" type="bloco_conteudo">
<parametro type="normal">passdb backend </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> tdbsam</valor>
</bloco>
<bloco id="12" type="bloco_conteudo">
<parametro type="normal">obey pam restrictions </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> no</valor>
</bloco>
<bloco id="13" type="bloco_conteudo">
<parametro type="normal">invalid users </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> root</valor>
</bloco>

```

```

    <bloco id="14" type="bloco_conteudo">
<parametro type="normal">passwd program </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> /usr/bin/passwd %u</valor>
    </bloco>
    <bloco id="15" type="bloco_conteudo">
<parametro type="normal">passwd chat </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> *Enter&#92;snew&#92;sUNIX&#92;spassword:*
%n&#92;n *Retype&#92;snew&#92;sUNIX&#92;spassword:*
%n&#92;n *password&#92;supdated&#92;ssuccessfully* .</valor>
    </bloco>
    <bloco id="16" type="bloco_conteudo">
<parametro type="normal">pam password change </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> no</valor>
    </bloco>
    <bloco id="17" type="bloco_conteudo">
<parametro type="normal">domain logons </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> no</valor>
    </bloco>
    <bloco id="18" type="bloco_conteudo">
<parametro type="normal">socket options </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> TCP_NODELAY</valor>
    </bloco>
    <bloco id="19" type="bloco_conteudo">
<parametro type="normal">domain master </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> no</valor>
    </bloco>
    <bloco id="20" type="bloco_secao" value="abre_secao" tag_inicio="[" tag_fim="]">
<parametro type="normal">homes</parametro>
<delimitador type="normal"></delimitador>
<valor type="normal"></valor>
    </bloco>
    <bloco id="21" type="bloco_conteudo">

```

```

<parametro type="normal">comment </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> Home Directories</valor>
  </bloco>
  <bloco id="22" type="bloco_conteudo">
<parametro type="normal">browseable </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> no</valor>
  </bloco>
  <bloco id="23" type="bloco_conteudo">
<parametro type="normal">writable </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> no</valor>
  </bloco>
  <bloco id="24" type="bloco_conteudo">
<parametro type="normal">create mask </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> 0700</valor>
  </bloco>
  <bloco id="25" type="bloco_conteudo">
<parametro type="normal">directory mask </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> 0700</valor>
  </bloco>
  <bloco id="26" type="bloco_conteudo">
<parametro type="normal">valid users </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> %S</valor>
  </bloco>
  <bloco id="27" type="bloco_secao" value="abre_secao" tag_inicio="[" tag_fim="]">
<parametro type="normal">printers</parametro>
<delimitador type="normal"></delimitador>
<valor type="normal"></valor>
  </bloco>
  <bloco id="28" type="bloco_conteudo">
<parametro type="normal">comment </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> All Printers</valor>

```

```

    </bloco>
    <bloco id="29" type="bloco_conteudo">
<parametro type="normal">browseable </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> no</valor>
    </bloco>
    <bloco id="30" type="bloco_conteudo">
<parametro type="normal">path </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> /var/spool/samba</valor>
    </bloco>
    <bloco id="31" type="bloco_conteudo">
<parametro type="normal">printable </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> yes</valor>
    </bloco>
    <bloco id="32" type="bloco_conteudo">
<parametro type="normal">public </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> no</valor>
    </bloco>
    <bloco id="33" type="bloco_conteudo">
<parametro type="normal">writable </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> no</valor>
    </bloco>
    <bloco id="34" type="bloco_conteudo">
<parametro type="normal">create mode </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> 0700</valor>
    </bloco>
    <bloco id="35" type="bloco_secao" value="abre_secao" tag_inicio="[" tag_fim=""]">
<parametro type="normal">print$</parametro>
<delimitador type="normal"></delimitador>
<valor type="normal"></valor>
    </bloco>
    <bloco id="36" type="bloco_conteudo">
<parametro type="normal">comment </parametro>

```

```

<delimitador type="normal">=</delimitador>
<valor type="normal"> Printer Drivers</valor>
</bloco>
<bloco id="37" type="bloco_conteudo">
<parametro type="normal">path </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> /var/lib/samba/printers</valor>
</bloco>
<bloco id="38" type="bloco_conteudo">
<parametro type="normal">browseable </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> yes</valor>
</bloco>
<bloco id="39" type="bloco_conteudo">
<parametro type="normal">read only </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> yes</valor>
</bloco>
<bloco id="40" type="bloco_conteudo">
<parametro type="normal">guest ok </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> no</valor>
</bloco>
<bloco id="41" type="bloco_secao" value="abre_secao" tag_inicio="[" tag_fim="]">
<parametro type="normal">pub</parametro>
<delimitador type="normal"></delimitador>
<valor type="normal"></valor>
</bloco>
<bloco id="42" type="bloco_conteudo">
<parametro type="normal">comment </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> Diretorio publico</valor>
</bloco>
<bloco id="43" type="bloco_conteudo">
<parametro type="normal">available </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> yes</valor>
</bloco>

```

```
<bloco id="44" type="bloco_conteudo">
<parametro type="normal">browseable </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> yes</valor>
</bloco>
<bloco id="45" type="bloco_conteudo">
<parametro type="normal">path </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> /var/samba/pub</valor>
</bloco>
<bloco id="46" type="bloco_conteudo">
<parametro type="normal">public </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> yes</valor>
</bloco>
<bloco id="47" type="bloco_conteudo">
<parametro type="normal">guest only </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> no</valor>
</bloco>
<bloco id="48" type="bloco_conteudo">
<parametro type="normal">writable </parametro>
<delimitador type="normal">=</delimitador>
<valor type="normal"> yes</valor>
</bloco>
</xml>
```