

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

**Classificação e Comparação de algumas
Ferramentas de Administração Computacional**

R. Gomes de Melo P. Lício de Geus

Technical Report - IC-10-10 - Relatório Técnico

March - 2010 - Março

The contents of this report are the sole responsibility of the authors.
O conteúdo do presente relatório é de única responsabilidade dos autores.

Classificação e Comparação de algumas Ferramentas de Administração Computacional

Robson Gomes de Melo ^{*} Paulo Lício de Geus [†]

Abstract

Due to constant changes in computer systems for greater automation of tasks in organizations and speed the exchange of information, the expansion of computer networks is something uncontrollable. However the task of managing these networks and large systems is an ongoing challenge for computer professionals. In an attempt to help these professionals, there are many computational tools for administration. This technical report is part of a working thesis and presents a study of the major and most popular tools, classifying them and comparing them to serve as a reference material for choosing the ideal tool for the needs of each administrator.

Resumo

Devido a constante evolução dos sistemas computacionais, no sentido de maior automação de tarefas nas organizações e agilidade na troca de informações, a expansão das redes de computadores é algo incontrollável. No entanto a tarefa de administrar essas redes de grandes dimensões é um desafio permanente para os profissionais da computação. Na tentativa de auxiliar esses profissionais, existem inúmeras ferramentas de administração computacional. Este relatório técnico é parte de um trabalho de dissertação de mestrado e apresenta um estudo das principais e mais populares ferramentas, classificando-as e comparando-as, a fim de servir como um material de consulta para a escolha da ferramenta ideal as necessidades de cada administrador.

1 Introdução

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, a ponto de serem consideradas indispensáveis em tempo integral

^{*}Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP. robinho@las.ic.unicamp.br

[†]Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP. paulo@las.ic.unicamp.br

[1]. Isto significa que as tarefas de Administração, Gerência e Monitoração dessas redes e sistemas computacionais, também tem ganhado um papel significativo neste novo cenário.

Essas tarefas se tornam cada vez mais desgastante e exigem 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 as necessidades de uma organização. Entretanto, a evolução dos mecanismos e ferramentas utilizadas pelos administradores para satisfazer essa nova demanda não acompanham a evolução das novas tecnologias almejadas pelas organizações.

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

O objetivo deste trabalho é realizar um estudo das mais populares ferramentas de administração computacional, afim de classificá-las e posteriormente compará-las, com o intuito de servir de guia para auxiliar os administradores no momento de escolha e adoção de um destes softwares como ferramenta de administração a ser utilizado em sua organização.

Deste modo, esse trabalho se inicia com uma breve seção que apresenta uma discussão sobre ferramentas de administração computacional. Seguidamente, tem-se uma subseção que ilustra a classificação dos modelos de ferramentas a fim de facilitar o estudo das mesmas em diferentes categorias como: Os *Scripts* desenvolvidos pelos próprios administradores como ferramentas “Caseiras” de administração, as ferramentas de administração baseadas em interface a “Modo Texto” (sem interface gráfica) e as 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. Ao fim de cada categoria é apresentada uma comparação entre as ferramentas que se enquadram à classificação.

Uma terceira seção deste trabalho, apresenta um comparação geral entre as ferramentas de administração que se sobressaíram em relação as mesmas de sua categoria. Para finalizar é apresentado um seção com as considerações finais.

2 Ferramentas de Administração Computacional

De acordo com Oliva [2], 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 [3]. Esta tarefa por sua vez exige que o administrador tenha conhecimento sobre o sistema operacional e os serviços a qual deseja configurar para que sejam executado sob suas necessidades.

Melo[4], dispõe ser 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, principalmente para administradores iniciantes e aqueles necessitem de agilidade e rapidez.

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 computacional.

Este trabalho a fim de facilitar a escolha e uso das ferramentas de administração, apresentará um estudo comparativo e classificatório de algumas das mais populares ferramentas utilizadas pelos administradores em cenários atuais.

2.1 Classificação das Ferramentas de Administração

Para dar início ao trabalho exploratório sobre as ferramentas de administração computacional, se faz necessário uma classificação das mesmas no intuito de propor uma comparação avaliativa minuciosa sobre cada categoria, apresentando os aspectos positivos e negativos de cada ferramenta.

As ferramentas de administração computacional foram classificadas neste trabalho da seguinte maneira:

1. Os *Scripts* desenvolvidos pelos próprios administradores como ferramentas “Caseiras” de administração;
2. As ferramentas de administração baseadas em interface “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 categorias, *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 utilizadas pelos administradores computacionais, para a realização de suas atividades profissionais.

2.1.1 Ferramentas Caseiras de Administração Baseadas em *Script*

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

Aliado e incorporado a esses scripts caseiros estão outras inúmeras ferramentas de fundamental importância como:

- Expressões Regulares;

- AWK;
- SED;
- GREP.

Embora essas ferramentas apresentadas sejam poderosíssimas, não são as únicas, há um amplo conjunto de outras como o *GREP*, *CUT*, *TR* que associadas entre si auxiliam o administrador a manter o seus sistemas operantes.

A associação destas ferramentas geralmente é feita através de arquivos de *scripts*, que podem ser escritos em algumas linguagens específicas para cada interpretador. Dentre as principais Linguagem de *scripting* utilizadas pelos administradores para construção de suas ferramentas caseiras para automatização de tarefas, é possível apontar:

1. *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 [5, 6].

2. *Python*

Segundo Lutz [7] *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, além de ser de altíssimo nível.

3. *Shell Scripts*

Um *Shell Script* ou *Script em Shell*, segundo Jargas [8] é um arquivo texto que reuni várias linhas de comandos a serem executados em sequência por um programa interpretador. A utilização de *Shell Script*, como linguagem de *script* para a construção de ferramentas caseiras de administração computacional em sistemas baseados em *UNIX*, se apresenta como 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 torna 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 para falhas de segurança ao sistema, nem possa causar mau funcionamento a algum serviço a qual é de importância para os seus usuários e sua organização.

Deste modo, verifica-se que apesar de toda a flexibilidade proporcionada pelas ferramentas caseiras de administração, a sua utilização deve ser realizada por profissionais experientes e altamente preparados para este propósito.

2.1.2 Ferramentas de Administração Baseadas em Interface Modo Texto

Como apresentado por Melo [4], boa parte de administradores principalmente os mais experientes, costuma escrever seus próprios *scripts* como ferramentas caseiras para poder automatizar determinadas tarefas repetitivas em seu cotidiano.

No entanto essa solução não atende ao grupo de administradores que não são excelentes programadores ou aqueles que procuram utilizar de parcimônia para a construção e utilização de *scripts*.

Pensando nisso, foram desenvolvidas várias ferramentas de administração que utilizam um console em modo texto para serem executadas. Muitas dessas ferramentas são evoluções de ferramentas caseiras que sofreram baterias exaustivas de testes e aprimoramento de suas funcionalidades, desta forma, foram disponibilizadas para satisfazerem um nicho de mercado que ainda estava desamparado.

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 aos comandos de uma configuração desejada

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 [9].

Segundo Oliva [2], 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*.

¹<http://www.magnicomp.com/rdist/>

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 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 [2].

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 [2].

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 [10].

Segundo Correia [3] 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 [10, 11] 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 esta disponível Gratuitamente sob licença *GPL*.

4. MakeConfig

O *MakeConfig* é uma ferramenta de administração proposta por Alexandre Oliva [2]. 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 [2] 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 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 Olso, na Noruega em 1993 [12, 2]. 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 [13].

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

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*.

De acordo com Melo[14], 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 [15], 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 [16].

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

⁴<http://code.google.com/p/slack/>

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 [17].

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.

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:

- (a) 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.
- (b) 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 a ferramentas apresentadas, o inconveniente na sua utilização está na configuração prévia necessária nas máquinas clientes que rodam os *Daemos* e também no servidor onde se conecta esses *Daemos*.

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 *Daemos* clientes e o servidor.

Comparação entre Ferramentas de Administração Baseadas em Interface Modo Texto

Após a análise de algumas características, a Tabela 1 apresenta de forma sucinta uma comparação sobre o aspecto de funcionamento das ferramentas até aqui estudadas.

⁵<http://reductivelabs.com/>

Nome/Caract.	Repositório de admin.	<i>Daemon</i> cliente	Execução de <i>scripts</i>	Ling. 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 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 ilustrar alguns dos modelos de ferramentas que se adequam a essa classificação sugerida, finaliza-se a lista desta categoria de ferramentas ressaltando que a mesma não é exaustiva e as outras ferramentas não foram descritas em função de sua baixa popularidade entre os administradores e a comunidade acadêmica, devido a pouca literatura encontrada.

As ferramentas de administração baseadas em interface a modo texto estudadas, 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 de conexão e troca de informação entre os módulos clientes e servidor das ferramentas.

De acordo com Melo [14], 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.

2.1.3 Ferramentas de Administração Baseadas em Interface Gráfica

Essas ferramentas de administração se dispõem a facilitar a atividade do administrador com a utilização de janelas gráficas para auxiliar a edição direta de arquivos de configuração. Entretanto, ferramentas de administração baseadas em “Interface Gráficas” que cubram automatização 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. Encontrar tal modelo de ambiente gráfico, que disponibilize as facilidades de automatização e que proporcione ao administrador a flexibilidade necessária para a árdua tarefa de administração/configuração.

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 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 1 e 2.

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

⁶<http://www.redhat.com>

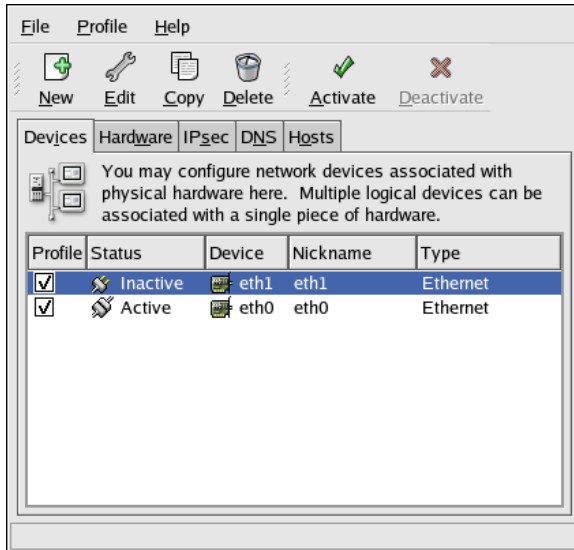


Figura 1: System-Config-Network

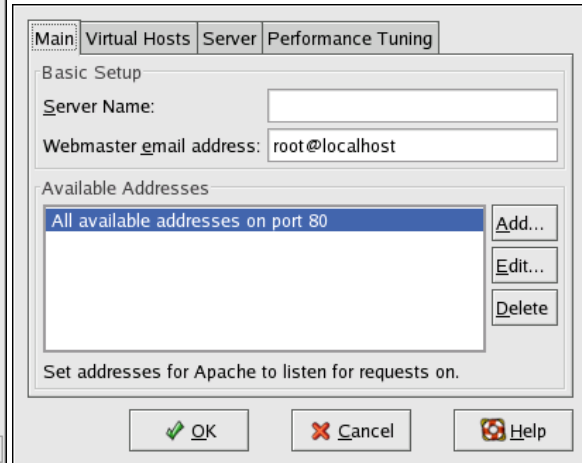


Figura 2: System-Config-Http

Dentre as ferramentas mais populares nesta linha evidenciam-se:

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 [18].

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* [19]. 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

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

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 [18]. Desta mesma maneira é possível que se habilite ou desabilite módulos específicos dentro da ferramenta.

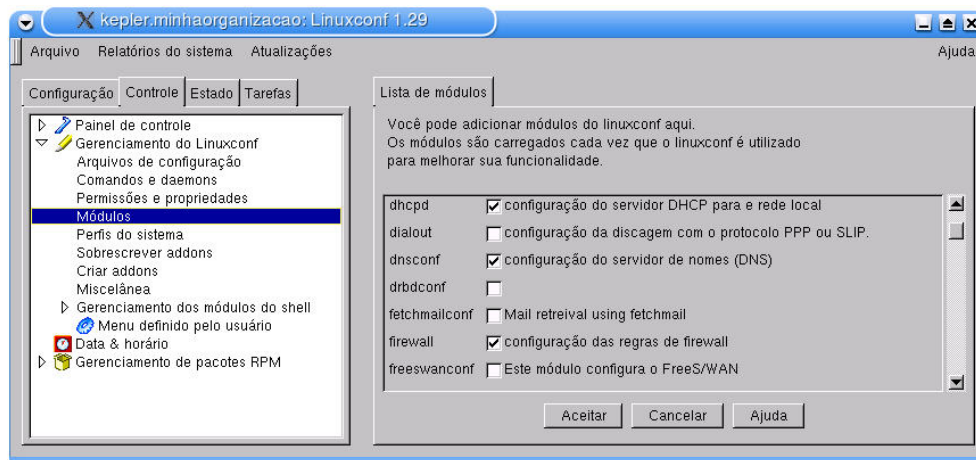


Figura 3: Módulos do Linuxconf

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 [19]. Entretanto, a sua interface mais consistente e madura, é a interface de acesso local, por isso essa ferramenta foi enquadrada nessa categoria.

O *Linuxconf* é uma ferramenta versátil e poderosa sendo utilizada por muitos administradores, 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

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

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

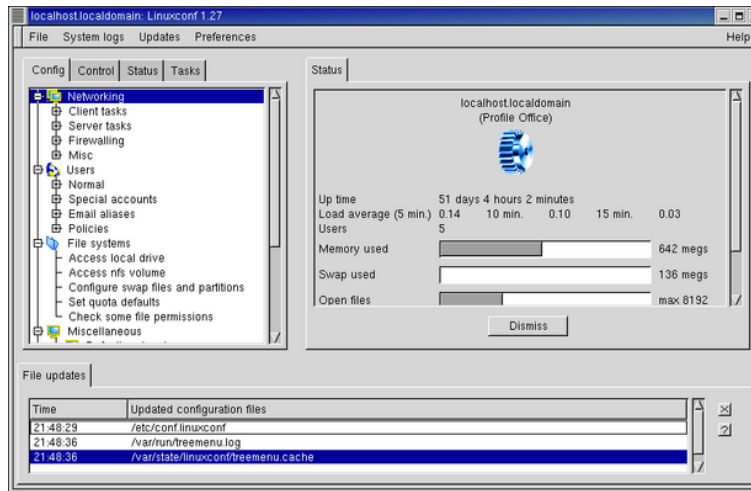


Figura 4: Interface do Linuxconf

descontinuado.

3. YaST

O *Yast* (*Yet another Setup Tool*) [20], 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 atrativa, 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 5 e 6.

a) Arquitetura do YaST.

Segundo Novell Inc. Team [21], 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*

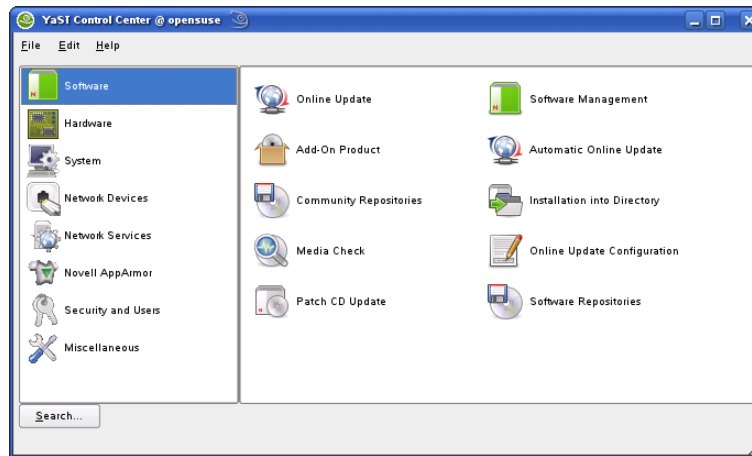


Figura 5: YaST Control Center

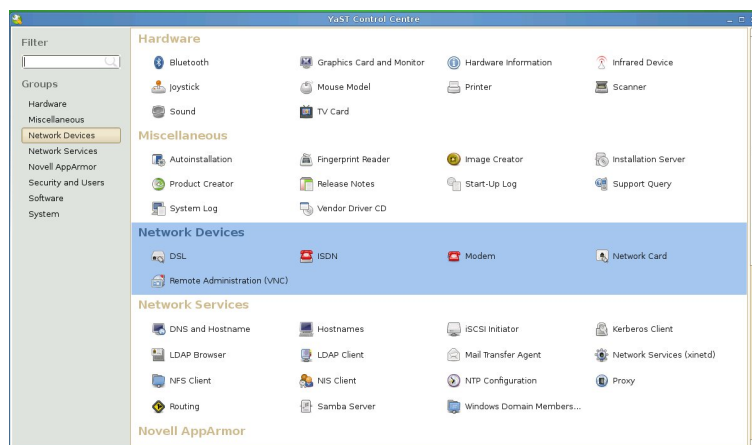


Figura 6: Seções do YaST

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 7, encontrada no Tutorial de Módulos do *YaST* [22], 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.

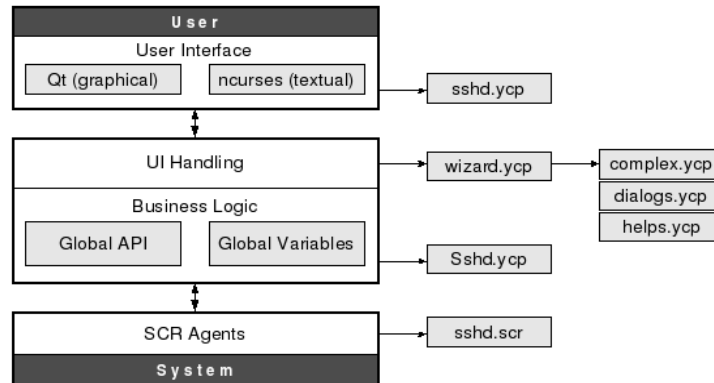


Figura 7: Arquitetura do YaST

Como ilustra a Figura 7 o *YaST* tem sua arquitetura subdivida em quatro (4) camadas:

- 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.
- 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.
- 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 contém 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.
- 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*.

Comparação entre as Ferramentas Gráficas de Acesso Local

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 2 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 2: 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. Neste sentido as ferramentas desenvolvidas utilizando tecnologias voltadas para a Internet tem se destacado. Entre as ferramentas de acesso remoto de apoio à administração, com essa característica podem-se citadas:

1. 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 8, apresenta a interface gráfica do *SWAT*.

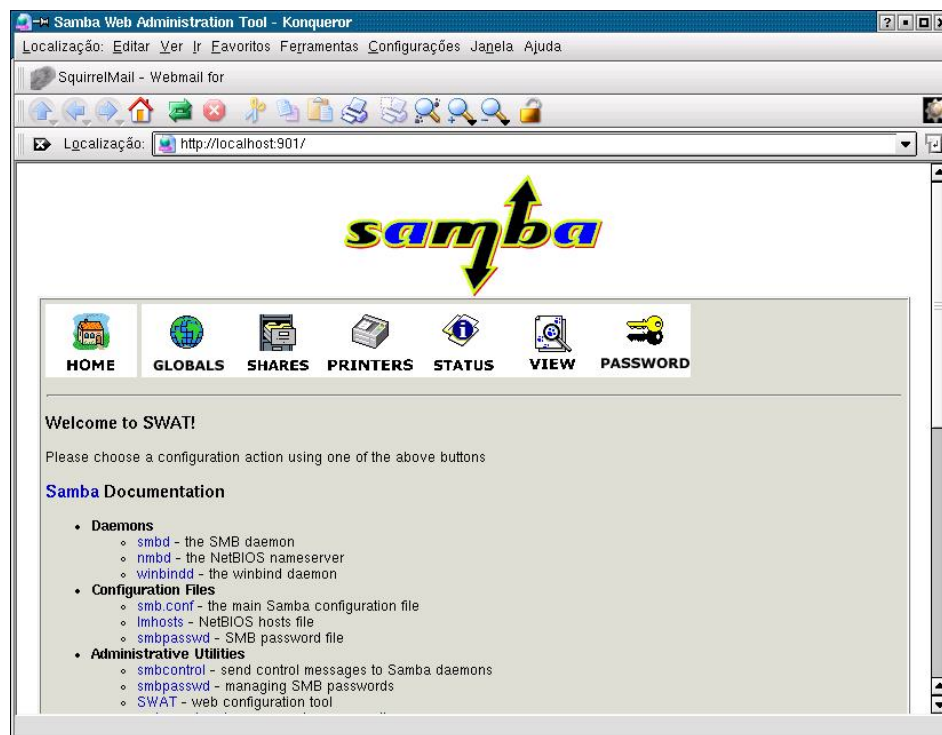


Figura 8: Interface do SWAT

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

No nível *Básico*, 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.

No nível *Avançado*, a ferramenta apresenta praticamente todos os parâmetros aceitos nos 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 ser colocada em 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 referente 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 9, mostra uma das interfaces da ferramenta com essas características.

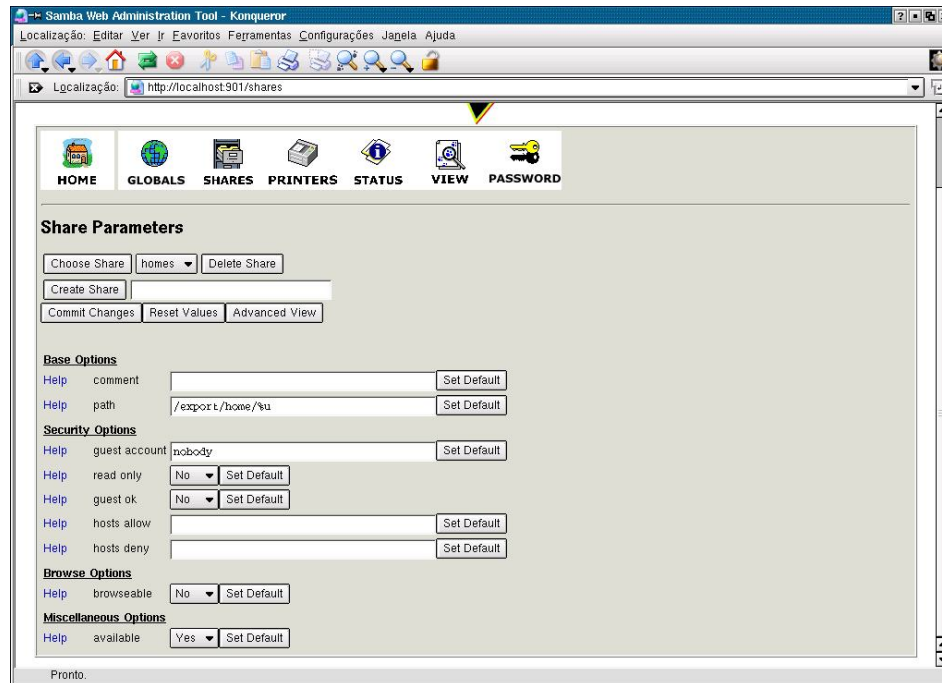


Figura 9: Interface de Formulários do SWAT

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 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. Neste sentido é possível mencionar um conjunto principal, como será apresentado na sequência, iniciada pelo:

2. 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 [23]. 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.
- **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 com os principais módulos que compõem cada serviço.

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

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: 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*.

Existem outras ferramentas com mesmas características encontradas no *IBM Tivoli*, como o caráter de gerência de *e-business* e possibilidades de acesso remoto, é possível ilustrar a Tabela 4 como quadro demonstrativo de algumas destas ferramentas, que no entanto também são proprietárias.

Com o *IBM Tivoli* e as ferramentas mencionadas na Tabela 4, 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

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 4: Ferramentas com Foco em Gerência de *e-business*

web que melhores se enquadram no perfil dos administradores computacionais, tendo como princípio a possibilidade de configuração de dispositivos e sistemas computacionais. Nesta linha podem-se apontadas:

3. 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*.

As Figuras 10 e 11, 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.

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

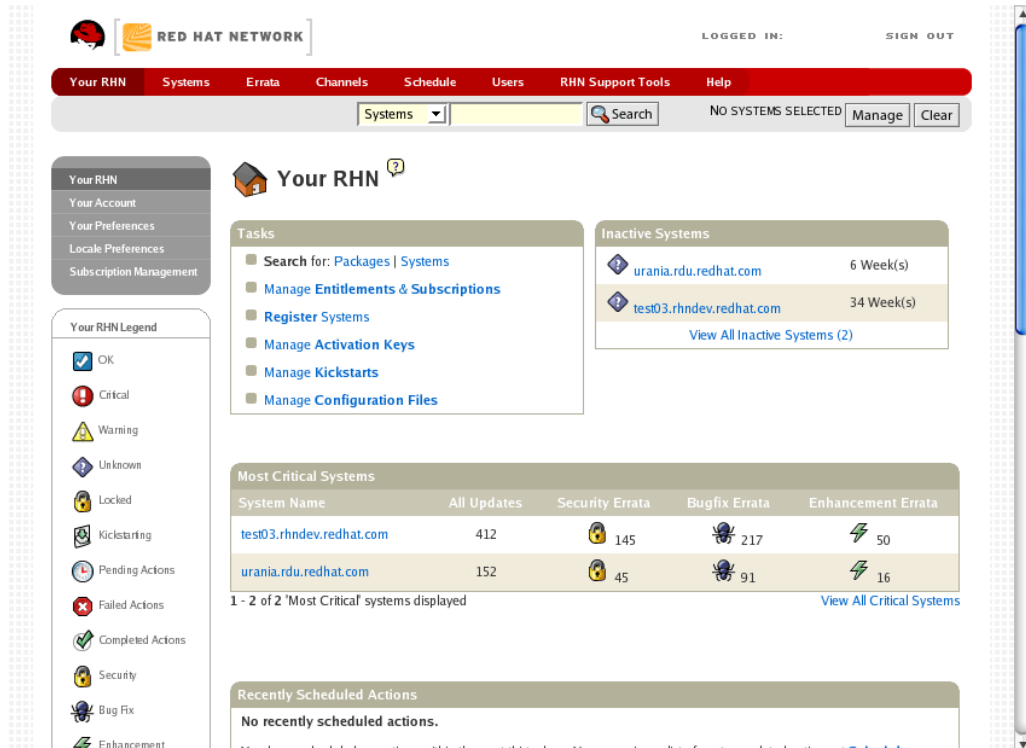


Figura 10: Interface do Red Hat Network

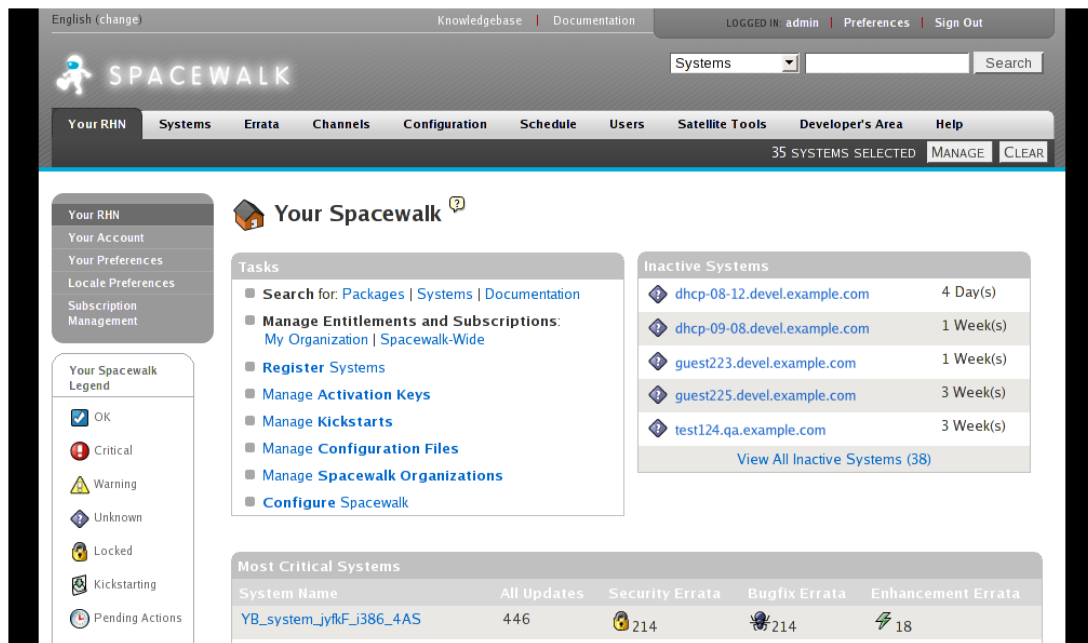


Figura 11: Interface do Spacewalk

a) Arquitetura do RHN-Spacewalk.

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 5:

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-existentes
<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 5: Funções dos Módulos 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*;

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

- (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 12 encontrada também na documentação de desenvolvedores, ilustrando o relacionamento e funcionalidade de cada módulo da ferramenta:

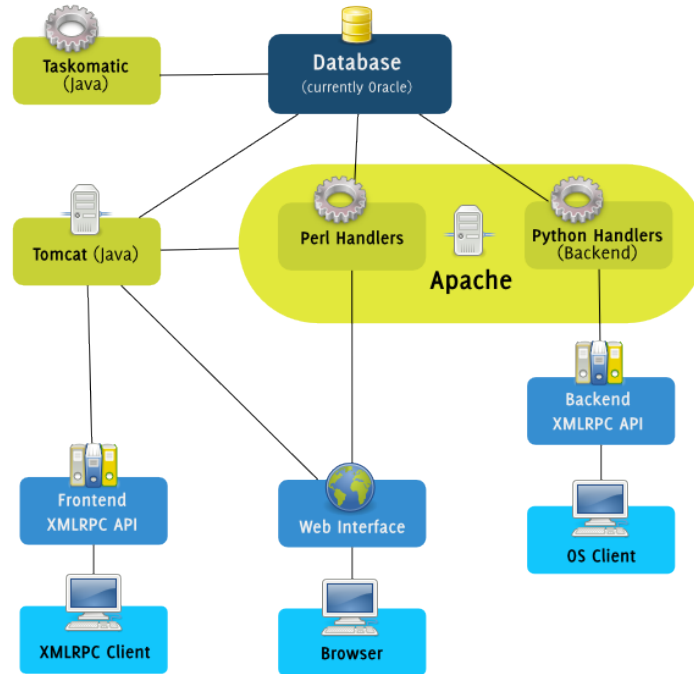


Figura 12: Arquitetura do Red Hat Network-Spacewalk

A Figura 12, 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-Spacwalk.

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-Spacwalk* pela internet, troca pacotes e informações utilizando uma interface gráfica WEB para facilitar a administração das operações. Acompanhe a Figura 13.

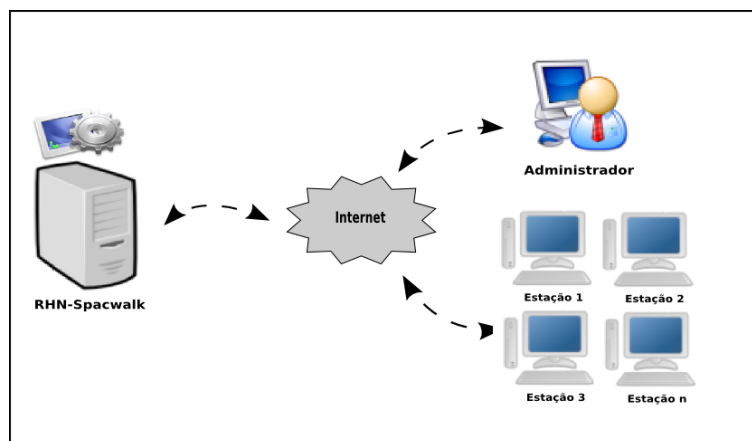


Figura 13: Funcionamento do RHN - Spacwalk no modelo Hosted

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 Figura14.

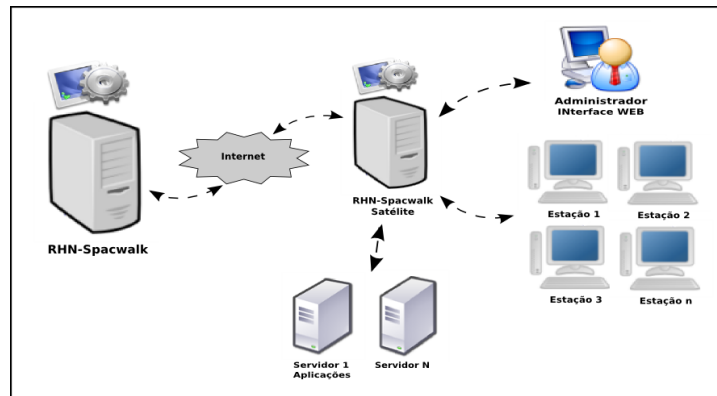


Figura 14: Funcionamento do RHN - Spacewalk no Modelo Satélite

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 15.

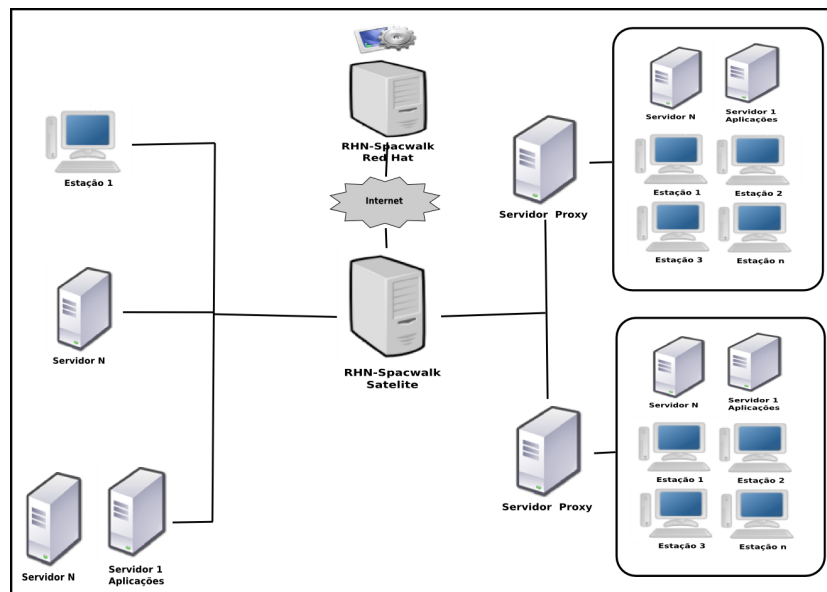


Figura 15: Funcionamento do RHN - Spacewalk no modelo Satélite com Proxy

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 15.

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

4. Webmin

O *Webmin* [24], 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* [11].

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 16, ilustra a interface gráfica acessível pela WEB, disponibilizada pelo *Webmin*.

Dentre as principais características do *Webmin* pode-se colocar em evidencia 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

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

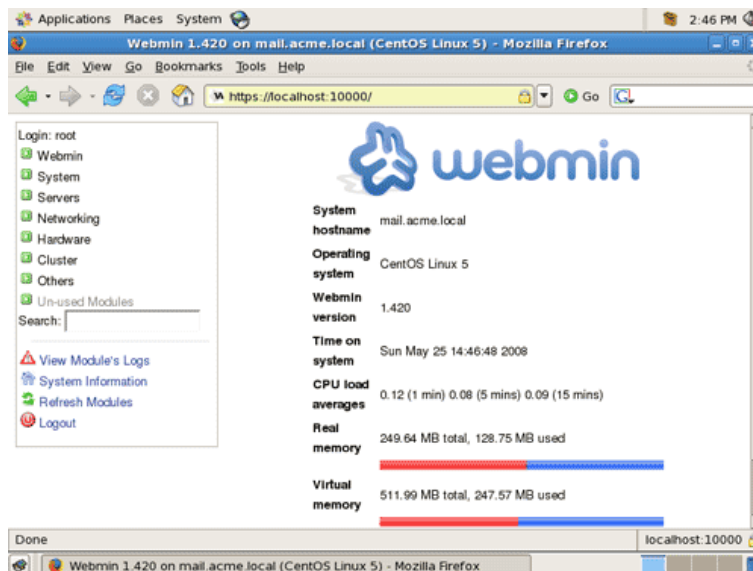


Figura 16: Interface do *Webmin*

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 [11, 5].

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[5]. 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*[5].

b) Organização

Como apresentado, a ferramenta de administração/configuração *Webmin*, é desenvolvida em módulos esse módulos por sua vez possuem a mesma estrutura, mas funcionalidades diferentes. Em uma instalação padrão do *Webmin*, os módulos e o resto do sistema se organizam sob dois (2) diretório 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 17 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.

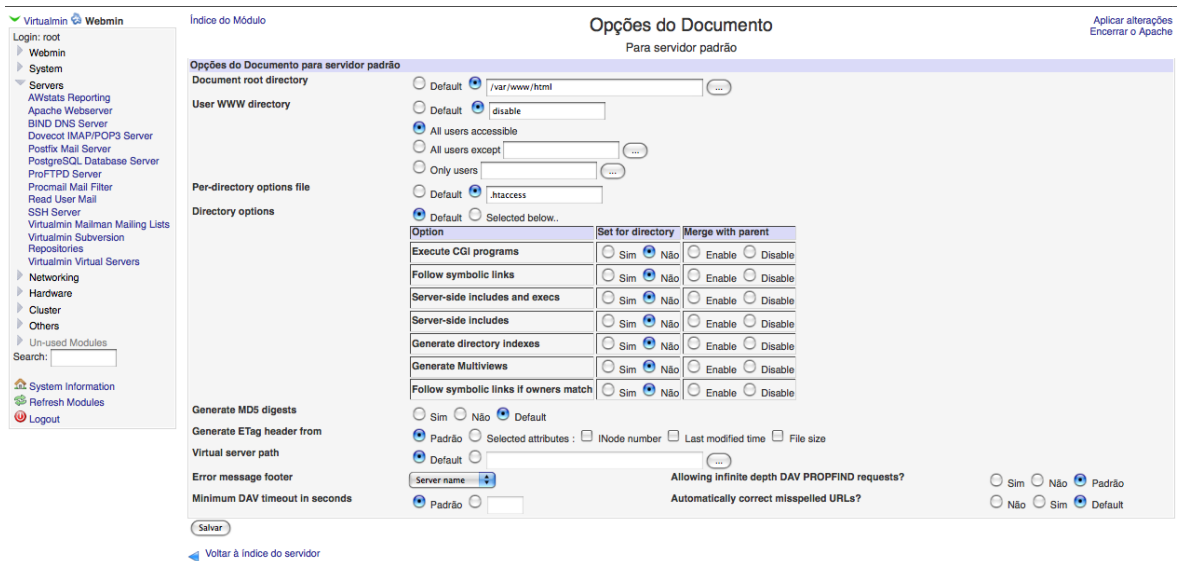


Figura 17: Formulário de Configuração do Servidor WEB pelo *Webmin*

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 18 ilustra o funcionamento do *Webmin*.

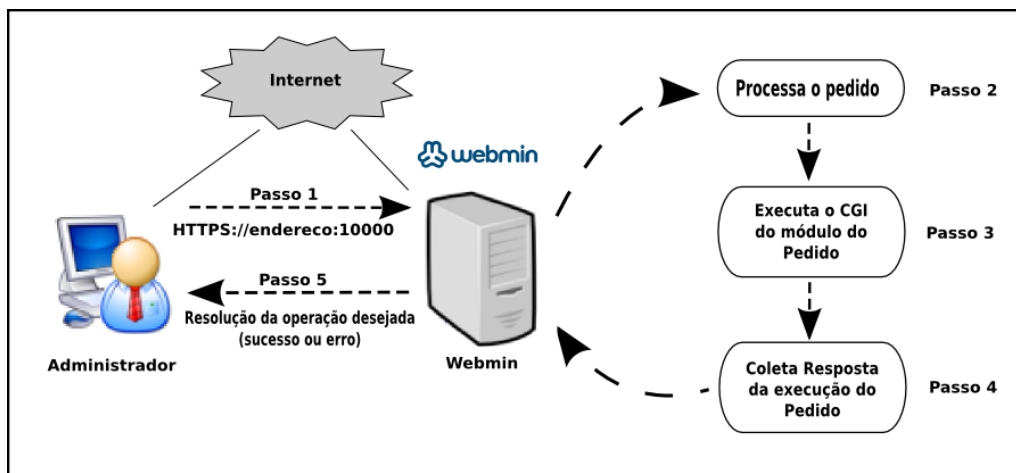


Figura 18: Esquema de Funcionamento do *Webmin*

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 6 e 7 demonstrada por Silva [5], 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 6: 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 <i>RPC</i> , o que dificulta o desenvolvimento de <i>Firewalls</i> , pois as comunicação sob <i>RPC</i> não possuem portas definidas de comunicação.

Tabela 7: 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 6, 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 examinadas neste trabalho.

Comparação entre Ferramentas de Administração de Acesso Remoto

Uma breve comparação, entre o *Webmin* e as outras ferramentas de administração de acesso remoto analisadas, é apresentada na sequência pela Tabela 8.

Caract./Ferram.	<i>Webmin:</i>	IBM Tivoli	RHN/Spacewalk
Foco da ferramenta	Administração Configuração	Administração e Gerência de Negócio	Atualização e gerencia- mento de Conteúdo de Software
Desenvolvimento modu- lar	Sim	Sim	Sim
Inserção de módulos no- vos	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 confi- guração 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 8: 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

satisfazerem essa premissa.

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

3 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 9, 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 de maior expressão 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 9, 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 previa	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 9: 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 a Modo Texto ou Gráfico, tanto de acesso local como de acesso remoto que realizam configurações de grupos de serviços ou serviços isolados, específicos, como configuração de interface de rede, criação de usuários, gerenciamento de base de dados, etc. Entretanto, as ferramentas neste trabalho apresentadas foram consideradas as mais populares e utilizadas pelos administradores para a realização de suas atividades.

4 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.

Devido a enorme variedade de atribuições a esse novo perfil de profissional, as 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 administração 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, desta forma, foi possível encontrar algumas características comuns e essenciais as ferramentas de administração computacional, como também, caracte-

terísticas peculiares a somente algumas delas.

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 estudo da ferramenta Webmin, a qual considera-se a mais madura e popular entre as ferramentas de administração.

Com base no estudo realizado, é importante ressaltar que a adoção e utilização de uma ferramenta como única solução para a administração computacional, passa por um processo intenso de avaliação contínua, em relação as reais necessidades do administrador em satisfazer os anseios de seus usuários e sua organização.

Referências

- [1] 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.
- [2] Alexandre Oliva and Paulo Lício de Geus. Makeconfig: Configuração diferenciada de máquinas de redes unix heterogênes. *WAIS*, 1996.
- [3] 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.
- [4] Robson Gomes de Melo. Proposta de um modelo de ferramenta para auxílio a administração computacional. Master's thesis, Instituto de Computação; Universidade Estadual de Campinas - UNICAMP ; Brasil, Dezembro 2009.
- [5] 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.
- [6] The Perl Foundation. Perl foundation. Website - <http://www.perl.org>, Abril 2008.
- [7] Mark Lutz. *Python Guia de Bolso*. Guia de Bolso. Alta Books, 3 edition, 2006.
- [8] Aurélio Marinho Jargas. *Shell Script Professional*, volume 1. Novatec, 2008.
- [9] Michae A. Cooper. Overhauling rdist for the '90s. *USENIX System Administration Conference (LISA VI)*; California, USA, Outubro 1992.
- [10] Harlander Magnus. Central system administration in a heterogeneous unix environment: Genuadmin. *USENIX System Administration Conference (LISA)*, 1994.
- [11] 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.

- [12] Mark Burgess. Cfengine: host configuration manager. *USENIX Computing system*, 1995.
- [13] Mark Burgess. Cfengine: a site configuration engine. *USENIX Computing systems*, 8(3), 1998.
- [14] 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.
- [15] Charles Beadnall ; Andrew Mayhew. Cfadmin: A user interface for cfengine. *Proceedings of the 15th System Administration Conference (USENIX/LISA)*, 2001.
- [16] Redney Gedda. Google gets slack with software updates. website – Computerworld – <http://www.computerworld.com.au/article/206619/>, Outubro 2006.
- [17] Juliet Kemp. Marionete: Manipulando maquinas com o puppet. *Linux Magazine*, 42, Maio 2008.
- [18] Equipe Conectiva. *Entendendo o Conectiva Linux*. Conectiva S.A., 2003.
- [19] Jacques Gélinas. Linuxconf documentacion. website - <http://www.solucorp.qc.ca/linuxconf/>, Janeiro 2005.
- [20] Dee-Ann LeBlanc. Linux system administration tools: Introduces linuxconf, webmin, yast and coas. *Linux Jornal*, Maio 2002.
- [21] YaST Novell Inc. Team. *YaST Documentation*. Novell Inc., 2006.
- [22] Inc. Novell. Simple yast module. website, http://en.opensuse.org/YaST/Tutorials/Simple_YaST_Module, Fevereiro 2008.
- [23] Edison T. L. Melo, Elvis M. Vieira, and Solange T. Sari. Implantação de um sistema de gerenciamento distribuído utilizando tivoli. *II Workshop RNP2*, 2000.
- [24] Jamie Cameron. *Managing Linux systems with Webmin*. Pearson Education, 2004.