



Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"

**GIOVANI DE SOUZA BUENO**

**IMPLEMENTAÇÃO DE INTERFACE PARA CONFIGURAÇÃO DE  
REGRAS DE FIREWALL**

**ASSIS**

**2014**

GIOVANI DE SOUZA BUENO

## IMPLEMENTAÇÃO DE INTERFACE PARA CONFIGURAÇÃO DE REGRAS DE FIREWALL

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientando(a): Giovani de Souza Bueno

Orientador(a): Me. Fábio Eder Cardoso

ASSIS

2014

## FICHA CATALOGRÁFICA

BUENO SOUZA, Giovani.

Implementação de interface para configuração de regras de firewall. Giovani de Souza Bueno. Fundação Educacional do Município de Assis – FEMA – Assis, 2014.

52 p.

Orientador: Fábio Eder Cardoso

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA

CDD: 001.61

Biblioteca da FEMA

# IMPLEMENTAÇÃO DE INTERFACE PARA CONFIGURAÇÃO DE REGRAS DE FIREWALL

GIOVANI DE SOUZA BUENO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientador(a): \_\_\_\_\_

Analizador(1): \_\_\_\_\_

ASSIS

2014

## DEDICATÓRIA

Dedico este trabalho a minha família, amigos, professores e a todos que confiaram em mim.

## AGRADECIMENTOS

Agradeço a Deus, a minha família pelo apoio e compreensão, aos meus amigos que estiveram presentes nessa jornada, e a todos que contribuíram e fizeram parte dessa conquista.

“A menos que modifiquemos a nossa maneira de pensar, não seremos capazes de resolver os problemas causados pela forma como nos acostumamos a ver o mundo”.  
(Albert Einstein)

## RESUMO

A segurança da informação tem sido a principal preocupação das corporações, que utilizam várias ferramentas para proteger suas informações. Entretanto, as técnicas e ferramentas utilizadas não conseguem acompanhar a evolução de uso do poder computacional.

Na tentativa de auxiliar os administradores de rede, foi desenvolvido várias ferramentas com intuito de facilitar a configuração e administração de políticas de segurança.

Neste contexto, este trabalho tem como objetivo apresentar um estudo das ferramentas de administração de regras de *Firewall*, analisando-as e comparando-as, propondo também um novo modelo de ferramenta de administração buscando solucionar as limitações encontradas nas ferramentas estudadas.

Palavras-chaves: IPTABLES; Segurança; Linux; Firewall, *IPTABLES*; Regras.

## ABSTRACT

Information security has been a primary concern of corporations, which use various tools to protect your information. However, the techniques and tools used can not follow the evolution of the use of computational power.

In an attempt to assist network administrators with various tools to facilitate the configuration and administration of security policies has been developed.

In this context, this work aims to present a study of the management of Firewall rules tools, analyzing and comparing them also proposing a new model of administration tool seeking to solve the limitations found in the studied tools.

Keywords: IPTABLES; Security; Linux; Firewall; *IPTABLES*; Rules.

## LISTA DE ILUSTRAÇÕES

Figura 1– Tela adicionando uma regra no Linux .....	21
Figura 2 – Interface da Ferramenta KMyFirewall .....	22
Figura 3 – Arquitetura do modelo proposto. ....	25
Figura 4- UC 01: Movimentação (Visão Geral).....	36
Figura 5 – UC 02: Efetuar Login.....	37
Figura 6 – UC 03: Cadastrar Usuário .....	38
Figura 7 – UC 04: Cadastrar Firewall .....	39
Figura 8 – UC 05: Cadastrar Interfaces.....	40
Figura 9 – UC 06: Manter Regras .....	41
Figura 10 – UC 07: Imprimir relatório Regras.....	42
Figura 11 – Diagramas de Sequência: Cadastrar Firewall .....	43
Figura 12 – Modelo Entidade e Relacionamento.....	44
Figura 13 – Diagrama de Classes .....	45
Figura 14 - Diagrama de Atividades: Cadastro de firewall. ....	46
Figura 15 – Diagrama de Atividades: Cadastro Interface.....	47
Figura 16 – Diagrama de Atividades: Cadastro de Regras .....	48
Figura 17 – Diagrama de Atividades: Cadastrar Usuários. ....	49

## LISTA DE TABELAS

Tabela 1 – Estimativa de duração das etapas de desenvolvimento.....	32
Tabela 2- Cronograma .....	33
Tabela 3 -Orçamento do Projeto .....	35
Tabela 4 – Efetuar Login .....	37
Tabela 5 – Cadastrar Usuários.....	38
Tabela 6 – Cadastrar Firewall .....	39
Tabela 7 – Cadastrar Interfaces .....	40
Tabela 8 – Manter Regras.....	41
Tabela 9 – Imprimir Relatório de Regras.....	42

## LISTA DE ABREVIATURAS E SIGLAS

<b>TOS</b>	Type Of Service
<b>SQL</b>	Structure Query Language
<b>UC</b>	Use Case
<b>PHP</b>	Hypertext Preprocessor
<b>IP</b>	Internet Protocol
<b>HTTP</b>	Hypertext Transfer Protocol
<b>UDP</b>	User Datagram Protocol
<b>FIFO</b>	First In First Out

## SUMÁRIO

<b>CAPÍTULO 1 – INTRODUÇÃO .....</b>	<b>14</b>
1.1 IDENTIFICAÇÃO DO PROBLEMA.....	14
1.2 OBJETIVO.....	15
1.3 PÚBLICO ALVO .....	15
1.4 JUSTIFICATIVA .....	15
<b>CAPÍTULO 2 – LEVANTAMENTO BIBLIOGRÁFICO .....</b>	<b>16</b>
2.1 FIREWALL .....	16
2.2 TIPOS DE FIREWALL.....	16
2.3 O FIREWALL IPTABLES.....	18
2.3.1 TABELA FILTER .....	18
2.3.2 TABELA NAT.....	19
2.3.3 TABELA MANGLE .....	19
2.3.4 PRINCIPAIS COMANDOS DE IPTABLES.....	20
2.4 FERRAMENTAS DE ADMINISTRAÇÃO DE FIREWALL.....	20
2.4.1 WEBMIN.....	21
2.4.2 KMYFIREWALL.....	22
2.4.3 PONTOS POSITIVOS .....	23
2.4.4 PONTOS NEGATIVOS.....	23
<b>CAPÍTULO 3 - PROPOSTA DE UM MODELO DE FERRAMENTA</b>	<b>24</b>
3.1 ARQUITETURA DO MODELO.....	25
<b>CAPITULO 4 - METODOLOGIA DE ANÁLISE.....</b>	<b>27</b>
<b>CAPÍTULO 5 – TECNOLOGIAS E FERRAMENTAS UTILIZADAS</b>	<b>29</b>
5.1 APTANA STUDIO 3 (PHP).....	29
5.2 MYSQL.....	29
5.3 PHP .....	29

5.4 BOOTSTRAP .....	30
<b>CAPÍTULO 6 – ESPECIFICAÇÃO DOS REQUISITOS.....</b>	<b>31</b>
6.1 LEVANTAMENTO DOS REQUISITOS .....	33
6.2 LISTA DE EVENTOS .....	33
6.3 ESTIMATIVA DE DURAÇÃO DAS ETAPAS DE DESENVOLVIMENTO ...	33
6.4 CRONOGRAMA.....	33
6.5 RECURSOS NECESSARIOS .....	34
6.6 ESTIMATIVA DE CUSTOS .....	35
6.7 ORÇAMENTO DO PROJETO.....	35
6.8 MOVIMENTAÇÕES GERAIS:.....	36
6.9 CASOS DE USO: EFETUAR ACESSO.....	37
6.10 CASOS DE USO: CADASTRAR USUÁRIO.....	38
6.11 CASOS DE USO: CADASTRAR FIREWALL .....	39
6.12 CASOS DE USO: CADASTRAR INTERFACES .....	40
6.13 CASOS DE USO: MANTER REGRAS.....	41
6.14 CASOS DE USO: IMPRIMIR RELÁTORIOS REGRAS .....	42
<b>7- DIAGRAMAS DO SISTEMA .....</b>	<b>43</b>
7.1 DIAGRAMAS DE SEQUÊNCIA: CADASTRAR FIREWALL.....	43
7.2 MODELO DE ENTIDADE RELACIONAMENTO .....	44
7.3 DIAGRAMA DE CLASSES.....	45
7.4 DIAGRAMA DE ATIVIDADES – CADASTRO DE FIREWALL. ....	46
7.5 DIAGRAMA DE ATIVIDADES – CADASTRO DE INTERFACE.....	47
7.6 DIAGRAMA DE ATIVIDADES – CADASTRO DE REGRAS .....	48
7.7 DIAGRAMA DE ATIVIDADES - CADASTRAR.....	49
<b>CONCLUSÃO .....</b>	<b>50</b>
<b>REFERÊNCIAS.....</b>	<b>51</b>

## CAPÍTULO 1 – INTRODUÇÃO

À medida que a rede mundial de computadores cresce, cresce também o número de ataques e invasões, que exploram as falhas de segurança presentes nos sistemas computacionais, tornando-se problema sério para organizações e até mesmo usuários comuns.

Um dos principais mecanismos de segurança capaz de minimizar os riscos de ataques é o *firewall*, a partir de um conjunto de regras e instruções, o *firewall* analisa o tráfego de rede para determinar quais operações de transmissão ou recepção de dados podem ser executados.

Mesmo com a crescente evolução tecnológica, o *firewall* continua a empregar os mesmos conceitos, tendo a evolução apenas de algumas funcionalidades, deste modo os objetivos específicos desse trabalho são, estudar as ferramentas de administração de *firewall* existentes e apresentar uma proposta de ferramenta capaz de suprir as necessidades encontradas nas ferramentas estudadas.

### 1.1 IDENTIFICAÇÃO DO PROBLEMA

Segundo Neto (2004, p.17):

No Linux as funções do *firewall* são agregadas a própria arquitetura do *kernel*, isso o torna superior aos demais concorrentes. Enquanto a maioria é definida como subsistema, o *linux* possui a capacidade de transformar o *firewall* no próprio sistema.

Para a configuração de um *firewall IPTABLES* é preciso realizar uma série de comandos que são interpretados pelo *kernel* do sistema operacional, esses comandos podem ser executados em scripts (arquivos - texto) ou diretamente no *Shell*.

Entretanto, a criação e configuração dessas regras se torna mais difícil pois exige um conhecimento maior do administrador do sistema, que precisa utilizar de ferramentas para sua configuração, ferramentas essas muitas vezes pagas e com recursos limitados.

## 1.2 OBJETIVO

Este projeto tem como objetivo realizar o estudo das principais ferramentas de configuração de *firewall IPTABLES*, e propor um novo modelo de ferramenta buscando solucionar as limitações encontradas nas ferramentas estudadas.

Pretende-se também apresentar a eficácia da ferramenta em ambiente real criando regras de controle *Firewall IPTABLES*.

## 1.3 PÚBLICO ALVO

O sistema será utilizado por empresas, profissionais da área, e servirá como estudo para alunos, professores e interessados no assunto.

## 1.4 JUSTIFICATIVA

Devido a grande dificuldade de compreensão e configuração de um *Firewall IPTABLES* e o grande número de regras necessárias para sua configuração, o processo de administração do *firewall* torna-se mais difícil.

A principal justificativa para realização desse projeto é deixar mais fácil a configuração de um *Firewall IPTABLES* que hoje é feito manualmente via *Shell* e de difícil compreensão.

Este projeto irá contribuir para a construção de uma ferramenta que facilite essa configuração.

## CAPÍTULO 2 – LEVANTAMENTO BIBLIOGRÁFICO

### 2.1 FIREWALL

O *Firewall* como o nome sugere (parede de fogo) é uma barreira tecnologia entre dois pontos de uma rede, onde normalmente é o único ponto de acesso entre a rede interna e a internet.

Segundo Neto (2004, p.10), “podemos afirmar que um *Firewall* é um programa que detém autonomia concedida pelo próprio sistema para pré-determinar e disciplinar todo o tipo de tráfego existente entre o mesmo e outros hosts/redes”.

O *Firewall* pode ser tanto software quanto hardware, o *firewall* pode se localizar tanto no host quanto na rede. Os localizados em máquinas é conhecidos como firewall pessoal podendo proteger apenas a máquina que foi instalado, já os de rede são mais complexos protegendo uma rede inteira.

### 2.2 TIPOS DE FIREWALL

Os *firewalls* podem ser classificados da seguinte forma:

- Filtro de Pacotes

O *firewall* filtro de pacotes possui a capacidade de analisar cabeçalhos (*Headers*) enquanto os mesmos trafegam, baseando-se apenas em campos básicos de cabeçalhos dos protocolos, a filtragem pode simplesmente deixar tal pacote trafegar na rede livremente ou simplesmente parar sua trajetória.

Segundo Neto (2004, p.11) “É sem dúvida a classe mais utilizada do firewall”.

- Firewall NAT

De acordo com Neto (2004, p.12), o *firewall* NAT tem a capacidade de manipular a rota padrão dos pacotes que passam pelo *Kernel* do *Firewall*, utilizando o conceito de “tradução de endereço”, o que agrega algumas funcionalidades, sendo as principais manipular o endereço de origem (SNAT<sup>1</sup>) e destino (DNAT<sup>2</sup>), realizar o *masquerading*, mascaramento de endereço IP.

- Firewall Híbrido

Conforme Neto (2004, p.13), esse tipo de *firewall* agrega funções de filtro de pacotes como também de NAT, não sendo somente a união de dois tipos, mas um tipo isolado com propriedades próprias.

Este tipo de *firewall* também implementa o mascaramento, o SNAT e DNAT.

- Firewall Proxy

Segundo Equipe Conectiva (2001, p.133), “o servidor *Proxy* tem como principal objetivo possibilitar a máquina de rede privada acessar a rede pública, como a internet, sem que exista uma ligação direta entre a rede privada e pública.”

O *proxy* pode ser considerado um *firewall* de camada de aplicação de modelo de referência TCP/IP, pois atua diretamente com os protocolos de transporte e de aplicação.

---

<sup>1</sup> SNAT – Sua funcionalidade é de manipular o endereço de origem.

<sup>2</sup> DNAT –Disponível no firewall NAT sua funcionalidade é alterar o endereço de destino de uma rota final, ou realizando a função inversa para eventuais respostas.

## 2.3 O FIREWALL IPTABLES

Segundo Neto (2004, p.25), “o *IPTABLES* trata-se, na verdade, de uma ferramenta de *Front-End* para lhe permitir manipular as tabelas do *Netfilter* embora o mesmo seja constantemente confundido com um *firewall* por si só”.

O *IPTABLES* foi concebido em 1994 por Rusty Russel em colaboração com Michel Neuling, e faz parte do *kernel* 2.4. Ele funciona com base em porta/endereço de destino/origem e informações contidas no cabeçalho IP.

Dividido em três tabelas que são subdivididas por sua vez em listas de regras, denominadas *chains*. As tabelas são *Filter*, *Network Address Translation* (NAT) e *Mangle*.

### 2.3.1 TABELA FILTER

Segundo Neto (2004, p.20), “está é a tabela padrão do *IPTABLES* e é dividida em três *chains*, sendo utilizada para um firewall de filtro de pacotes”.

- **INPUT:** é a regra que faz a filtragem do tráfego de entrada no firewall, tudo que entra no *Host*.
- **FORWARD:** é a regra que faz filtragem do tráfego que passa pelo *firewall*, tudo que chega ao *host* mas deve ser redirecionado a um *host* secundário ou outra interface de rede.
- **OUTPUT:** é a regra que faz filtragem de saída do *firewall*, tudo que sai do *Host*.

### 2.3.2 TABELA NAT

Segundo Neto (2004, p.20), “está tabela implementa funções de *NAT*(*Network Address Translation*) ao host do Firewall, e também é dividida em três *chains* que são:

- **PREROUTING:** utilizada quando há necessidade de se fazer alterações em pacotes antes que os mesmos sejam roteados.
- **OUTPUT:** é a regra que bloqueia ou libera todos os pacotes com origem no host firewall, trata os pacotes emitidos pelo *host Firewall*.
- **POSTROUTING:** utilizado quando há necessidade de se fazer alterações em pacotes após o tratamento de roteamento.

### 2.3.3 TABELA MANGLE

De acordo com Neto (2004, p.21), essa tabela implementa alterações especiais em pacotes em um nível mais complexo. Sendo capaz de alterar a propriedade de entrada e saída de um pacote baseado no tipo de serviço *Type of Service (TOS)*<sup>3</sup>. A tabela *Mangle* é composta por cinco *chains* que são:

- **PREROUTING:** altera ou marca os pacotes antes do roteamento.
- **INPUT:** altera ou marca os pacotes de entrada no *host firewall*.
- **FORWARD:** altera ou marca os pacotes que são redirecionados outro *host* ou interface de rede.
- **OUTPUT:** altera ou marca os pacotes gerados localmente antes de serem roteados.
- **POSTROUTING:** altera ou marca os pacotes após o roteamento.

---

<sup>3</sup> TOS é um campo no cabeçalho do protocolo IP versão 4, que tem a finalidade de especificar qual prioridade do pacote, é a informação do cabeçalho *IP* que possui o tamanho de 8 bits.

### 2.3.4 PRINCIPAIS COMANDOS DE IPTABLES

Existem algumas comandos, que são empregados para a construção de regras de *firewall IPTABLES*.

Políticas *IPTABLES*:

- *Accept*: aceita o pacote recebido;
- *Drop*: nega pacote e não retorna nenhuma mensagem de volta;
- *Reject*: nega o pacote e retorna um aviso de erro ao emissor;

Comandos principais utilizados pelo *IPTABLES*:

- -A: adiciona ou atualiza uma regra no fim;
- -I: apenas adiciona uma nova regra no início;
- -D: exclui uma regra específica;
- -P: define a regra padrão
- -L: lista todas as regras armazenadas;
- -F: exclui todas as regras;
- -X: exclui uma regra com nome específico.

Parâmetros padrões:

- -p: define qual protocolo deve ser tratado (*TCP, UDP*);
- -s ou -d : define o endereço de origem ou de destino em que a regra irá atuar;
- -i : define a interface de rede por onde os pacotes são recebidos e enviados;
- -j : define a direção de uma ação baseada em regras similares.

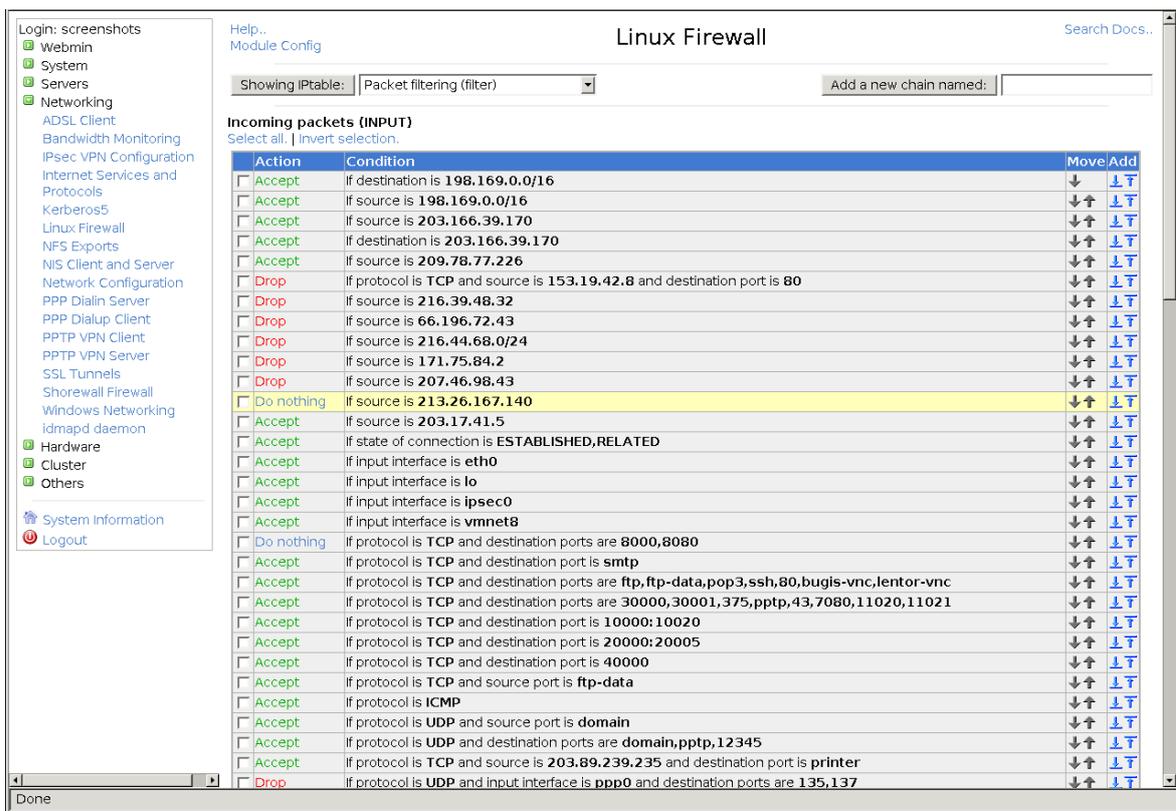
### 2.4 FERRAMENTAS DE ADMINISTRAÇÃO DE FIREWALL

Para a realização deste trabalho, foi feito um estudo de duas ferramentas de gerenciamento de regras e controle de *Firewall*, onde foram destacados os seus pontos positivos e negativos. Não foram pesquisados sistemas de grande porte pois necessitam ser adquiridos comercialmente.

## 2.4.1 WEBMIN

*Webmin*<sup>4</sup> é uma ferramenta que foi desenvolvida por Jamie Cameron em 1997, que tem como objetivo configurar e administrar sistemas e redes de computadores. O *webmin* possui suporte para diversas tarefas que vão desde configurar contas de usuários até criação de *Cluster* ou *RAID*, tornando a ferramenta poderosa.

A aplicação tem como objetivo administrar uma rede totalmente por meio de uma interface integrada, evitando que o usuário necessite editar arquivos de configuração ou executar comandos em console texto, conforme ilustra a figura 1.



**Figura 1– Tela adicionando uma regra no Linux. Fonte: (<http://www.webmin.com/screenshots/chapter19/figure2.png>)**

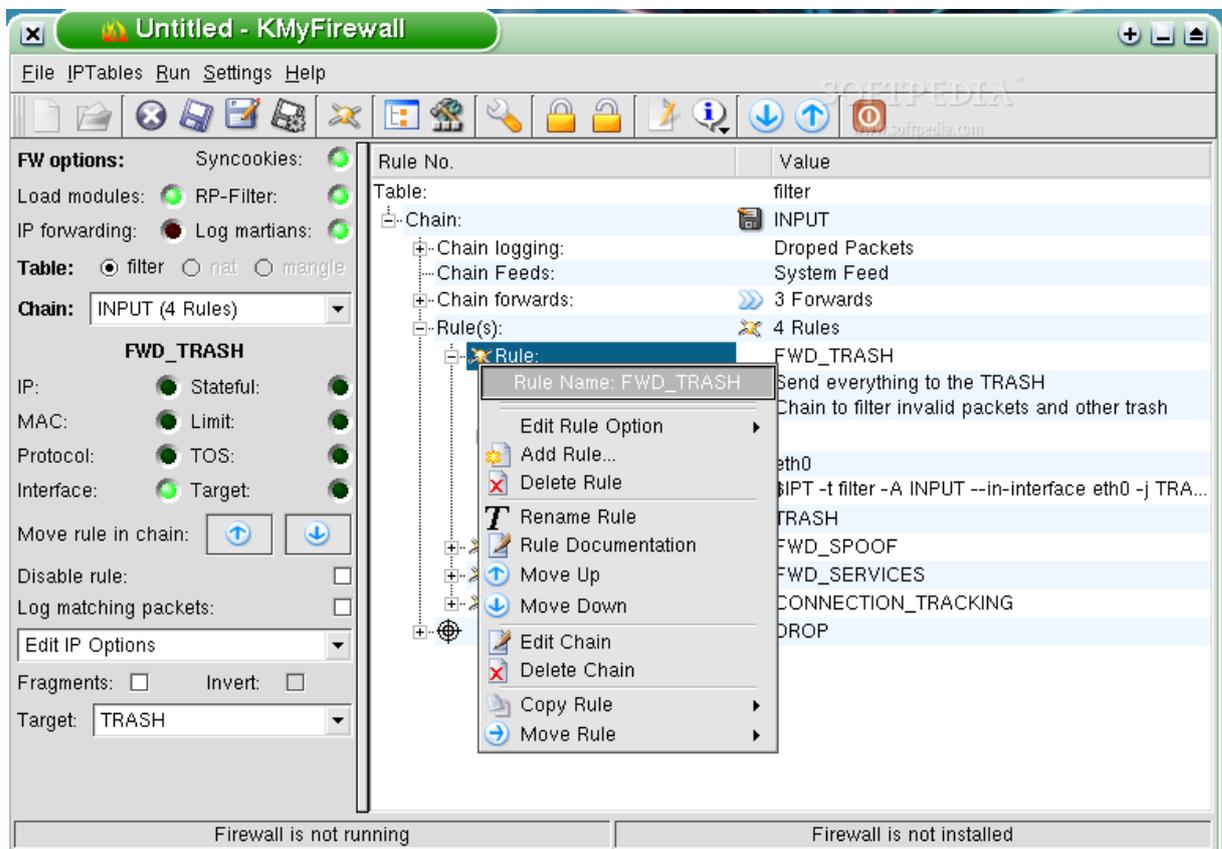
<sup>4</sup> <http://www.webmin.com/>

## 2.4.2 KMYFIREWALL<sup>5</sup>

É uma ferramenta gráfica para manipular regras do *IPTABLES*, feito para trabalhar em ambiente desktop do KDE, desenvolvido com o objetivo de facilitar a configuração do firewall, porém não dispensa conhecimentos básicos.

Projetado para apenas gerar scripts de *IPTABLES* e instalar na inicialização do sistema, sua configuração é estática o que significa que depois que o conjunto de regras é criado não se pode alterar.

Seu ponto negativo está em sua interface, sendo totalmente em inglês, tornando complexo a criação das regras *IPTABLES*.



**Figura 2<sup>6</sup> – Interface da Ferramenta KMyFirewall**  
(Fonte: <http://linux.softpedia.com/progScreenshots/KMyFirewall-Screenshot-2412.html>)

<sup>5</sup> <http://www.kmyfirewall.org/>

<sup>6</sup> A figura 2 acima mostra a tela de inclusão de regra no sistema *KmyFirewall*, onde podemos notar que sua linguagem é totalmente em inglês e sua interface complexa.

### 2.4.3 PONTOS POSITIVOS

De acordo com a pesquisa levantada foram encontrados diversos pontos positivos que são:

- **Suporte a diversos serviços:** O *Webmin* provê de diversos serviços disponíveis que podem ser necessários ao administrar a rede, sendo que não é necessário manipular arquivos de configuração e nem acessar diferentes interfaces.
- **Licença Aberta:** Licenciado pela *GPL*, o que o torna um programa aberto com gratuidade, tendo seu código fonte aberto para que pessoas possam aperfeiçoá-lo.
- **Modularidade:** As funcionalidades da ferramenta são desenvolvidas por módulos independentes entre si.
- **Segurança:** Possui autenticação de usuários e *SSL* para conexões *HTTP* seguras (*HTTPS*) tornando seu acesso mais seguro.
- **Acesso remoto:** Permite configurar servidores remotamente onde se encontra instalada a ferramenta.
- **Linguagem de Script:** Como o *Webmin* é todo escrito em *Perl*, sua modificação e correção se torna mais simples.

### 2.4.4 PONTOS NEGATIVOS

Assim como foi encontrado vários pontos positivos, o *Webmin* também tem vários pontos negativos e eles são:

- **Ausência de banco de dados:** O *Webmin* grava todos seus dados em arquivos, o que dificulta na rapidez da pesquisa dos dados. Ao gravar os dados coletados em arquivos, estes ficam vulneráveis a perdas se mal configurados.
- **Carregamento de Módulos:** Ao instalar o *Webmin*, todos os módulos são instalados, inclusive módulos que o administrador não necessita.

- **Ausência de Otimização:** Por ser desenvolvido totalmente em Perl, existe um *overhead* de informações e um gasto desnecessário de processamento ao manipular o arquivo texto que possui a lógica do sistema.
- **Criação de módulos:** Não seguem um padrão de desenvolvimento dificultando o entendimento e desenvolvimento de novos módulos.

### **CAPÍTULO 3 - PROPOSTA DE UM MODELO DE FERRAMENTA**

A partir do estudo realizado com as principais ferramentas de administração computacional, foi possível evidenciar características importantes que não foram atendidas pelas ferramentas estudadas.

E essa situação justifica a proposta de um novo modelo de ferramenta, que satisfaça as necessidades do cenário atual, entre elas é possível mencionar:

- **Interface gráfica de acesso remoto:** Para auxiliar no trabalho de criação e gerencia de regras firewall é imprescindível a utilização de interface gráfica, tornado seu processo mais seguro evitando edição manuais em arquivos de configuração. No entanto ferramentas gráficas com acesso remoto são poucas que satisfazem esse requisito, muitas com interfaces complexas e mal elaboradas, outras totalmente em inglês.
- **Ferramenta não necessita de Instalação:** Ferramenta que possibilita ao administrador de rede administrar diversos servidores firewall, e em máquinas remotas são imprescindíveis, porém as principais ferramentas não suprem essa necessidade, tendo o administrador que instalar a ferramenta em cada *Firewall* que for administrar, ou utilizar ferramentas instaladas em máquinas cliente, que ficam retornando informações periodicamente para a ferramenta centralizadora, local onde se encontra o administrador, isso se torna inadequada em função do grande fluxo de dados trafegados nos links de dados.
- **Dados gravados em banco de dados:** As principais ferramentas estudadas salvam seus dados em arquivos, tornado lento sua busca de acordo com o aumento de informações gravadas, ficando esses dados vulneráveis a perdas

ou erros. Gravar essas informações é de suma importância para dar maior controle e edição da política de segurança do *firewall*.

- **Interface Responsiva:** Verificando o cenário atual, é importante que o sistema possa ser utilizado em diferentes dispositivos, dando maior praticidade e inovação a ferramenta.

### 3.1 ARQUITETURA DO MODELO

O modelo de ferramenta de administração proposto prevê seu desenvolvimento em camadas ou módulos. Tornando a ferramenta mais organizada com cada módulo/camada mais autônomas.

A arquitetura bem como suas camadas/módulos seguem de acordo com o diagrama apresentado na Figura 2.

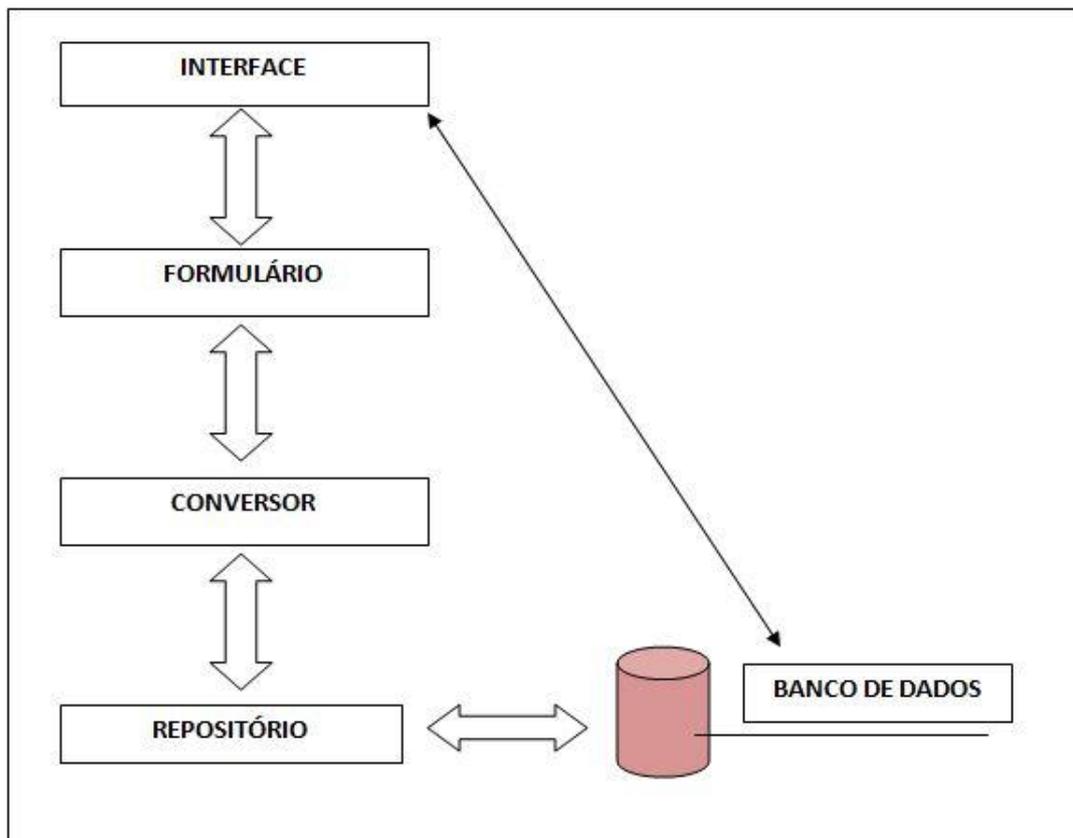


Figura 3 – Arquitetura do modelo proposto.

Assim como pode ser visto na Figura 3, o sistema terá 4 camadas todas com funções específicas que serão descritas na sequência :

- **INTERFACE**

Como o objetivo da ferramenta é ser utilizada na web, essa camada é a responsável pela configuração de tudo que for apresentado para ao usuário, desde cores, temas específicos até validação de campos, evitando que o usuário cometa erros de preenchimento de campos dos formulários, referentes aos parâmetros de configuração.

- **FORMULÁRIO**

Essa camada será responsável pela geração de formulários WEB, formulário esses que o administrador edita ou preenche os valores referentes a configuração desejada.

- **CONVERSOR**

O conversor será a camada responsável por tratar os dados vindos dos formulários gerando o arquivo responsável com todas as regras daquele firewall e repassando para a camada seguinte (Repositório) e gravando essas regras no banco de dados.

- **REPOSITÓRIO**

O repositório é a camada responsável por armazenar os arquivos de configuração do *firewall*, podendo o administrador editar a qualquer momento, caso ocorra algo no servidor, esse repositório serve de backup.

## CAPITULO 4 - METODOLOGIA DE ANÁLISE

O sistema foi modelado utilizando a metodologia de Análise Orientada a Objetos. Para essa análise foram utilizados conceitos de *Unified Modeling Language* (UML), que é uma linguagem de modelagem que ajuda a resumir os problemas do sistema, para obter uma visão. mais detalhada de como é o seu funcionamento.

- CASO DE USO

Um diagrama de caso de uso mostra um conjunto de casos de uso e atores - um tipo especial de classe - e seus relacionamentos. Esses diagramas são aplicados para ilustrar a visão estática do caso de uso de um sistema. Os diagramas de caso de uso são importantes principalmente para a organização e modelagem dos comportamentos de um sistema. (GRADY BOOCH, J. R.& IVAR J.)

- DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência é um diagrama comportamental que procura determinar a sequência de eventos que ocorrem em um determinado processo, identificando quais mensagens devem ser disparadas entre os elementos envolvidos e em que ordem (GRADY BOOCH, J. R.& IVAR J.)

- MODELO ENTIDADE E RELACIONAMENTO

Os relacionamentos ligam as classes/objetos entre si, criando relações lógicas entre estas as entidades. Os relacionamentos podem ser dos seguintes tipos:

Associação: é uma conexão entre classes, e em UML, uma associação é definida com um relacionamento que descreve uma série de ligações.

Generalização: É um relacionamento de um elemento mais geral e outro mais específico. O elemento mais específico pode conter apenas informações adicionais.

Dependência e Refinamentos: Dependência é um relacionamento entre elementos, um independente e outro dependente (GRADY BOOCH, J. R.& IVAR J.)

- **DIAGRAMA DE CLASSES**

Um diagrama de classes mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos. Os diagramas de classes são os diagramas mais encontrados em sistemas de modelagem orientados a objetos. Use esses diagramas para ilustrar a visão estática do projeto de um sistema. Os diagramas de classes que incluem classes ativas são empregados para direcionar a visão estática do processo de um sistema (GRADY BOOCH, J. R.& IVAR J.)

- **DIAGRAMA DE ATIVIDADES**

Um diagrama de atividades exibe o fluxo de uma atividade para outra em um sistema. Uma atividade apresenta um conjunto de atividades, o fluxo sequencial ou ramificado de uma atividade para outra e os objetos que realizam ou sofrem ações.

Os diagramas de atividades são utilizados para ilustrar a visão dinâmica de um sistema. São importantes principalmente para fazer a modelagem da função de um sistema. Os diagramas de atividades dão ênfase ao fluxo de controle na execução de um comportamento (GRADY BOOCH, J. R.& IVAR J.)

## CAPÍTULO 5 – TECNOLOGIAS E FERRAMENTAS UTILIZADAS

### 5.1 APTANA STUDIO 3 (PHP)

Aptana<sup>7</sup> Studio é uma IDE de código aberto para desenvolvimento web, está ferramenta tem suporte para diferentes linguagens de programação como por exemplo: PHP, Ruby, Python, CSS3, HTML5 e JavaScript.

O Aptana Studio foi adquirida pela empresa Appcelerator, uma das principais empresas para desenvolvimento multi-plataforma do mercado, sua última versão é a 3.2.4 e tem suporte a vários sistemas operacionais. (Appcelerator Inc., 2013)

### 5.2 MYSQL

O MYSQL<sup>8</sup> é um gerenciador de banco de dados relacional, onde armazena dados em tabelas separadas embora todos dos dados estejam armazenados em só local, proporcionando velocidade e flexibilidade, todas as tabelas são unidas por relações definidas tornando possível combinar dados de diferente tabelas nas requisições. (Manual de Referência do MySQL 4.1)

### 5.3 PHP

PHP<sup>9</sup>: o PHP é uma linguagem gratuita e de código fonte aberto muito utilizada para o desenvolvimento de aplicações Web. O código desenvolvido nessa linguagem é processado pelo servidor e apenas o resultado é enviado para o cliente. (NIEDERAUER, 2011, p. 23)

---

<sup>7</sup> <http://www.aptana.com/>

<sup>8</sup> <http://downloads.mysql.com/docs/refman-4.1-pt.a4.pdf>

<sup>9</sup> <http://php.net/>

## 5.4 BOOTSTRAP

BOOTSTRAP<sup>10</sup> é um framework em CSS utilizado para desenhar telas HTML, que contém diversos componentes prontos para a criação de telas web de forma mais fácil e objetiva.

Segundo SCHMITZ (2004, p.1):

O Bootstrap nasceu como uma biblioteca ligada ao Twitter (o nome era "Twitter Bootstrap"), no qual a equipe de desenvolvimento usa para o desenvolvimento desta aplicação, e como o resultado ficou muito bom, resolveram disponibilizar de forma gratuita para a comunidade. Com o avanço do produto, tendo ajuda de diversos desenvolvedores, o framework front-end evoluiu para esta terceira geração, na qual chamamos apenas de Bootstrap.

---

<sup>10</sup> <http://getbootstrap.com/>

## **CAPÍTULO 6 – ESPECIFICAÇÃO DOS REQUISITOS:**

### **6.1 LEVANTAMENTO DOS REQUISITOS**

- Cadastro de usuários, firewall, interfaces, configuração, regras, serviços.
- Controlar regras, entrada, saída, redirecionamentos,
- Relatórios de regras.
- Backup das regras.

### **6.2 LISTA DE EVENTOS**

1. Administrador cadastra Usuário
2. Administrador cadastra Firewall
3. Administrador cadastra Interfaces
4. Administrador cadastra Configuração
5. Administrador cadastra Serviços
6. Administrador libera acesso Internet rede interna
8. Administrador cadastra regras
- 9- Administrador gera Relatórios das regras
- 10- Administrador gera Backup das regras

### 6.3 ESTIMATIVAS DE DURAÇÃO DAS ETAPAS DE DESENVOLVIMENTO

<b>Atividades</b>	<b>Data Inicia</b>	<b>Data Fim</b>	<b>Total de Dias</b>
Levantamento de Requisitos	01/12/2013	31/03/2013	31
Especificação dos Requisitos	01/01/2014	10/01/2014	10
Diagrama De Caso de Uso	11/01/2014	20/01/2014	10
Especificação de Caso de Uso	21/01/2014	31/01/2014	10
Diagrama de Atividades	01/02/2014	10/02/2014	10
Diagrama de Sequencia	11/02/2014	21/02/2014	10
Diagrama de Classe	22/02/2014	28/02/2014	10
MER	01/03/2014	17/03/2014	17
Programação	18/03/2014	20/07/2014	125
Testes	21/07/2014	31/07/2014	10
Instalação	01/08/2014	07/08/2014	7
Treinamento	08/08/2014	18/09/2014	10

**Tabela 1 – Estimativa de duração das etapas de desenvolvimento**



## 6.5 RECURSOS NECESSÁRIOS

- **Humano:**
  - Analista/Desenvolvedor
- **Equipamentos:**
  - 01 Microcomputador core i3 2.0 Ghz, 4 Gb de memória Ram e HD de 700 Gb;
  - 01 Impressora (Copiadora, Scanner);
- **Software:**
  - Aptana Studio 3(PHP);
  - Banco de Dados MYSQL;

## 6.6 Estimativas de custos

- **Analista/Desenvolvedor**

Custo diário = R\$ 80,00

Total de dias = 125

Custo total: (Custo diário X Total de dias) = 80 X 138 = 10000

- **Microcomputador:**

Valor unitário = R\$ 1.900,00;

Depreciação 2 anos = R\$ 2.500,00/24 (meses) = R\$ 79,16 por mês;

Custo diário = R\$ 79,16/30 (dias) = R\$ 2,63;

Custo do Projeto (125 dias) = R\$ 2,63 \* 125 = 328,75

- **Impressora:**

Valor Unitário: R\$ 400,00;

Depreciação 2 anos: R\$ 400,00/24 (meses) = R\$ 16,67 por mês;

Custo Diário: R\$ 16,67/30 (dias) = R\$ 0,56;

Custo Projeto (125): R\$ 0,69\* 125 = R\$ 86,25;

- **Aptana Studio 3 (PHP):**

Valor unitário: R\$0,00 (Edição Livre);

- **Banco de dados MYSQL**

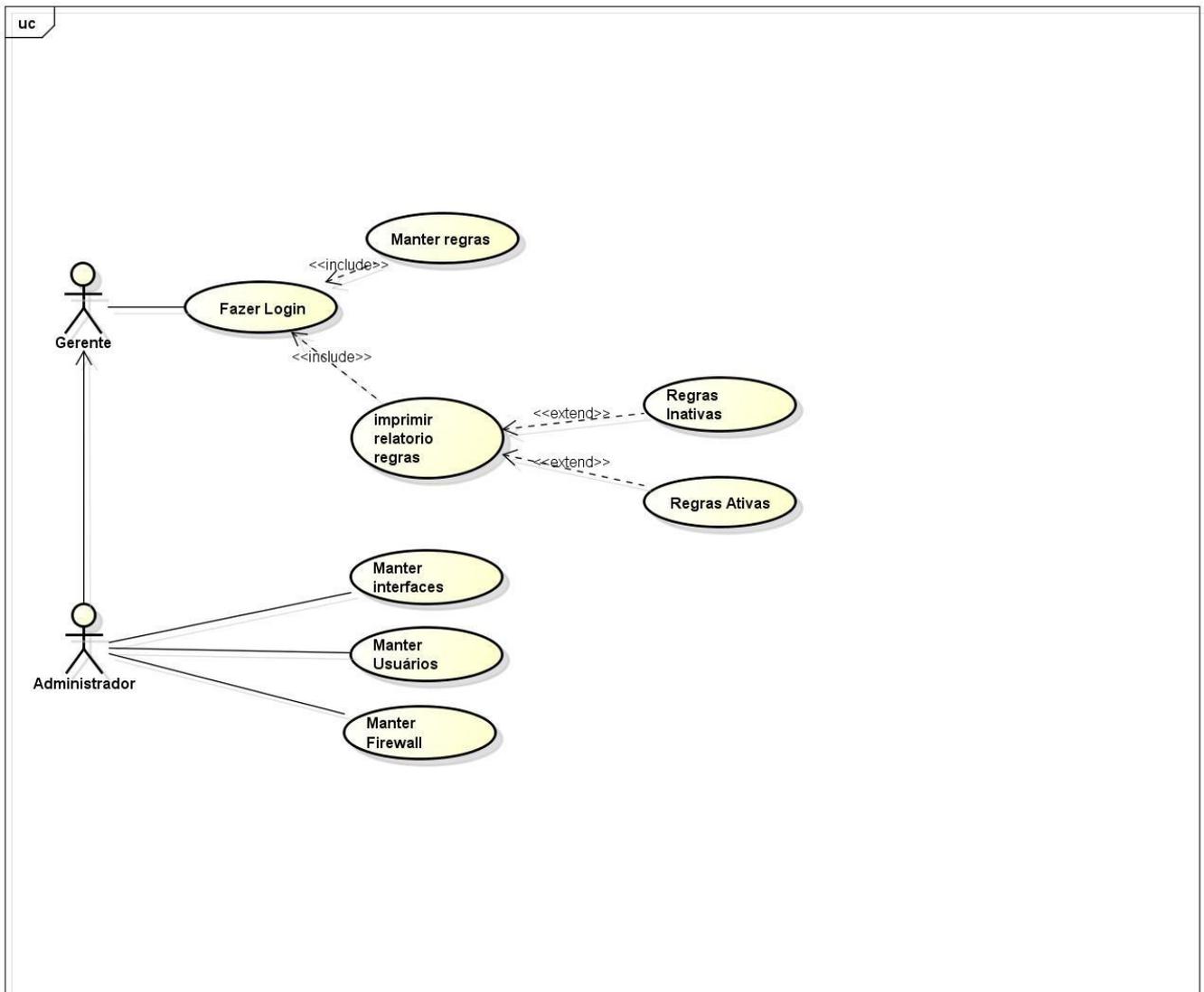
Valor unitário R\$ 0,00 (Edição Livre);

## 6.7 ORÇAMENTO DO PROJETO

Itens	Custo
Analista/Desenvolvedor	R\$ 10.000,00
Microcomputador	R\$328,75
Impressora	R\$86,25
Aptana Studio 3	R\$ 0,00
Banco de Dados MYSQL	R\$ 0,00
<b>Custo Total</b>	<b>R\$ 10.415,00</b>

**Tabela 3 Orçamento do Projeto**

## 6.8 MOVIMENTAÇÕES GERAIS:



powered by Astah

FIGURA 4- UC 01: Movimentação (Visão Geral)

## 6.9 CASOS DE USO: EFETUAR LOGIN

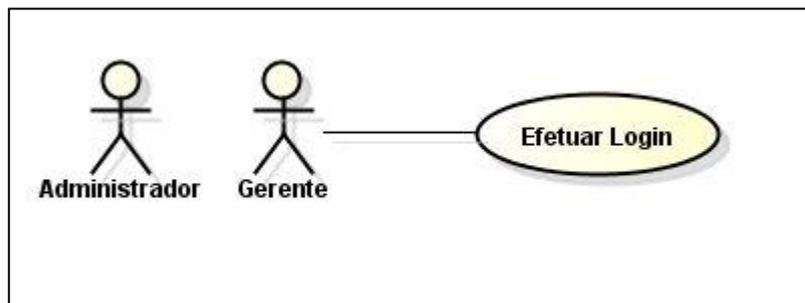


Figura 5 – UC 02: Efetuar Login

<b>NOME DO CASO DE USO</b>	Efetuar Login
<b>ATOR PRINCIPAL</b>	Administrador / Gerente
<b>RESUMO (UC)</b>	Realizar o acesso no sistema.
<b>AÇÕES DO ATOR</b>	<b>AÇÕES DO SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Sistema solicita os dados para efetuar o controle de acesso.</li> <li>2. Usuário informa os dados.</li> <li>3. O sistema valida os dados.</li> </ol>	<ol style="list-style-type: none"> <li>4. O Usuário conecta-se no sistema.</li> </ol>
<b>CENÁRIO ALTERNATIVO</b>	O Usuário pode informar os dados ou não.

Tabela 4 – Efetuar Login

## 6.10 CASOS DE USO: CADASTRAR USUÁRIO

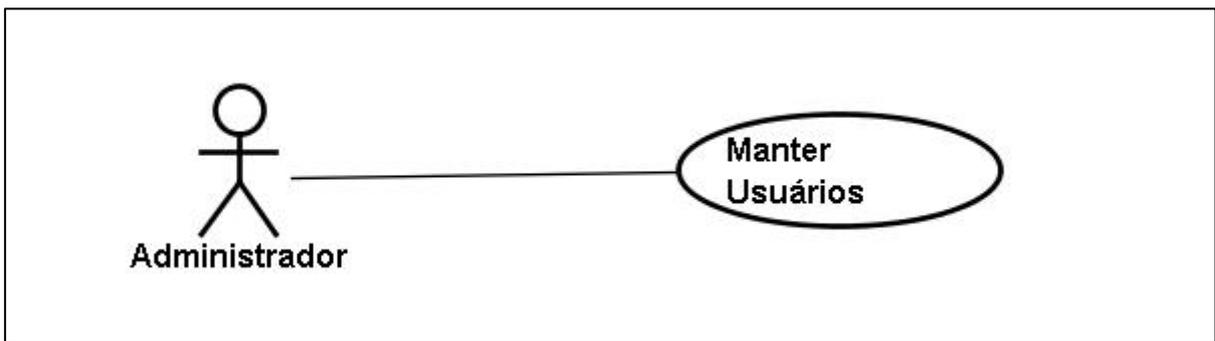


Figura 6 – UC 03: Cadastrar Usuário

<b>NOME DO CASO DE USO</b>	Cadastrar Usuário
<b>ATOR PRINCIPAL</b>	Administrador
<b>RESUMO (UC)</b>	Realiza o cadastro de Usuários.
<b>AÇÕES DO ATOR</b>	<b>AÇÕES DO SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Efetuar Login no sistema.</li> <li>2. Cadastra novo Usuário.</li> <li>3. Poderá Utilizar o Sistema para realizar qualquer operação.</li> </ol>	<ol style="list-style-type: none"> <li>4. Disponibiliza os campos para inserir os dados necessários para o cadastro.</li> <li>5. Efetiva a inclusão dos dados</li> </ol>
<b>CENÁRIO ALTERNATIVO</b>	<ol style="list-style-type: none"> <li>1- Administrador pode excluir um Usuário.</li> <li>2- Administrador pode alterar um Usuário.</li> <li>3- Administrador pode cancelar a operação.</li> </ol>

Tabela 5 – Cadastrar Usuários

## 6.11 CASOS DE USO: CADASTRAR FIREWALL

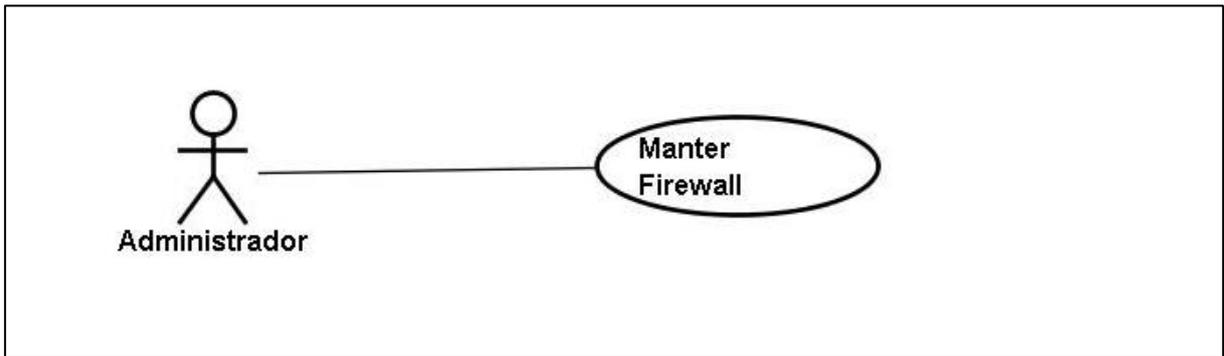


Figura 7 – UC 04: Cadastrar Firewall

<b>NOME DO CASO DE USO</b>	Cadastrar Firewall
<b>ATOR PRINCIPAL</b>	Administrador
<b>RESUMO (UC)</b>	Realizar o cadastro do firewall.
<b>AÇÕES DO ATOR</b>	<b>AÇÕES DO SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Efetuar Login.</li> <li>2. Cadastrar novo firewall.</li> <li>3. Poderá Utilizar o Sistema para realizar qualquer operação</li> </ol>	<ol style="list-style-type: none"> <li>4. Disponibiliza os campos para inserir os dados necessários para o cadastro.</li> <li>5. Confirma a inclusão do firewall.</li> </ol>
<b>CENÁRIO ALTERNATIVO</b>	<ol style="list-style-type: none"> <li>1- Administrador pode excluir um Firewall.</li> <li>2- Administrador pode alterar um Firewall.</li> <li>3- Administrador pode cancelar a operação.</li> </ol>

Tabela 6 – Cadastrar Firewall

## 6.12 CASOS DE USO: CADASTRAR INTERFACES

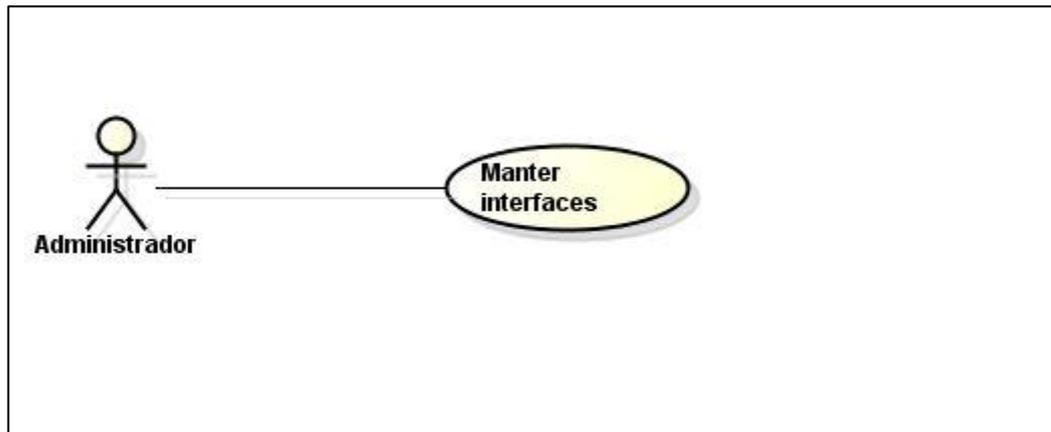


Figura 8 – UC 05: Cadastrar Interfaces

<b>NOME DO CASO DE USO</b>	Cadastrar Interfaces
<b>ATOR PRINCIPAL</b>	Administrador
<b>RESUMO (UC)</b>	Realiza o cadastro das interfaces.
<b>AÇÕES DO ATOR</b>	<b>AÇÕES DO SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Efetuar Login.</li> <li>2. Cadastra a interfaces</li> </ol>	<ol style="list-style-type: none"> <li>3. Solicita os dados necessários para o cadastro.</li> <li>4. Efetiva a inclusão dos dados.</li> </ol>
<b>CENÁRIO ALTERNATIVO</b>	<ol style="list-style-type: none"> <li>1- Administrador pode excluir uma interface.</li> <li>2- Administrador pode alterar uma interface.</li> <li>3- Administrador pode cancelar a operação.</li> </ol>

Tabela 7 – Cadastrar Interfaces

## 6.13 CASOS DE USO: MANTER REGRAS

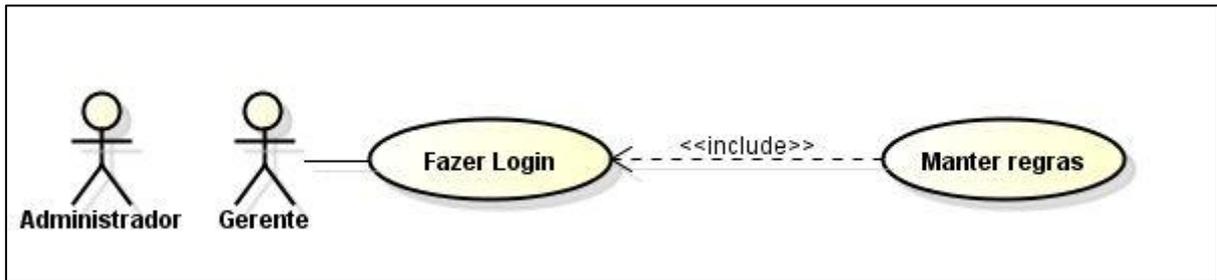


Figura 9 – UC 06: Manter Regras

<b>NOME DO CASO DE USO</b>	Manter Regras
<b>ATOR PRINCIPAL</b>	Administrador/Gerente
<b>RESUMO (UC)</b>	Manter as regras.
<b>AÇÕES DO ATOR</b>	<b>AÇÕES DO SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Efetuar Login.</li> <li>2. Usuário preenche os dados no formulário referente e regra. <ol style="list-style-type: none"> <li>2.1 Clica no botão salvar regra.</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>3. Sistema envia mensagem de regra criada.</li> <li>4. Efetiva a inclusão dos dados.</li> </ol>
<b>CENÁRIO ALTERNATIVO</b>	O Usuário poderá cancelar a operação a qualquer momento e deverá estar identificado por um login para efetuar a transação.

Tabela 8 – Manter Regras

## 6.14 CASOS DE USO: IMPRIMIR RELÁTORIOS REGRAS

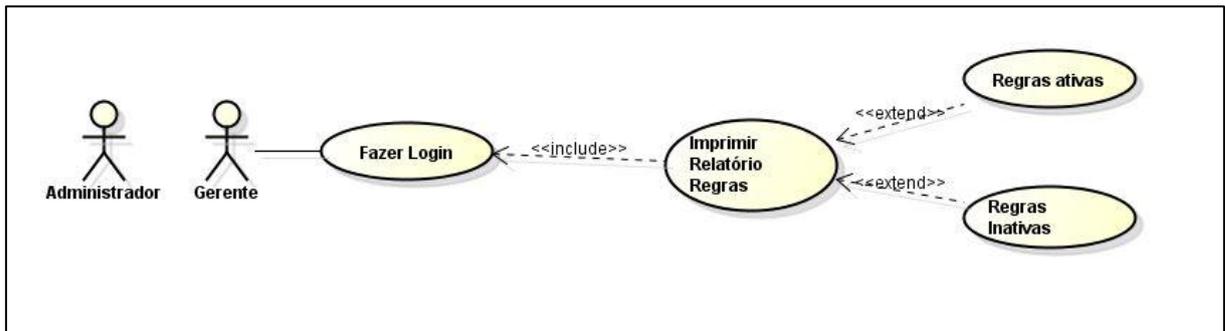


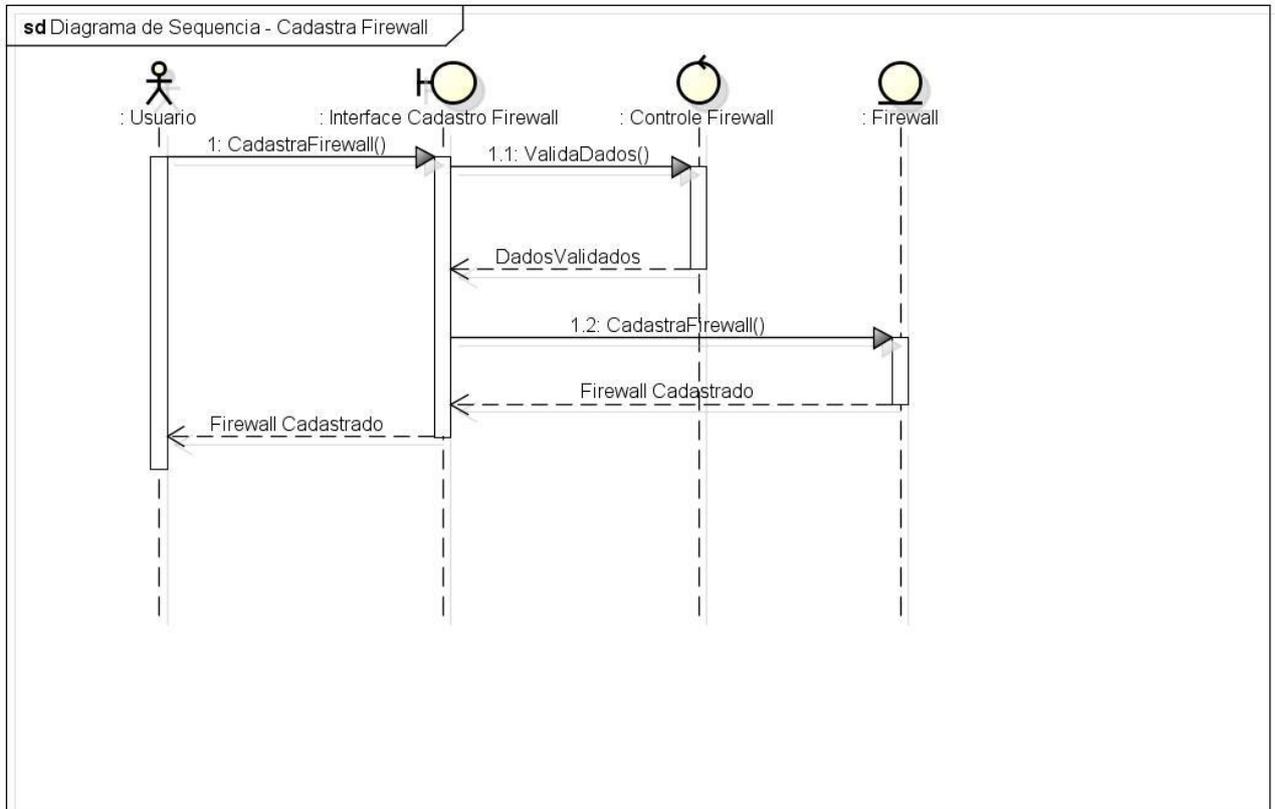
Figura 10 – UC 07: Imprimir relatório Regras

<b>NOME DO CASO DE USO</b>	Imprimir Regras
<b>ATOR PRINCIPAL</b>	Administrador /Gerente
<b>RESUMO (UC)</b>	Imprimir relatório regras do sistema.
<b>AÇÕES DO ATOR</b>	<b>AÇÕES DO SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Efetuar Login no sistema.</li> <li>2. Selecionar o firewall de onde quer imprimir o relatório de regras             <ol style="list-style-type: none"> <li>2.1 Selecionar regras ativas ou inativas.</li> <li>2.2 Clica no botão gerar</li> <li>2.3 Clicar em imprimir.</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>3. Sistema exibe as regras selecionadas.</li> <li>4. Faz a impressão das regras.</li> </ol>
<b>CENÁRIO ALTERNATIVO</b>	<ol style="list-style-type: none"> <li>1- O usuário pode imprimir ou não.</li> <li>2- Usuário pode cancelar a operação.</li> </ol>

Tabela 9 – Imprimir Relatório de Regras

## CAPÍTULO 7 – DIAGRAMAS DO SISTEMA

### 7.1 DIAGRAMAS DE SEQUÊNCIA: CADASTRAR FIREWALL



powered by Astah

Figura 11 – Diagramas de Sequência: Cadastrar Firewall

## 7.2 MODELO DE ENTIDADE RELACIONAMENTO

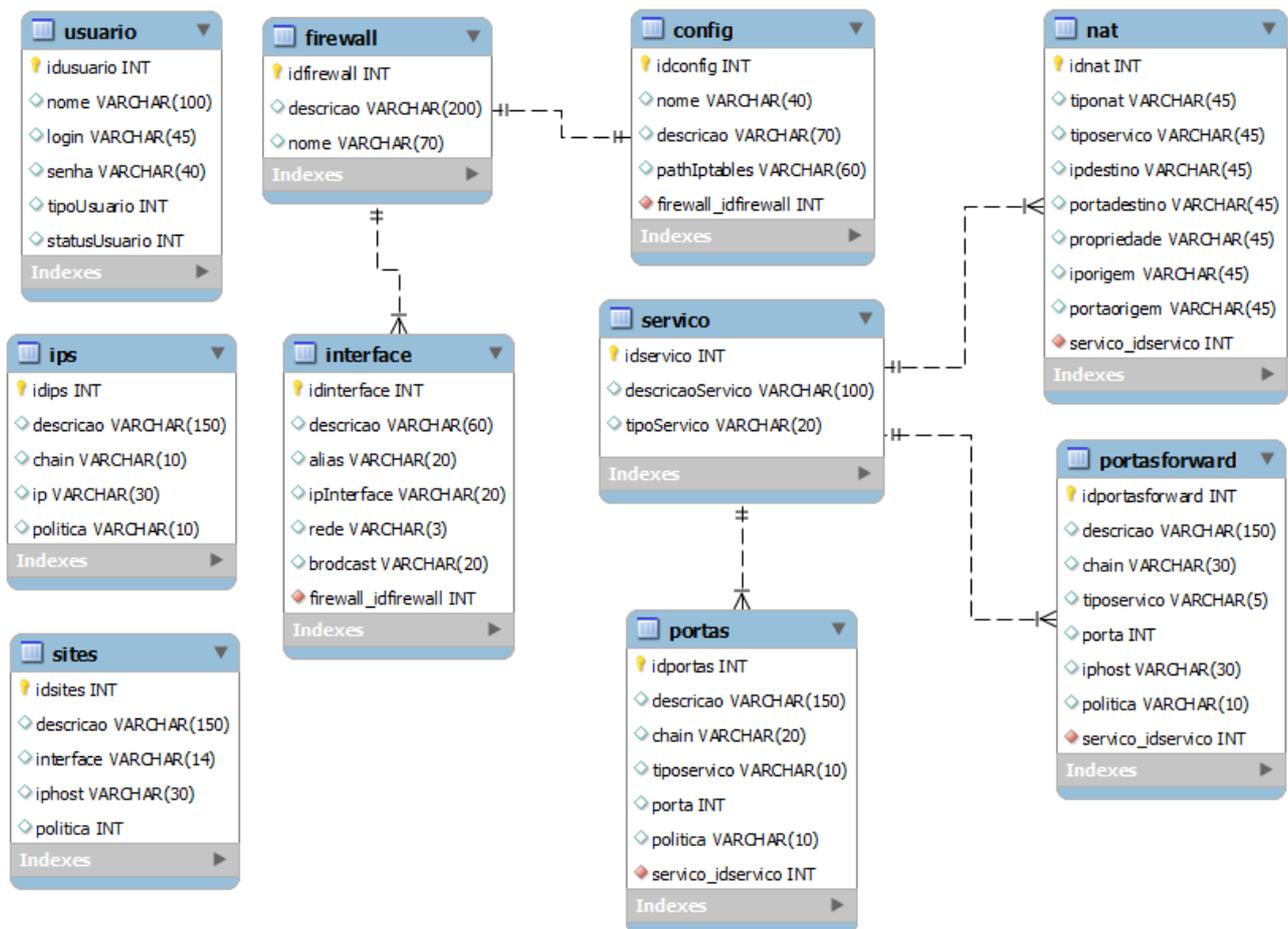


Figura 12 – Modelo Entidade e Relacionamento

### 7.3 DIAGRAMA DE CLASSES

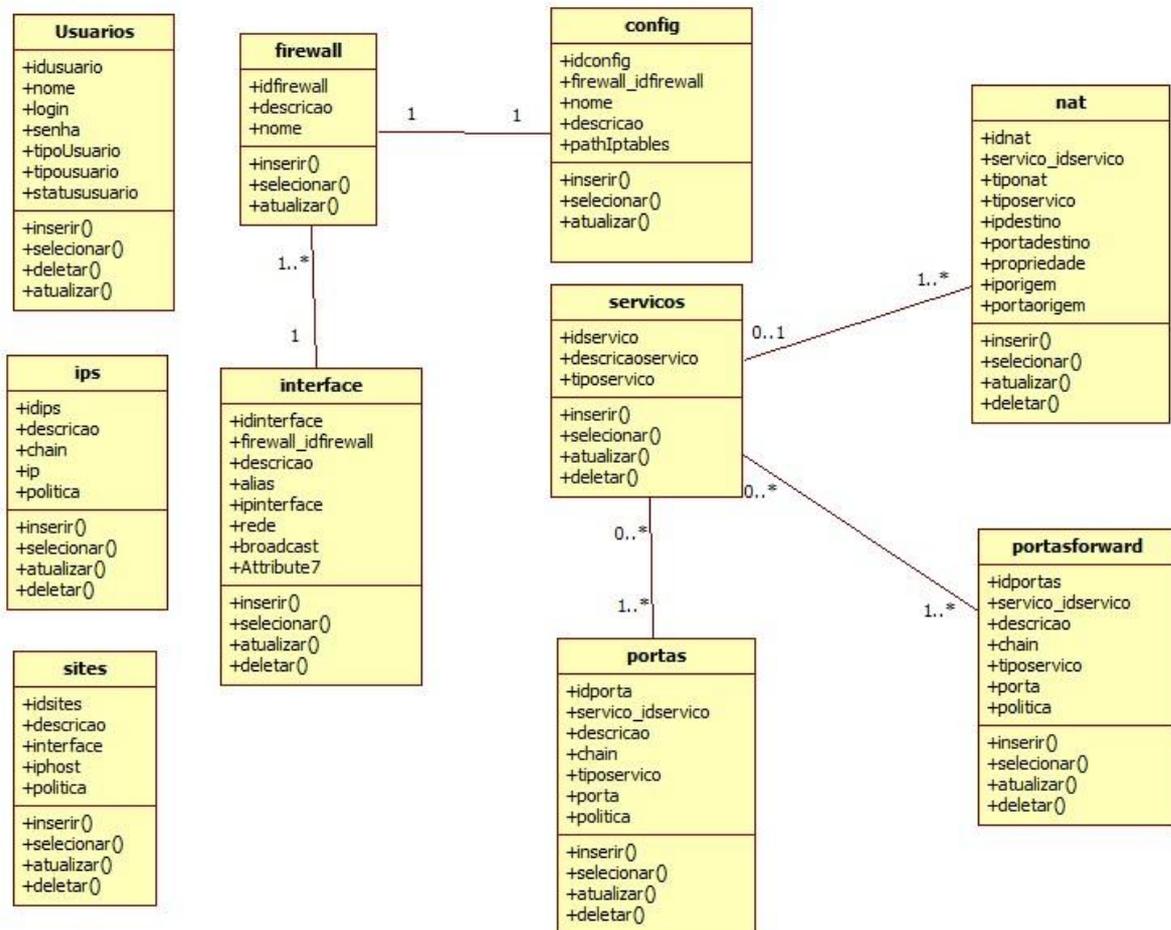
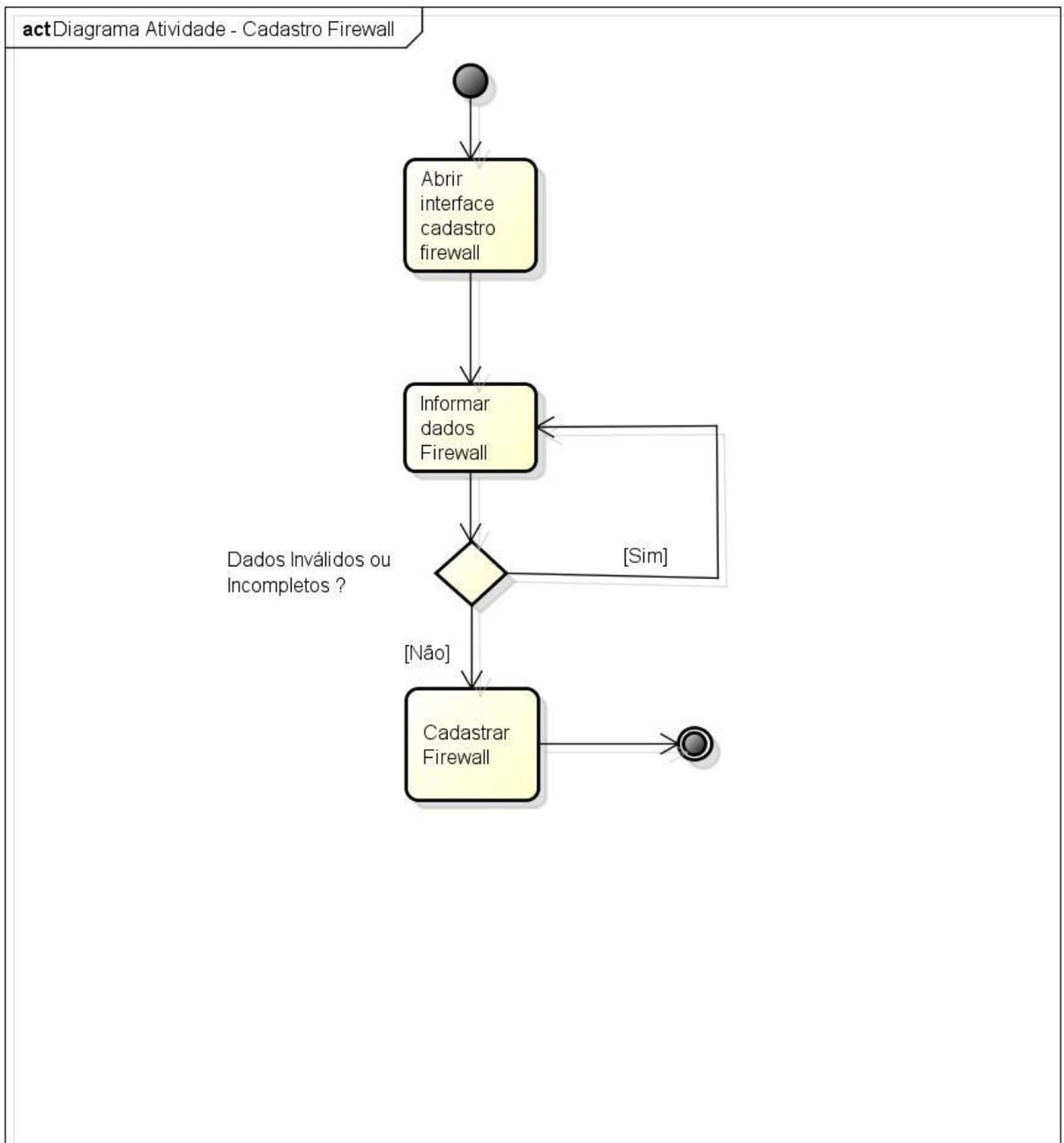


Figura 13 – Diagrama de Classes

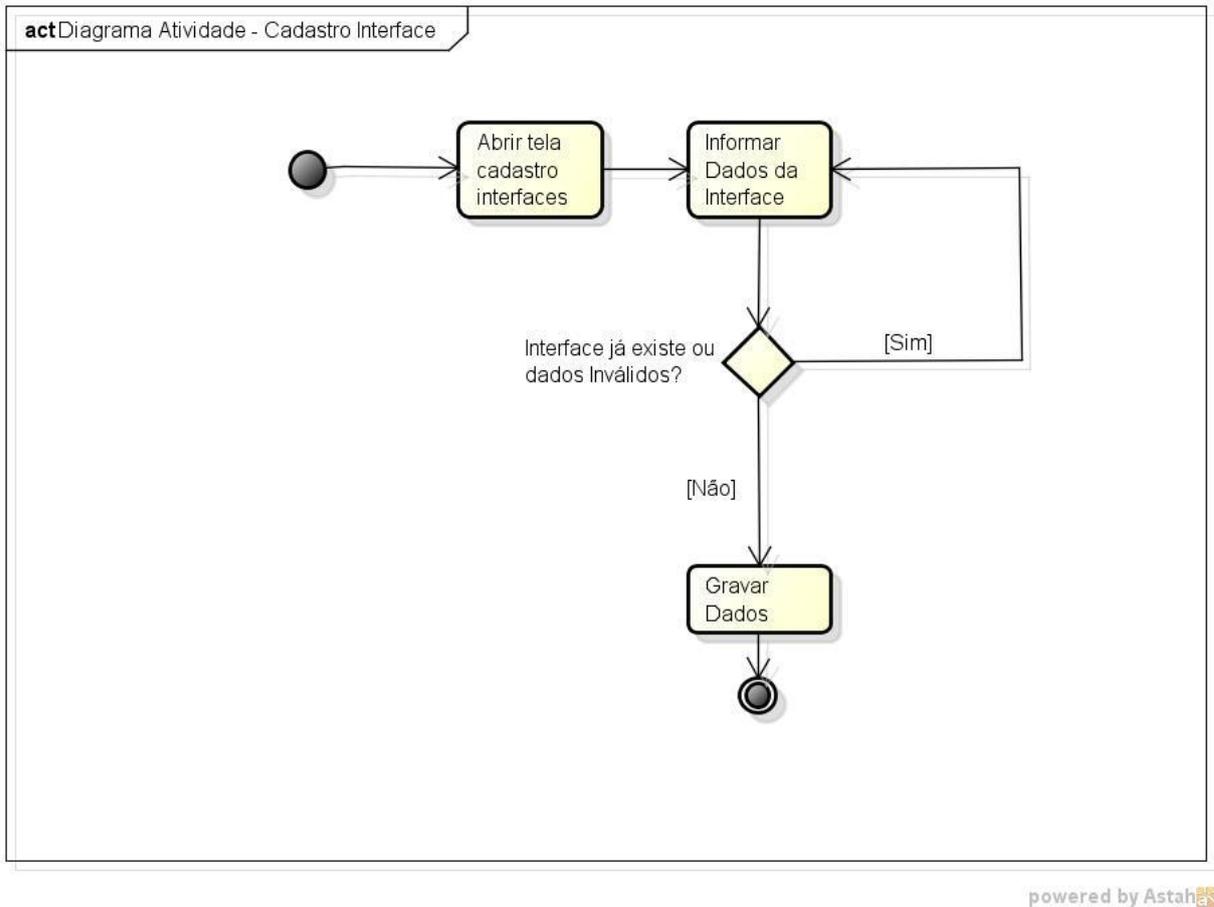
## 7.4 DIAGRAMA DE ATIVIDADES – CADASTRO DE FIREWALL.



powered by Astah

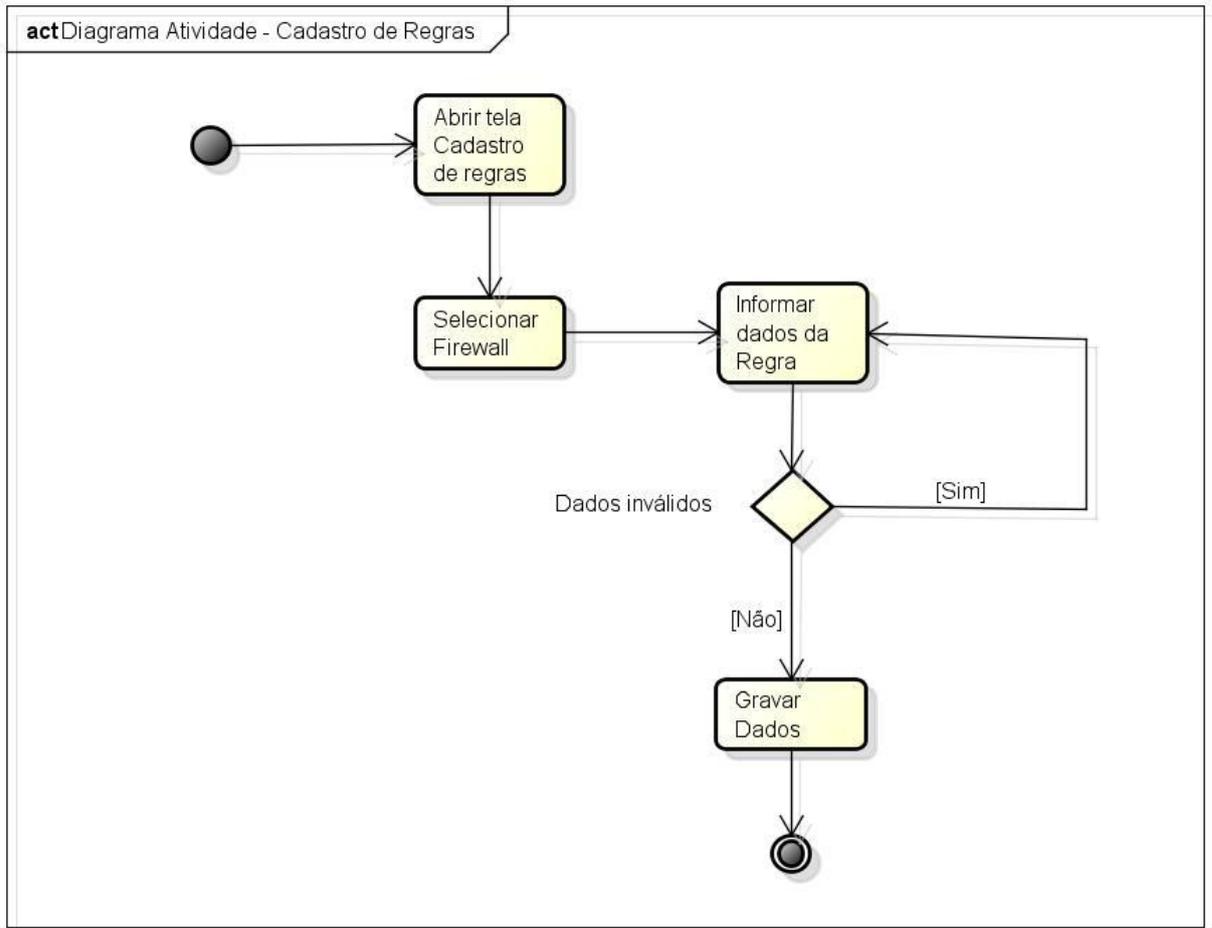
Figura 14 - Diagrama de Atividades: Cadastro de firewall.

## 7.5 DIAGRAMA DE ATIVIDADES – CADASTRO DE INTERFACE.



**Figura 15 – Diagrama de Atividades: Cadastro Interface.**

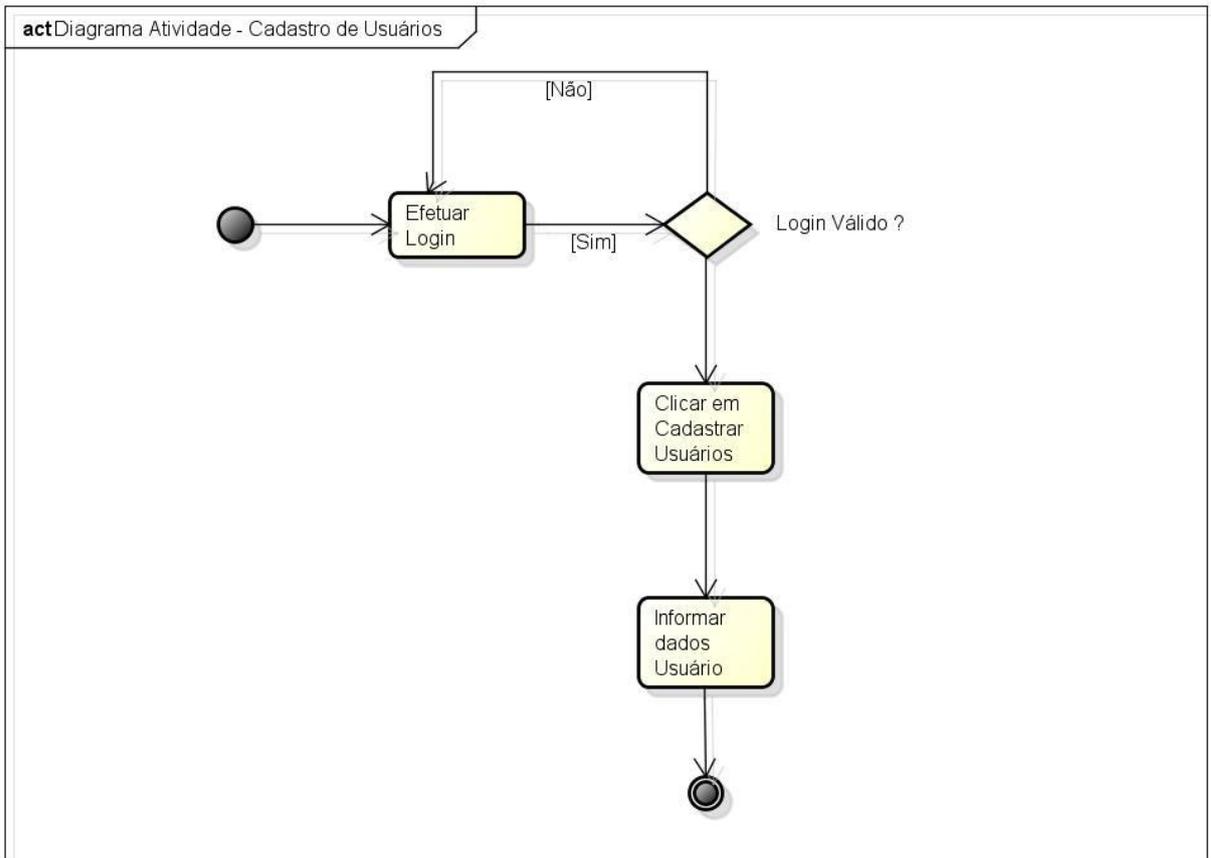
## 7.6 DIAGRAMA DE ATIVIDADES – CADASTRO DE REGRAS



powered by Astah

Figura 16 – Diagrama de Atividades: Cadastro de Regras

## 7.7 DIAGRAMA DE ATIVIDADES - CADASTRAR



powered by Astah

**Figura 17 – Diagrama de Atividades: Cadastrar Usuários.**

## CONCLUSÃO

Administrar redes e sistemas de computadores sempre foi um grande desafio para profissionais da computação.

Deste modo, na tentativa de auxiliar os administradores e iniciantes existem inúmeras ferramentas de administração e 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.

A partir do estudo realizado entre as diversas ferramentas, apresentou-se uma proposta de ferramenta que contemple as principais características positivas das ferramentas estudadas, aliada a características não encontradas em nenhuma delas nas quais são julgados essenciais para o cenário atual.

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

Considera-se que muito há de ser feito para o desenvolvimento completo da ferramenta de administração computacional idealizada nesse trabalho.

## REFERÊNCIAS

### Bibliográficas

- BURNETTE, Ed Burnette. **Eclipse IDE – Guia de Bolso**. 1ª Edição. Bookman. 2006.
- EQUIPE CONECTIVA. Conectiva S.A. **Segurança de redes: Firewall**. Curitiba. 2001.
- GRADY BOOCH, J. R.& IVAR J. UML – **Guia do Usuário**, Editora: Campus, 2000.
- NETO,UBARATAN, **Dominando Linux Firewall Iptables**, Rio de Janeiro : Editora Ciência Moderna LTDA, 2004.
- NIEDERAUER. Juliano Niederauer. **Desenvolvendo Websites com PHP. 2ª Edição**. Novatec Editora Ltda .São Paulo.2011.
- SILVA, Maurício Samy, CSS3 :**Desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3**. Novatec Editora.São Paulo. 2012.

### Digitais

- APTANA STUDIO, Appceletator. Disponível em <http://aptana.com/support>. Acesso em: 15/03/2014 às 11:20 min.
- KMyfirewall, Christian Hubinger, **The KMyFirewall Handbook**. Revision 1.0 (03/04/2005). Disponível em [http://www.kmyfirewall.org/kmf\\_doc/index.html](http://www.kmyfirewall.org/kmf_doc/index.html). Acesso em : 18/06/2014 ás 17:40 min.
- MYSQL, MySQL AB. **Manual de Referência do MySQL 4.1**.Disponívelmem <http://downloads.mysql.com/docs/refman-4.1-pt.a4.pdf>. Acesso em: 16/03/2014 às 9:00 min.
- PHP. **Manual do PHP**.Disponível em [http://www.php.net/manual/pt\\_BR/index.php](http://www.php.net/manual/pt_BR/index.php). Acesso em: 14/03/2014 às 9:46 min.
- WebMin, Docs for Webmin.Disponível em <http://doxfer.webmin.com/Webmin>. Acesso em: 13/03/2014 às 8:33 min.

SCHMITZ, Daniel Schmitz. **Bootstrap 3 framework front-end para desenvolvimento web e móbile**. Disponível em: <https://leanpub.com/livro-bootstrap>. Acesso em: 10/05/2014 às 12:33 min.