

# *Revista da Graduação*

---

Vol. 4

No. 1

2011

32

---

**Seção: FACULDADE DE ADMINISTRAÇÃO, CONTABILIDADE E  
INFORMÁTICA – CAMPUS URUGUAIANA**

Título: SQUIDHELP: interface web para  
administração do servidor proxy Squid

Autor: Ícaro Lins Iglesias

Este trabalho está publicado na Revista da Graduação.

ISSN 1983-1374

<http://revistaseletronicas.pucrs.br/ojs/index.php/graduacao/article/view/8836/6189>

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE ADMINISTRAÇÃO, CONTABILIDADE E INFORMÁTICA  
CURSO DE SISTEMAS DE INFORMAÇÃO

ÍCARO LINS IGLESIAS

**SQUIDHELP: INTERFACE WEB PARA ADMINISTRAÇÃO DO SERVIDOR PROXY  
SQUID**

Uruguiana

2010

ÍCARO LINS IGLESIAS

**SQUIDHELP: INTERFACE WEB PARA ADMINISTRAÇÃO DO SERVIDOR PROXY  
SQUID**

Trabalho de conclusão de curso de graduação apresentado à Faculdade de Administração, Ciências Contábeis e Informática da Pontifícia Universidade Católica do Rio Grande do Sul, campus Uruguaiana, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

**Orientador(a): Denise Michels Doviggi**

Uruguaiana

2010

ÍCARO LINS IGLESIAS

**SQUIDHELP: INTERFACE WEB PARA ADMINISTRAÇÃO DO SERVIDOR PROXY  
SQUID**

Trabalho de conclusão de curso de graduação apresentado à Faculdade de Administração, Ciências Contábeis e Informática da Pontifícia Universidade Católica do Rio Grande do Sul, campus Uruguaiana, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovada em \_\_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_.

**BANCA EXAMINADORA:**

Prof<sup>o</sup>. Msc. Cezar Luiz Krause – PUCRS

---

Prof<sup>o</sup>. Marcelo Amador Lucas - PUCRS

---

Prof<sup>a</sup>. Denise Michels Doviggi – PUCRS

---

Dedico este trabalho a meus pais  
Paulo Roberto da F. Iglesias e Conceição  
Lins Espelocin, por sempre acreditarem no  
meu potencial, dando o apoio necessário  
para que essa realização fosse possível.

## AGRADECIMENTOS

À Deus pelo dom da vida concebido e oportunidade de conclusão desta etapa.

À minha mãe, Conceição Lins Espelocin, pela educação que me proporcionou, sempre sonhando em ver seu filho formado, dando todo o apoio necessário para que este sonho se tornasse realidade.

À minha amada, Narjara Figueira Barbosa, que me fez ingressar na Universidade, sempre estando ao meu lado, compartilhando conhecimentos, alegrias e sofrimentos, entendendo todo meu *stress* e mau humor a cada final de semestre.

Ao meu pai, Paulo Roberto da F. Iglesias, que direta ou indiretamente contribuiu para minha graduação.

Ao meu padrasto, Haroldo Gomes Espelocin (*in memoriam*) que, de onde estiver, tenho a certeza de que sempre estará torcendo pelo meu sucesso.

À minha madrinha, Miete Lima Lins, por sempre me incentivar a estudar, tendo grande importância na realização dessa etapa.

A todos os colegas (a galera da “coca-cola” do intervalo), em especial aos colegas de ônibus Claiton Correa, Michel Borges e Guilherme Benites, pelos bons momentos de aprendizado vividos nestes quatro anos de diárias viagens.

Aos “monitores” Claiton Marques Correa e Fábio Quintana Júnior que sempre me ajudaram pacientemente em todas as etapas da graduação.

À todos os professores que além do conhecimento didático transmitido, nos compartilharam suas experiências de vida.

Aos funcionários do campus, em especial a funcionária Clara Valim do laboratório de informática, por nos aturar e dar o apoio necessário durante toda graduação.

À direção e administração da empresa Farmácias Fronteira que me oportunizou a conciliação entre trabalhar e estudar.

Aos amigos que sempre apoiaram e acreditaram no meu potencial, em especial aos casais Paulo Sanhotene e Lilian Borscheid, Fabiano Montesano e Maiara Barbosa.

Aos demais familiares que de alguma forma colaboraram para que este trabalho fosse realizado.

Se você tem uma maçã e eu tenho  
uma maçã, e nós trocamos as maçãs,  
então você e eu ainda teremos uma  
maçã.

Mas se você tem uma idéia e eu  
tenho uma idéia, e nós trocamos essas  
idéias, então cada um de nós terá duas  
idéias.

**George Bernard Shaw**

## RESUMO

Este trabalho visa o desenvolvimento de uma aplicação *web* tendo como finalidade facilitar a administração do *proxy* para GNU/Linux, o servidor Squid. O projeto foi idealizado a partir da necessidade de uma interface amigável ao usuário, sem a necessidade de conhecimentos avançados em sistemas GNU/Linux, tendo em vista que toda a administração de servidores desta plataforma dá-se via console, exigindo certa familiaridade na manipulação de comandos em modo texto de seus usuários.

O sistema desenvolvido foi dividido em quatro módulos: contas de usuários, administração de ACL's, *scripts* no *proxy* e relatórios. O módulo Contas de usuários contém as telas de informações cadastrais dos usuários e manutenção destas contas. No módulo Administração de ACL's são realizadas as liberações ou bloqueios dos acessos de internet (sites, domínios, endereços IP, endereços MAC, palavras proibidas e extensões de arquivos) conforme a política adotada por cada empresa. O módulo *scripts* no *proxy* é utilizado para monitoração de *logs* de acesso e *cache*, reconfigurar, parar ou reiniciar o servidor *proxy* Squid. Por fim, foi desenvolvido o módulo de Relatórios que possibilita a emissão de relatórios de acesso interoperabilizando com o sistema SARG, além de relatórios sobre os cadastros, liberações e bloqueios efetuados no banco de dados da própria aplicação, gerados no formato pdf.

Para o desenvolvimento do presente trabalho foram utilizadas as linguagens de programação PHP, HTML e JavaScript; CSS para estilização das páginas e MySQL como SGBD.

Palavras-chave: GNU/Linux. Administrador *web*. Servidor *proxy*. Squid.



## **ABSTRACT**

This work aims to develop a web application intended to facilitate the administration of the proxy for GNU / Linux, the Squid server. The project was conceived from the need to have a user-friendly interface, without requiring advanced knowledge in GNU / Linux systems, given that all the server administration gives to this platform via the console, requiring some familiarity manipulation commands in text mode of its users.

The system has been divided into four modules: user accounts, management of ACL's, scripts and reports on the proxy. The module contains the User Accounts screen users of cadastral information and maintaining these accounts. In the Administration module ACLs are made releases or locks the access to the Internet (sites, domains, IP addresses, MAC addresses, words and banned file extensions) as the policy adopted by each company. The scripts on the proxy module is used for monitoring access logs and cache, reset, stop or restart the Squid proxy server. Finally, the module was developed that allows the reporting reporting access to the system interoperating SARG, plus reports on entries, releases and locks are taken on the database application itself, generated in pdf format.

For the development of this work were used programming languages PHP, HTML, JavaScript, CSS for styling pages and MySQL as DBMS.

**Keywords:** GNU/Linux. Web Administrator. Proxy Server. Squid

## LISTA DE ILUSTRAÇÕES

<b>Figura 1</b> – Total de incidentes reportados ao CERT.br por ano.....	13
<b>Figura 2</b> - Funcionamento servidor <i>proxy</i> numa rede.....	21
<b>Figura 3</b> - Relatório de acessos ao servidor <i>proxy</i> Squid.....	24
<b>Figura 4</b> - Exemplo de script em PHP.....	25
<b>Figura 5</b> - Exemplo de script em Javascript.....	26
<b>Figura 6</b> - Exemplo de arquivo CSS.....	27
<b>Figura 7</b> - Modelo de aplicação com Ajax.....	28
<b>Figura 8</b> - Passo a passo servidor Apache.....	29
<b>Figura 9</b> - Exemplo de configuração para <i>proxy</i> em modo autenticado.....	30
<b>Figura 10</b> - Tela de <i>login</i> .....	32
<b>Figura 11</b> - Campos obrigatórios tela de <i>login</i> .....	33
<b>Figura 12</b> - <i>Login</i> ou senha inválidos.....	33
<b>Figura 13</b> - Usuário sem permissão ao SquidHelp.....	33
<b>Figura 14</b> - Painel de controle.....	34
<b>Figura 15</b> - Menu Rápido.....	35
<b>Figura 16</b> - Configuração de usuários.....	36
<b>Figura 17</b> - Novo usuário.....	36
<b>Figura 18</b> - Campos obrigatórios tela novo usuário.....	38
<b>Figura 19</b> – Alerta de <i>login</i> já cadastrado.....	38
<b>Figura 20</b> - Tela cadastro de usuário efetuado com sucesso.....	39
<b>Figura 21</b> - Tela manutenção de contas de usuário.....	39
<b>Figura 22</b> - Tratamento de erros da tela manutenção de usuários.....	41
<b>Figura 23</b> - Alerta de dados alterados com sucesso.....	42
<b>Figura 24</b> - Módulo Administrar ACL's.....	42
<b>Figura 25</b> - Tela domínios/sites.....	44
<b>Figura 26</b> - Campo obrigatório.....	45
<b>Figura 27</b> - Alerta sobre endereço já bloqueado.....	45
<b>Figura 28</b> - Alerta sobre endereço já liberado.....	45
<b>Figura 29</b> - Alerta sobre endereço bloqueado com sucesso.....	46
<b>Figura 30</b> - Alerta sobre endereço liberado com sucesso.....	46
<b>Figura 31</b> - Módulo de scripts no <i>proxy</i> .....	47

<b>Figura 32</b> - Tela monitoramento dos arquivos <code>access.log</code> e <code>cache.log</code> .....	47
<b>Figura 33</b> - Mensagem de <i>proxy</i> reconfigurado com sucesso.....	48
<b>Figura 34</b> - Mensagem de <i>proxy</i> interrompido com sucesso.....	48
<b>Figura 35</b> - Mensagem de <i>proxy</i> reiniciado com sucesso.....	49
<b>Figura 36</b> - Módulo de relatórios.....	49
<b>Figura 37</b> - Relatórios de acessos – SARG.....	50
<b>Figura 38</b> - Relatório com os acessos do dia ao <i>proxy</i> .....	50
<b>Figura 39</b> - Visualização de relatórios gerados no diretório <i>Daily</i> .....	51
<b>Figura 40</b> - Relatório com os acessos de uma semana ao <i>proxy</i> .....	51
<b>Figura 41</b> - Visualização de relatórios gerados no diretório <i>Weekly</i> .....	51
<b>Figura 42</b> - Relatório com os acessos de um mês ao <i>proxy</i> .....	52
<b>Figura 43</b> - Visualização de relatórios gerados no diretório <i>Monthly</i> .....	52
<b>Figura 44</b> - Tratamento de erros formulário de relatório customizado.....	52
<b>Figura 45</b> - Relatório com os acessos num período informado pelo usuário.....	53
<b>Figura 46</b> - Relatórios de políticas SquidHelp.....	53
<b>Figura 47</b> - Relação de sites/domínios URL.....	54
<b>Figura 48</b> - Relatório em pdf dos sites/domínios URL contido no banco de dados...55	55
<b>Figura 49</b> - Relação de endereços IP.....	55
<b>Figura 50</b> - Relatório em pdf dos endereços IP contido no banco de dados.....	55
<b>Figura 51</b> - Relação de endereços MAC.....	55
<b>Figura 52</b> - Relatório em pdf dos endereços MAC contido no banco de dados.....	56
<b>Figura 53</b> - Relação de palavras proibidas.....	57
<b>Figura 54</b> - Relatório em pdf das palavras proibidas contido no banco de dados.....	57
<b>Figura 55</b> - Relação de extensões de arquivos.....	58
<b>Figura 56</b> - Relatório em pdf das extensões de arquivos contido no bd.....	58
<b>Figura 57</b> - Relação de usuários.....	59
<b>Figura 58</b> - Relatório em pdf dos usuários contido no banco de dados.....	59
<b>Figura 59</b> - Alerta Sobre.....	60
<b>Figura 60</b> - Alerta Sair.....	60

## LISTA DE SIGLAS

ACL – *Access Control List*

GNU – *Gnu's Not Unix*

GPL – *General Public License*

HTML – *HyperText Markup Language*

HTTP – *HyperText Transfer Protocol*

IHC – *Interface Homem Computador*

IP – *Internet Protocol*

MAC – *Media Access Control*

MD5 – *Message Digest 5*

NAT – *Network Address Translation*

PHP – *Hypertext PreProcessor*

SARG – *Squid Analysis Report Generator*

SGBD – *Sistema Gerenciador de Banco de Dados*

SQL - *Structured Query Language*

TCP/IP – *Transfer Control Protocol/Internet Protocol*

TI – *Tecnologia da Informação*

URL – *Uniform Resource Locator*

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>13</b>
1.1 ESTRUTURA DO TRABALHO.....	14
<b>2 OBJETIVOS.....</b>	<b>16</b>
2.1 OBJETIVO GERAL.....	16
2.2 OBJETIVOS ESPECÍFICOS.....	16
<b>3 MOTIVAÇÃO.....</b>	<b>17</b>
<b>4 JUSTIFICATIVA.....</b>	<b>18</b>
<b>5 REFERENCIAL TEÓRICO.....</b>	<b>19</b>
5.1 GNU/LINUX.....	19
5.2 SQUID.....	20
<b>5.2.1 Acl.....</b>	<b>21</b>
<b>5.2.2 Modo de configuração.....</b>	<b>22</b>
5.2.2.1 <i>Proxy transparente.....</i>	22
5.2.2.2 <i>Proxy autenticado.....</i>	22
5.3 SARG.....	23
5.4 PHP.....	24
5.5 HTML.....	25
5.6 JAVASCRIPT.....	25
5.7 CSS.....	26
5.8 AJAX.....	27
5.9 APACHE.....	28
5.10 MYSQL.....	29
<b>6 DESCRIÇÃO DO SISTEMA.....</b>	<b>30</b>
6.1 FUNCIONAMENTO.....	30
6.2 LOGIN DO SISTEMA.....	32
<b>6.2.1 Descrição da interface.....</b>	<b>32</b>
<b>6.2.2 Tratamento de erros.....</b>	<b>33</b>
6.2.2.1 <i>Descrição dos erros.....</i>	33
6.3 PAINEL DE CONTROLE.....	34

<b>6.3.1 Descrição da interface</b> .....	<b>34</b>
6.4 MENU LATERAL.....	35
<b>6.4.1 Descrição da interface</b> .....	<b>35</b>
6.5 MÓDULO DE USUÁRIOS.....	36
<b>6.5.1 Novo usuário</b> .....	<b>36</b>
6.5.1.1 <i>Descrição da interface</i> .....	37
6.5.1.2 <i>Tratamento de erros</i> .....	37
6.5.1.3 <i>Cadastro efetuado com sucesso</i> .....	38
<b>6.5.2 Manutenção de conta</b> .....	<b>39</b>
6.5.2.1 <i>Descrição da interface</i> .....	40
6.5.2.2 <i>Tratamento de erros</i> .....	41
6.5.2.3 <i>Cadastro efetuado com sucesso</i> .....	42
6.6 MÓDULO ADMINISTRAÇÃO DE ACL's.....	42
<b>6.6.1 Descrição da interface</b> .....	<b>43</b>
<b>6.6.2 Sites/domínios URL</b> .....	<b>44</b>
6.6.2.1 <i>Descrição da interface</i> .....	44
6.6.2.2 <i>Tratamento de erros</i> .....	45
6.6.2.3 <i>Cadastro efetuado com sucesso</i> .....	46
6.7 MÓDULO SCRIPTS NO PROXY.....	46
<b>6.7.1 Descrição da interface</b> .....	<b>47</b>
6.8 MÓDULO DE RELATÓRIOS.....	49
<b>6.8.1 Relatórios de acessos</b> .....	<b>50</b>
<b>6.8.2 Relatórios de políticas SquidHelp</b> .....	<b>53</b>
6.8.2.1 <i>Descrição da interface</i> .....	54
6.9 SOBRE.....	60
6.10 SAIR.....	60
<b>7 Conclusão</b> .....	<b>61</b>
7.1 TRABALHOS RELACIONADOS.....	61
7.2 TRABALHOS FUTUROS.....	62
<b>REFERÊNCIAS</b> .....	<b>63</b>
<b>GLOSSÁRIO</b> .....	<b>64</b>

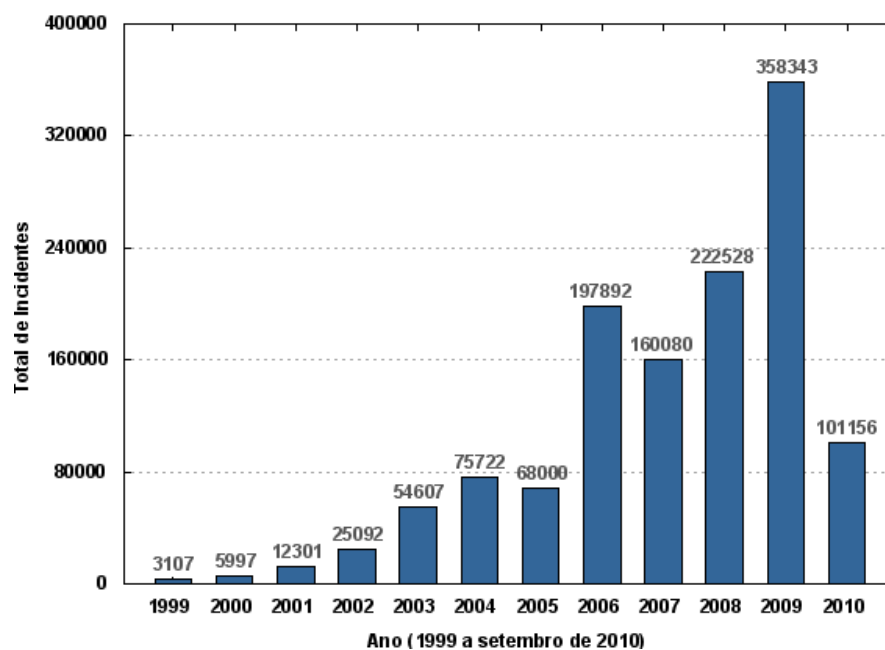
## 1 INTRODUÇÃO

A acelerada disseminação da internet torna o acesso à rede mundial de computadores cada vez mais explorado, servindo tanto como ferramenta de trabalho ou simples forma de lazer e entretenimento.

O conteúdo acessado pelos colaboradores de uma empresa é algo que precisa ser controlado, seja por questões de segurança ou até mesmo de produtividade, cabendo aos administradores implantarem soluções que visem garantir a segurança da rede local, a partir de ações como: bloquear o acesso a sites indesejáveis, armazenar páginas mais acessadas em *cache* e controlar *downloads* dos usuários, buscando otimizar os recursos de internet disponíveis, além de emitir relatórios sobre os acessos de cada usuário da rede para análise de sites que podem ser permitidos ou negados, conforme a política adotada pela empresa.

O CERT.br é o Grupo de Resposta a Incidentes de Segurança para a Internet brasileira, mantido pelo NIC.br, do Comitê Gestor da Internet no Brasil. É responsável por tratar incidentes de segurança em computadores que envolvam redes conectadas à Internet brasileira.(CERT.br, 2010).

A figura 1 demonstra o total de incidentes reportados espontaneamente ao CERT.br, no período de 1999 a setembro de 2010.



**Figura 1** – Total de incidentes reportados ao CERT.br por ano

Fonte: CERT.br (2010)

Analisando o gráfico, constatamos uma crescente de incidentes reportados, com destaque para os anos 2006 e 2009 que obtiveram um aumento considerável em relação ao seu ano anterior. As estimativas apuradas até setembro de 2010 demonstram que há uma tendência de queda nos números para o corrente ano.

Diante desse cenário, surgem ferramentas para auxiliar o administrador de redes a ter um controle eficiente sobre uma política de segurança em internet. Segundo Furmankiewicz e Figueiredo (2000), essas ferramentas são conhecidas por *firewalls*, divididos em duas categorias básicas:

- *Firewalls* de nível de rede: são analisadores de conteúdo de pacote e cabeçalho IP, operantes da camada de rede do protocolo TCP/IP;
- *Firewalls* de *gateway* de aplicativo: são analisadores de conteúdo de URL, também chamados de servidores *proxy*, que atuam na camada de aplicativo do protocolo TCP/IP.

Sobre servidores *proxy*, a Wikipedia (2010) considera:

*Proxy* é um servidor que atende a requisições repassando os dados do cliente à frente. Um usuário (cliente) conecta-se a um servidor *proxy*, requisitando algum serviço, como um arquivo, conexão, website, ou outro recurso disponível em outro servidor.

“O *proxy* é como um burocrata que não se limita a repassar as requisições: ele analisa todo o tráfego de dados, separando o que pode ou não pode passar e guardando informações para uso posterior.” (Morimoto, 2009, p.133).

Para Carmona (2007), entre as alternativas de servidores *proxy* existentes no mercado, destaca-se o *software* Squid como referência para construção de *proxy* de tráfego e de *cache*, estando presente na maioria das distribuições GNU/Linux.

Porém, esta eficiente solução possui uma administração via *console* que exige um conhecimento mais avançado sob sua manipulação, pois a aplicação não proporciona uma interface amigável ao usuário, precisando este saber utilizar muitos comandos em modo texto.

## 1.1 ESTRUTURA DO TRABALHO

A estrutura deste trabalho foi dividida em sete capítulos, apresentados a seguir:



**Capítulo 1 – Introdução** – Apresenta ao leitor uma introdução ao tema, expondo a necessidade que objetivou o desenvolvimento do sistema.

**Capítulo 2 – Objetivos** – Apresenta os objetivos geral e específicos sobre o trabalho desenvolvido e o que se pretende fazer com a ferramenta desenvolvida.

**Capítulo 3 – Motivação** – Neste capítulo destacam-se as motivações que levaram o autor ao desenvolvimento da aplicação.

**Capítulo 4 – Justificativa** – Este capítulo visa expor ao leitor uma justificativa para o desenvolvimento do projeto.

**Capítulo 5 – Referencial teórico** – Este capítulo aborda todo o referencial teórico pesquisado para desenvolvimento da aplicação, abordando conceitos como PHP, Squid, GNU/Linux, entre outros.

**Capítulo 6 – Desenvolvimento do sistema** – Neste capítulo são mostradas as interfaces da aplicação web, bem como a descrição de suas funcionalidades.

**Capítulo 7 – Conclusão** – Por fim, são apresentadas as conclusões sobre o projeto desenvolvido, destacando os resultados obtidos e as dificuldades encontradas, apresentando os trabalhos relacionados e sugerindo trabalhos futuros.

## 2 OBJETIVOS

### 2.1 OBJETIVO GERAL

Desenvolver uma ferramenta web para administração do servidor *proxy* Squid, permitindo que essa ferramenta possa ser utilizada por usuários que desconheçam o princípio de funcionamento deste servidor.

### 2.2 OBJETIVOS ESPECÍFICOS

- Cadastrar e editar usuários que terão acesso à aplicação e ao servidor *proxy*;
- Criar uma estrutura de grupos com diferentes permissões de acesso ao *proxy*;
- Permitir ao administrador do sistema a liberação ou bloqueio de sites e domínios, extensões de arquivos, endereços IP ou MAC, bloquear por palavras específicas;
- Emitir relatórios sobre os acessos realizados ao *proxy* e dados cadastrados na aplicação;
- Monitorar *logs* de acesso e *cache*; reconfigurar, parar ou reiniciar o servidor.

### 3 MOTIVAÇÃO

A motivação para realização deste trabalho dá-se inicialmente pela admiração ao *software* livre e seus benefícios.

Em conseqüência dessa admiração, outro fator em destaque é a necessidade de aprofundar os conhecimentos em sistemas operacionais GNU/Linux, onde há uma escassez de profissionais qualificados no mercado para atuarem nessa área.

Outro fator motivador que levou a confecção deste trabalho foi a necessidade de se utilizar de maneira eficiente os recursos de internet aliado a uma possibilidade de administração realizada por usuários que não contenham relação com a TI, pois nem todas as empresas possuem profissionais nesta área, delegando tais funções a gerentes ou chefes de produção.

## 4 JUSTIFICATIVA

O presente trabalho denominado SquidHelp se justifica pelo fato de possibilitar uma administração simples a usuários de menor experiência em sistemas GNU/Linux.

O SquidHelp proporcionará ao administrador uma interface amigável tendo a interação realizada através de simples seleções e formulários de texto, permitindo efetuar controle sobre as principais configurações oferecidas pelo *proxy* Squid.

Por fim, a possibilidade de emissão de relatórios de acessos através da interação com o programa SARG, aliado aos demais relatórios proporcionados pela aplicação SquidHelp, possibilitará ao administrador um eficiente controle sobre os recursos corporativos de internet, amenizando os riscos tangíveis a segurança e evitando perdas na produtividade das operações.

## 5 REFERENCIAL TEÓRICO

Este capítulo descreve os conceitos utilizados para desenvolvimento da aplicação *web* SquidHelp, servindo de referencial teórico para implantação das funcionalidades e esclarecimentos de eventuais dúvidas que surgiram durante o desenvolvimento do projeto.

### 5.1 GNU/LINUX

GNU é um sistema operacional *Unix-Like* que possui um código fonte que pode ser copiado, modificado e distribuído. Seu significado corresponde a *GNU's not Unix*, um termo acrônimo recursivo.

O projeto GNU foi iniciado em 1983 por Richard Stallman, considerado o pai do software livre, e alguns outros hackers que formaram a FSF (*Free Software Foundation*). Stallman acreditava que os usuários deveriam ser livres para fazer o que desejassem com o software que adquirissem, incluindo fazer cópias para amigos e modificar o código fonte, distribuindo-o com as alterações realizadas.

Para o desenvolvimento do sistema GNU, Stallman precisava de um editor de textos para escrever seus códigos e um compilador para tornar a linguagem de texto em linguagem de máquina, ou seja, um programa executável. Desenvolveu então o editor de texto Emacs, seguido do compilador GCC (GNU C Compiler).

Definiu também uma nova definição de licença, chamada *copyleft*, sendo uma paródia ao *copyright*, onde estipula que qualquer pessoa que fizer a redistribuição de um software livre deve passar ele com sua liberdade também, permitindo que futuras alterações e cópias deste possam acontecer. A idéia dessa modalidade de licença está em não permitir que alguém se intitule dono de uma versão, impondo restrições para o usuário. Desta maneira, um *software* livre sempre será livre.

Em 1991, o sistema operacional GNU estava praticamente pronto, mas faltava ainda um de seus principais componentes, o núcleo do sistema, também chamado de *kernel*, responsável pela execução dos programas, escalonamento das tarefas e acesso aos recursos de hardware.

Eis então que surge o estudante finlandês Linus Torvalds, com o desenvolvimento de uma versão inicial de um núcleo de sistema operacional

compatível com o UNIX e específico para computadores pessoais. Este núcleo foi batizado de LINUX, disponibilizado à internet sob a licença GNU GPL.

Segundo Carmona (2007), a partir da junção do projeto de Richard Stallman com o *kernel* desenvolvido por Linus Torvalds, nasce um dos maiores fenômenos da indústria de software de todos os tempos, o sistema operacional GNU/LINUX, um sistema robusto e confiável, totalmente grátis, obtendo uma reputação de um sistema com excelente performance e eficiência.

A distribuição GNU/Linux escolhida para desenvolvimento da aplicação *web* SquidHelp foi o Ubuntu Server 8.04, precisando estarem instalados neste servidor os programas Apache, Squid e SARG.

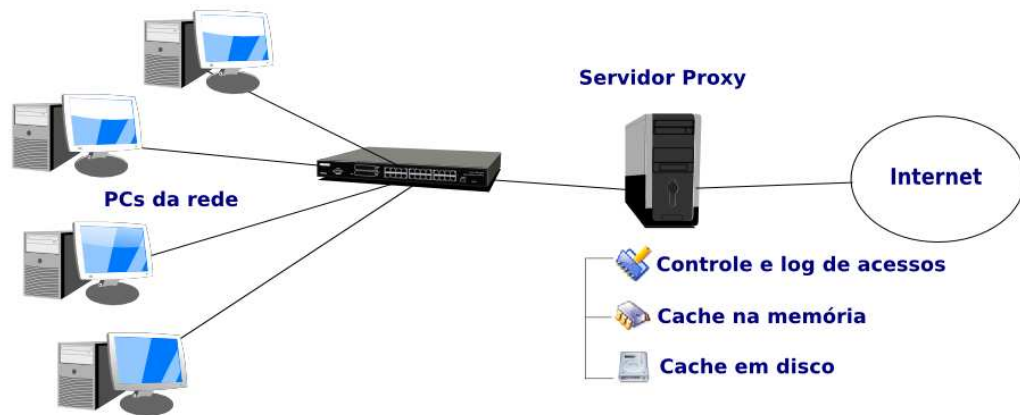
## 5.2 SQUID

O Squid é um servidor *proxy* de código-aberto presente na maioria das distribuições GNU/Linux, capaz de suportar os protocolos HTTP, HTTPS, FTP, Gopher e outros. Serve para reduzir a utilização da conexão numa rede e melhora os tempos de resposta fazendo *cache* de requisições freqüentes de páginas *web* nesta rede de computadores.( Wikipedia, 2010). O nome Squid é uma tradução que quer dizer lula.

Segundo Morimoto (2009, p.133), “O Squid permite compartilhar a conexão entre vários micros, servindo como um intermediário entre eles e a internet”. O autor destaca ainda que as vantagens de usar um servidor *proxy* são basicamente três:

1. Possibilidade de impor restrições de acesso com base em horários, *login*, endereço IP ou MAC, bloqueio de palavras, extensões de arquivos e páginas indesejadas;
2. O *proxy* funciona como um *cache* de páginas e arquivos, armazenando localmente as informações já acessadas, economizando banda de internet e tornando o acesso mais rápido;
3. Reúne todos os acessos realizados num único lugar. Desta forma, é possível visualizar os acessos posteriormente utilizando o SARG, um gerador de relatórios que transforma as longas listas de acessos dos arquivos de *logs* em páginas HTML bem formatadas. (Orso, 2010).

A figura 2 esboça o funcionamento de um servidor *proxy* numa rede de computadores:



**Figura 2** - Funcionamento servidor *proxy* numa rede

Fonte: Morimoto (2009, p. 134)

A seguir, serão descritos alguns conceitos fundamentais para melhor entendimento no funcionamento do servidor *proxy* Squid.

### 5.2.1 ACL

O Squid possui um eficiente mecanismo de controle dos acessos de internet numa rede. Através deste mecanismo, torna-se possível a criação de listas altamente customizadas, capazes de filtrar desde simples domínios até conteúdos mais específicos. (Morimoto, 2009).

As ACL's - (*Access Control Lists*) ou listas de controle de acesso, constituem a grande flexibilidade e eficiência do Squid. Através delas podem-se criar regras para controlar o acesso à Internet das mais diferentes formas. Praticamente todo o processo de controle do Squid é feito com o seu uso.

O uso das listas de controle de acesso é a parte mais importante da configuração de um servidor *proxy* Squid, pois se bem configuradas podem trazer um nível de segurança muito bom para a rede. Entretanto se mal configuradas podem ter o resultado oposto, já que além da falsa sensação de segurança não será aproveitada a principal funcionalidade do Squid.

## 5.2.2 MODO DE CONFIGURAÇÃO

O arquivo de configuração do servidor *proxy* Squid é o `squid.conf`. Este arquivo em sua versão original é composto por mais de 1500 linhas, contendo comentários e exemplos sobre quase todas as opções disponíveis pelo *proxy*. Com relação à autenticação dos usuários do *proxy*, o Squid oferece suporte a dois modos distintos, conhecidos como transparente e autenticado.

### 5.2.2.1 Proxy transparente

Sobre Proxy transparente, Morimoto (2009, p.153) destaca:

Uma garantia de que os usuários realmente vão usar o proxy e, ao mesmo tempo, uma grande economia de trabalho e dor de cabeça é o recurso de *proxy* transparente. Ele permite configurar o Squid e o firewall de forma que o servidor *proxy* fique escutando todas as conexões na porta 80. Mesmo que alguém tente desabilitar o *proxy* manualmente nas configurações do navegador, ele continuará sendo usado. Outra vantagem é que este recurso permite usar o *proxy* sem precisar configurar manualmente o endereço em cada estação. Basta usar o endereço IP do servidor rodando o *proxy* como *gateway* da rede.

Existe uma limitação importante a ser considerada nesse modo, onde o *proxy* atende apenas ao tráfego realizado pela porta 80. Como a conexão é compartilhada via NAT, todo o tráfego de outros protocolos (incluindo páginas em HTTPS, que são acessadas através da porta 443) é encaminhado diretamente, sem passar pelo *proxy*. Ou seja, embora seja uma forma simples de implementar um sistema de *cache* e algumas restrições de acesso, o uso do proxy transparente está longe de ser uma solução ideal.

### 5.2.2.2 Proxy autenticado

No modo autenticado, o usuário precisa informar suas credenciais para usufruir dos recursos do *proxy*. Morimoto (2009) destaca que este modo de configuração não permite ser usado em conjunto com o *proxy* transparente, fazendo com que seja realizada uma escolha entre as duas formas.



Conforme Vesperman (2001), o Squid utiliza processos auxiliares para processar as solicitações de autenticação para evitar que o mesmo seja parado ou bloqueado por causa de conexões lentas. Esses processos auxiliares são conectados por *pipes* Unix padrão e o Squid se comunica através de entradas e saídas padrão. Se o processo responder “OK”, a autenticação foi feita; se responder “ERR”, a autenticação falhou.

Como cada solicitação deve ser autenticada, o Squid guarda o nome de usuário e a senha junto com os retornos de autenticações bem sucedidas no seu *cache* por um período pré-determinado, permitindo que envie solicitações para cada página solicitando a autenticação ao usuário uma única vez.

Este foi o modo de autenticação escolhido para desenvolvimento da aplicação *web* SquidHelp, onde é utilizado o *script* `mysqlt_auth`, construído em PHP e baseado em autenticação de tabelas no banco de dados MySQL.

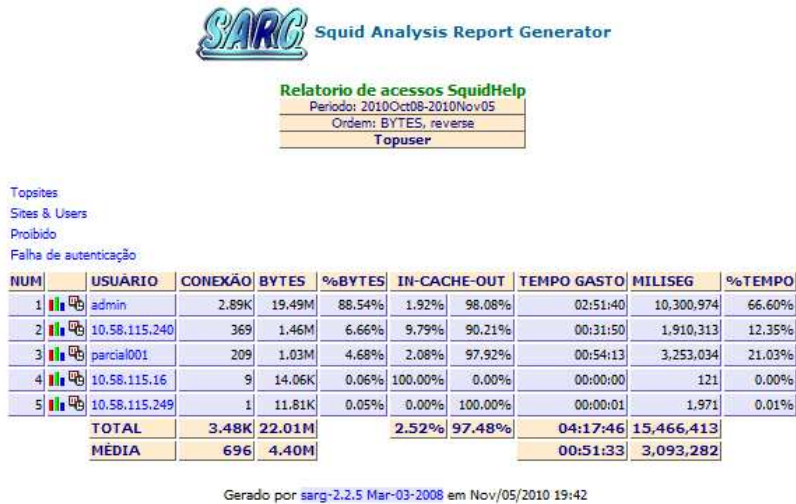
### 5.3 SARG

O SARG (*Squid Analysis Reporter Generator*) é um projeto *open source* desenvolvido por Lineo Orso com a finalidade de interpretar os acessos registrados no arquivo de *log* do programa Squid. Este programa é capaz de transformar as longas listas de acessos dos *logs* em arquivos HTML muito bem organizados. (Morimoto, 2009).

Toda vez que é executado, ele cria um conjunto de páginas organizadas por dia, contendo todos os acessos realizados e a partir de qual máquina este acesso foi realizado. Quando utilizado o *proxy* em modo autenticado, o relatório gerado mostrará o *login* dos usuários, caso contrário, será mostrado o endereço IP das máquinas.

Desta forma, é possível realizar o acompanhamento sobre as páginas que estão sendo acessadas e tomar as medidas cabíveis nos casos de abuso.

A figura 3 demonstra como é gerado um relatório através do programa SARG:



**Figura 3** - Relatório de acessos ao servidor *proxy* Squid

**Fonte:** O autor (2010).

Através da aplicação *web* SquidHelp, é possível gerar relatórios customizados interoperabilizando com o programa SARG.

## 5.4 PHP

Entre as variadas linguagens de programação existentes para *web* (Perl, Microsoft ASP.NET, Java Server Pages e outros), o PHP (*Hypertext PreProcessor*) destaca-se na geração de páginas dinâmicas pela sua facilidade e segurança, possuindo interface para muitos sistemas de diferentes banco de dados, além de ser uma ferramenta livre.

Sobre PHP, Welling e Thomson (2005, p.xxvi) descrevem:

O PHP é uma linguagem de criação de scripts do lado do servidor que foi projetada especificamente para a Web. Dentro de uma página HTML, você pode embutir código de PHP que será executado toda vez que a página for visitada. O código de PHP é interpretado no servidor Web e gera HTML ou outra saída que o visitante verá.

A figura 4 exemplifica um *script* realizado em PHP, utilizado no desenvolvimento do projeto:

```

1 <?php
2     require ("verifica.php");
3     include("connect.php");
4
5     // Pesquisa usuário pelo seu código para amostra do nome
6     $sql = "SELECT * FROM `\$tabusers` WHERE cod_usuar = ".$_SESSION['id_user'];
7     $rs = mysql_query($sql);
8     if(mysql_num_rows($rs)) {
9         // Busca na matriz o resultado correspondente ao campo do nome do usuário
10        $user = mysql_fetch_array($rs);
11        echo " Bem-vindo(a) <strong>".$user['nom_usuario']."</strong>. | ";
12    }
13
14
15     ?>

```

**Figura 4 -** Exemplo de script em PHP

Fonte: O autor (2010).

Por todos estes motivos, essa foi a linguagem de programação escolhida para desenvolvimento da aplicação *web* SquidHelp.

## 5.5 HTML

O HTML (*HyperText Markup Language*) é uma linguagem de marcação de textos utilizada para exibição de páginas na *web*, tendo seu conteúdo interpretado por navegadores.(Wikipedia, 2010).

Sobre HTML, Venetianer (1997) destaca:

HTML é a abreviação de HyperText Markup Language (Linguagem de Anotação de Hipertexto). É uma linguagem de programação muito simples, utilizada para criar documentos hipertexto, que pode ser portada de uma plataforma computacional para outra. Isto significa que você pode escrever códigos-fonte HTML sem se preocupar em qual computador e por qual sistema operacional este documento será visualizado.

Os comandos aceitos pela linguagem HTML são texto comum e códigos especiais conhecidos por *tags*.

A aplicação *web* SquidHelp utilizou HTML para confecção e exibição de suas telas.

## 5.6 JAVASCRIPT

Javascript é uma linguagem de programação utilizada para criar efeitos especiais nas páginas e definir interatividades com o usuário. O navegador do cliente é o encarregado de interpretar as instruções Javascript e executá-las para

realizar estes efeitos e interatividades, de modo que o maior recurso, e talvez o único, com que conta esta linguagem é o próprio navegador.

Sobre Javascript, Alvarez (2004) destaca:

Javascript é uma linguagem com muitas possibilidades, permite a programação de pequenos scripts, e também de programas maiores, orientados a objetos, com funções, estruturas de dados complexas, etc. Toda esta potência de Javascript se coloca à disposição do programador, que se converte no verdadeiro dono e controlador de cada coisa que ocorre na página.

A figura 3 exemplifica como deve ser realizado uso de Javascript em páginas HTML pelo padrão W3C:

```
<script type="text/javascript">  
/* aqui fica o script */  
</script>
```

**Figura 5** - Exemplo de script em Javascript

Fonte: O autor (2010).

Para o desenvolvimento da aplicação web SquidHelp, foram criadas algumas funções na linguagem Javascript para otimização das páginas.

## 5.7 CSS

CSS é a sigla em inglês para *Cascading Style Sheet* que em português foi traduzido para folha de estilo em cascata.

Maujor (2010) define CSS como: "Folha de estilo em cascata é um mecanismo simples para adicionar estilos (p.ex., fontes, cores, espaçamentos) aos documentos *web*."

O autor destaca ainda que:

A grande vantagem do uso de CSS é a de separar a marcação HTML, da apresentação do site. Em outras palavras, vale dizer que o HTML destina-se unicamente a estruturar e marcar o conteúdo, ficando por conta das CSS toda a responsabilidade pelo visual do documento.

A figura 4 exemplifica a forma como é composto um arquivo CSS:

```

1  body {
2  /*background:#F5F5F5 url(../images/fundo.jpg)repeat-x fixed top left*/
3  background:#F5F5F5;
4  font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
5  font-size: 14px;
6  }
7
8  .imgpos {
9  position: absolute;
10 top:50px;
11 left:300px;
12 }
13
14 .imgpos2 {
15 position: absolute;
16 top:40px;
17 left:440px;
18 }
19
20 .imgerro {
21 position: absolute;
22 top:40px;
23 left:480px;
24 }
25
26 a:link {text-decoration: none}
27 a:visited {text-decoration: none;
28 color: #444;}
29 a:hover {text-decoration: none;
30 color: #FFCB3B;
31 }
32 a:active {text-decoration: none}
33

```

**Figura 6** - Exemplo de arquivo CSS

Fonte: O autor (2010).

Para o desenvolvimento aplicação web SquidHelp, arquivos CSS foram criados para uma melhor estilização e visualização de páginas e menu do sistema.

## 5.8 AJAX

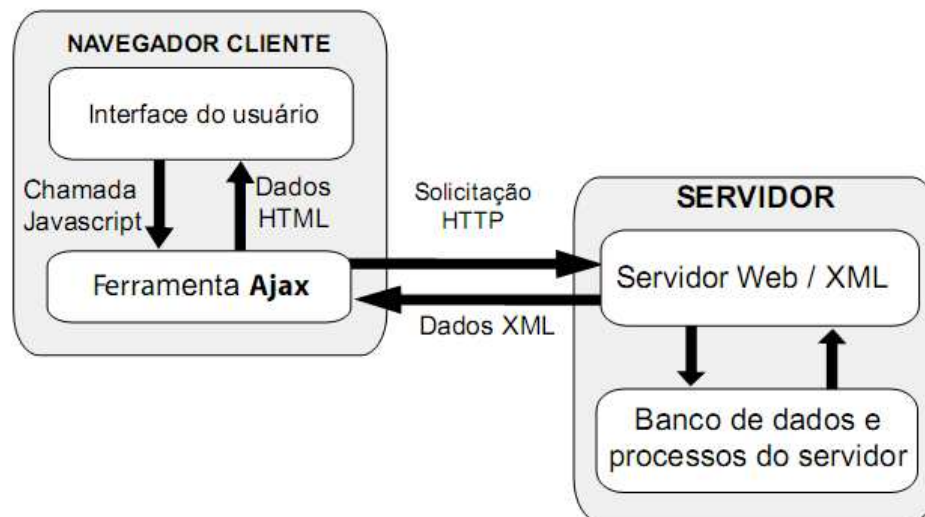
Ajax é a sigla para *Asynchronous Javascript and XML*. Parte da idéia de utilizar Javascript para transformar páginas em aplicações, de modo que não precise recarregar a tela cada vez que o usuário clicar em alguma coisa.

Sobre Ajax, Soares (2006) destaca:

AJAX significa Asynchronous Javascript and XML (JavaScript e XML assíncrono), porém o que temos é muito mais que a junção de JavaScript com XML. É um conceito de navegação e atualização de páginas web (algumas partes descritas não são novas, denominadas muitas vezes de DHTML (HTML Dinâmico) e Script Remoto).

Com o AJAX não é necessário que a atualização de determinadas informações implique na modificação da página inteira, pois é possível realizar a busca das informações no servidor web dentro da própria página, através de alguns componentes disponíveis no JavaScript e um pouco de manipulação do HTML, como se fosse um documento XML via API DOM.

O modelo de uma aplicação que utilize Ajax está representado na figura 5:



**Figura 7** - Modelo de aplicação com Ajax

Fonte: Niederauer (2010, p.17)

Diferente do modelo clássico de aplicação *web*, nesse modelo a interação entre o navegador e o servidor *web* não ocorre de forma totalmente direta, mas por meio da ferramenta Ajax. Como o Ajax é ativado por uma chamada JavaScript, o usuário pode permanecer visualizando a página normalmente enquanto ocorre a comunicação com o servidor *web*.

Essa tecnologia também se encontra presente na aplicação *web* desenvolvida, onde foram utilizados *frameworks* para Ajax *spry* - validação dinâmica de campos - e *sexyalertbox* para melhor aparência nas janelas de alerta.

## 5.9 APACHE

O Apache é um servidor *web open source*, sendo a principal tecnologia da *Apache Software Foundation*. Ele encontra-se disponível em diversas plataformas de sistemas operacionais, como Windows, Novell Netware, OS/2 e diversos outros do padrão POSIX (Unix, Linux, FreeBSD, etc...).

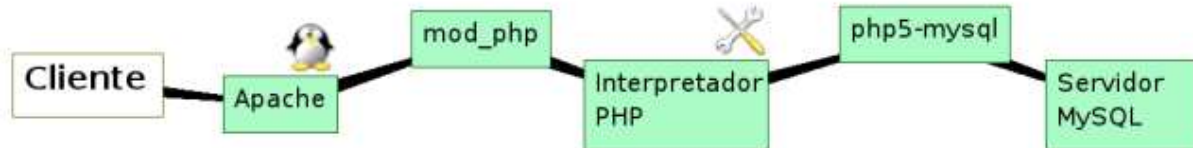
Segundo Morimoto (2009, p.347) “Os servidores *web* são a espinha dorsal da internet [...]”. Estes servidores hospedam as páginas, servindo de base para toda aplicação *web*. Cabe a eles a responsabilidade de aceitar pedidos HTTP de clientes e servi-los com respostas a estes pedidos.

Sobre o servidor Apache, Morimoto (2009, p.351) ainda descreve:

A principal característica do Apache é a modularidade. Ao invés de ser um aplicativo grande e complexo, que tenta desempenhar sozinho todas as funções, o Apache se limita a executar uma única

tarefa: entregar páginas html e outros tipos de arquivos aos clientes. Qualquer outra coisa é invariavelmente feita por um módulo externo.

A figura 6 ilustra como os passos realizados pelo servidor web para realizar a entrega de uma página ao usuário em tempo hábil:



**Figura 8** - Passo a passo servidor Apache

Fonte: Morimoto (2009, p.351).

Desta forma, o autor destaca que através da divisão de tarefas (modularidade), o servidor Apache consegue obter uma maior rapidez e segurança.

Devido a estes fatores, o servidor *web* Apache foi o escolhido para hospedagem da aplicação SquidHelp.

## 5.10 MYSQL

O MySQL é um SGBD extremamente versátil, usado para os mais diversos fins. Permite acesso a seu banco de dados a partir de *scripts* em PHP, através de um aplicativo desenvolvido em C ou C++, ou praticamente qualquer outra linguagem (Morimoto, 2009).

Sobre MySQL, Welling e Thomson (2005, p.165) consideram:

O MySQL é um sistema de gerenciamento de banco de dados relacional (relational database management system - RDBMS) poderoso e muito rápido. Um banco de dados permite armazenar, pesquisar, classificar e recuperar dados de forma eficiente. O servidor de MySQL controla o acesso aos dados para assegurar que vários usuários possam trabalhar com os dados ao mesmo tempo, fornecer acesso rápido aos dados e assegurar que somente usuários autorizados obtenham acesso. Portanto, o MySQL é um servidor multiusuário e multiencadeado (ou multitbreaded). Ele utiliza SQL (Structured Query Language), a linguagem de consulta padrão de banco de dados em todo o mundo.

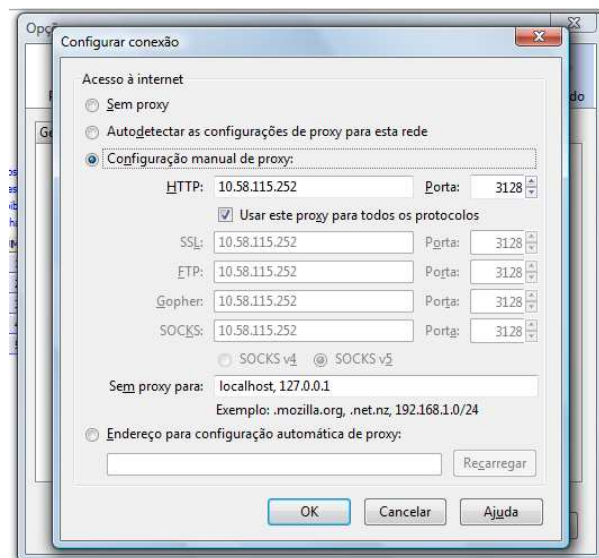
Por conter uma fácil integração com GNU/Linux, Apache e PHP, formadores do conjunto LAMP, MySQL foi o SGBD escolhido para armazenamento de dados da aplicação SquidHelp.

## 6 DESCRIÇÃO DO SISTEMA

### 6.1 FUNCIONAMENTO

SquidHelp foi desenvolvido nos moldes de uma aplicação *web* - um sistema que é executado em um servidor HTTP e que pode ser acessado por equipamentos conectados a rede através do mesmo - escrito nas linguagens HTML e PHP, utilizando MySQL como SGBD. O modo de autenticação adotado foi o script `mysqlt_auth`, criado em PHP, que valida a autenticação do usuário ao *proxy* baseado em tabela do SGBD MySQL. Devido ao uso de autenticação no servidor *proxy*, torna-se preciso “setar” manualmente no navegador o endereço deste servidor.

A figura 9 exemplifica como deve estar configurado o navegador para utilização da ferramenta SquidHelp:



**Figura 9** - Exemplo de configuração para *proxy* em modo autenticado

Fonte: O autor (2010).

O sistema SquidHelp foi desenvolvido para conter um único perfil de usuário, sendo este o administrador do sistema, com permissão total sobre as funcionalidades da aplicação *web*.

O acesso é dado mediante autenticação do usuário, onde a partir do *login* e senha fornecidos, realiza-se um pesquisa na tabela de usuários do SGBD MySQL. Esta tabela é alimentada através das inserções da interface de cadastro de usuários, que utiliza o *framework spry* para validação dinâmica de seus campos. Esta interface



contém os dados de *login* e senha (criptografada pelo *hash* do algoritmo MD5), grupo de acesso do usuário junto ao *proxy* Squid (são três grupos possíveis: TOTAL, PARCIAL ou RESTRITO) e um *flag* utilizado para definir se o usuário também terá acesso ao sistema SquidHelp, pois a base de autenticação utilizada pelo *proxy* Squid e pela aplicação web SquidHelp é a mesma. Também foi desenvolvida uma tela para manutenção de cadastros, onde o administrador poderá alterar a senha dos usuários, ativar/inativar seu acesso ao *proxy*, permitir ou negar a utilização da aplicação SquidHelp e alterar o grupo de privilégios de acesso.

Caso negada a sua entrada, seja por motivos de inativação ao *proxy* ou não permissão à aplicação SquidHelp, será apresentado ao usuário uma tela solicitando o contato junto ao administrador da rede. Estando autorizada sua entrada, este será redirecionado para a tela principal, onde poderá utilizar os quatro módulos apresentados pelo sistema:

1. Módulo de Usuários: cadastrar usuários, manutenção de usuários;
2. Módulo Administrar ACL's: bloquear ou liberar sites e domínios, endereços IP e MAC, palavras e extensões de arquivos;
3. Módulo de Scripts no *proxy*: monitorar em tempo real o *log* de acessos e *cache*; reconfigurar, para ou reiniciar o servidor Squid ;
4. Módulo de Relatórios: gerar relatórios de acesso, interagindo com o sistema SARG e gerar relatórios pdf sobre cadastros efetuados na aplicação.

O sistema é hospedado pelo servidor *web* Apache e possui controle de sessões e *cookies* a cada autenticação, evitando acesso indevido as páginas da aplicação sem a devida autenticação.

Foram necessárias seis tabelas para construção do banco de dados, que comportam: cadastro de usuários, sites, endereços IP, endereços MAC, palavras e extensões de arquivos. Estas tabelas são alimentadas pelos seus respectivos formulários, descritos a seguir.

Devido ao arquivo de configuração do servidor Squid - `squid.conf` - não obter acesso diretamente às tabelas, este precisa buscar as informações de autenticação, bloqueios ou liberações através de simples arquivos em texto puro.

Para suprir esta necessidade, desenvolveu-se *scripts* para que ao final de cada inserção nas tabelas do SGBD, fosse também adicionada esta informação em um arquivo com seu nome e local já pré-definidos nas ACL's do arquivo

`squid.conf`, para que o *proxy* consiga “entender” os acessos que precisa negar ou liberar, bem como a autenticação dos usuários na rede.

A seguir serão apresentadas as interfaces da aplicação *web* desenvolvida, bem como seus alertas e tratamentos de erros.

## 6.2 LOGIN DO SISTEMA

Ao digitar no navegador o endereço da aplicação, o usuário será direcionado a esta tela, solicitando suas credenciais para acesso ao SquidHelp.

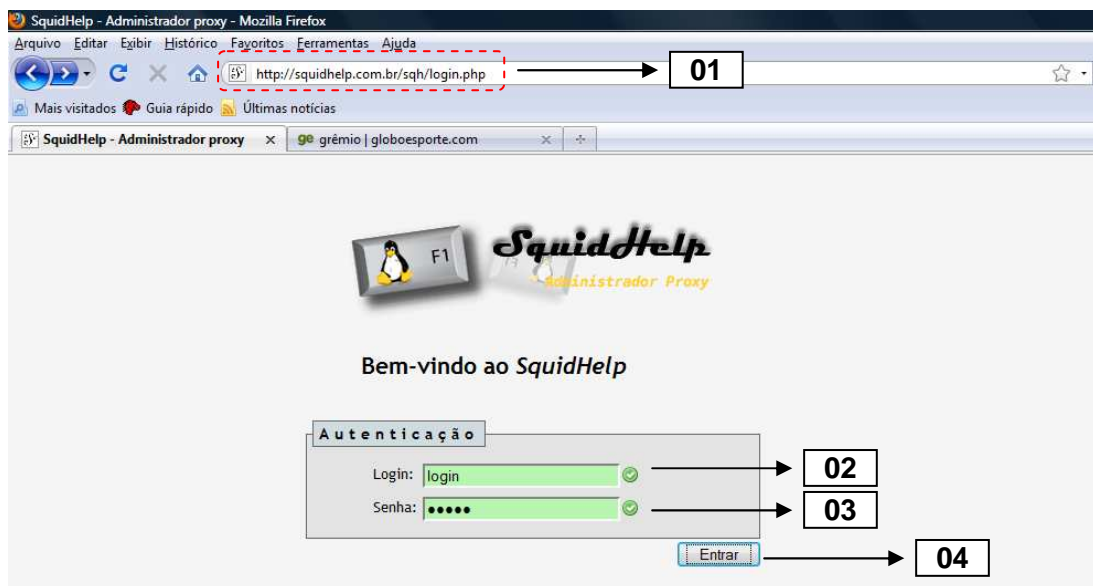


Figura 10 - Tela de *login*

### 6.2.1 Descrição da interface

01 – Local a ser digitado o endereço do servidor web, seja pelo endereço IP ou pelo nome qualificado.

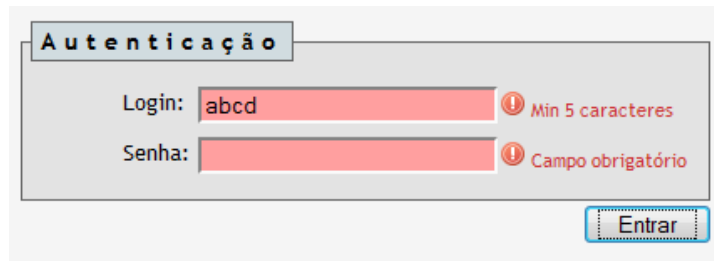
02 – Formulário de texto a ser digitado o login previamente cadastrado na aplicação.

03 – Formulário de texto a ser digitada a senha do usuário.

04 – Botão que submete à validação os dados fornecidos pelo usuário.

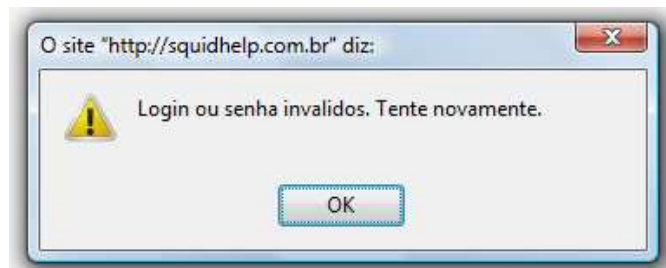
## 6.2.2 Tratamento de erros

Abaixo serão apresentadas as interfaces de tratamento de erros para falhas de autenticação.



A captura de tela mostra uma caixa de diálogo intitulada "Autenticação". Ela contém dois campos de entrada: "Login:" e "Senha:". O campo "Login:" contém o texto "abcd" e possui uma mensagem de erro vermelha que diz "Min 5 caracteres". O campo "Senha:" está vazio e possui uma mensagem de erro vermelha que diz "Campo obrigatório". Abaixo dos campos, há um botão "Entrar".

**Figura 11** - Campos obrigatórios tela de *login*



**Figura 12** - Login ou senha inválidos



**Figura 13** - Usuário sem permissão ao SquidHelp

### 6.2.2.1 Descrição dos erros

Figura 11 – Quando o usuário tentar acesso sem o preenchimento de algum campo, a validação dinâmica através do *framework spry* impedirá-lo, notificando-o sobre o motivo. Observe também que nessa tela, o campo *login* possui uma validação extra, com obrigatoriedade sobre uma quantidade mínima de 5 caracteres.

Figura 12 – Esta figura representa o alerta apresentado ao usuário no caso de divergências em um dos campos preenchidos, login ou senha.

Figura 13 – Por fim, esta tela será apresentada ao usuário quando sua autenticação for válida, porém este não possui acesso a aplicação SquidHelp, estando irregular em duas possíveis situações: **a)** estar inativo como usuário do *proxy* ou **b)** não possuir permissão de acesso à aplicação, sendo apenas usuário do *proxy*.

### 6.3 PAINEL DE CONTROLE

Estando autorizada sua entrada, o usuário será redirecionado para a página principal do sistema, o painel de controle da aplicação SquidHelp.



Figura 14 - Painel de controle

#### 6.3.1 Descrição da interface

01 – Texto de boas-vindas ao usuário, onde através de uma função busca o nome do usuário cadastrado no banco de dados.

02 – Opção para o usuário abandonar o sistema, destruindo sua sessão criada anteriormente. Será apresentado um alerta confirmando sua saída, descrito em breve.

03 – Área com os módulos pertencentes ao SquidHelp, permitindo a navegação através de clique na imagem correspondente ou utilização do menu lateral.

04 – Menu lateral *drop down*, construído em CSS, para que o usuário possa acesso rápido a todas funcionalidades do sistema.

## 6.4 MENU LATERAL

A partir desta tela, este menu estará presente em todas as demais interfaces do sistema, com o intuito de proporcionar um acesso mais direto ao usuário.

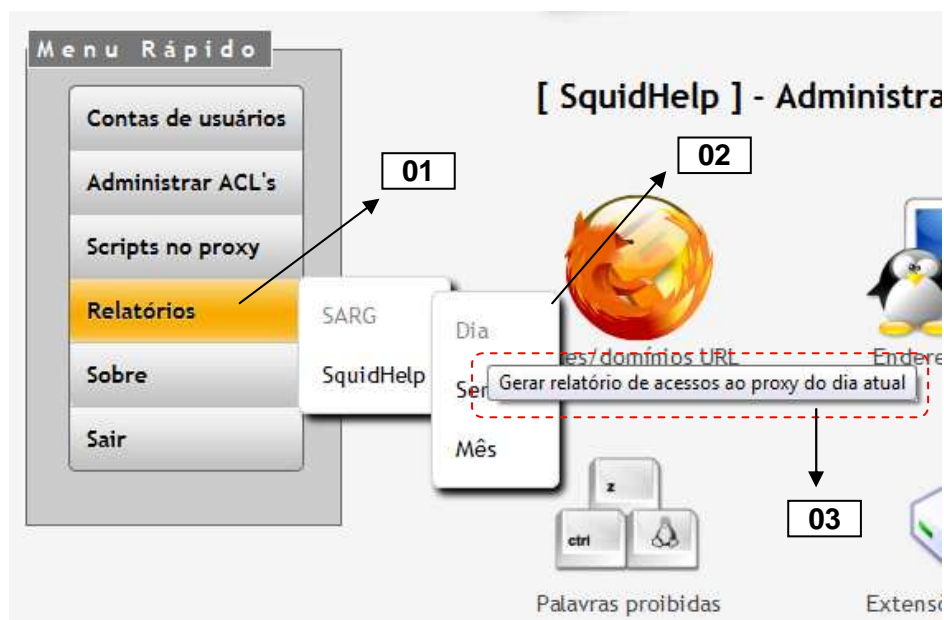


Figura 15 - Menu Rápido

### 6.4.1 Descrição da interface

01 – Recurso *hover* proporcionado pelo CSS, onde apresenta um destaque a opção selecionada, mudando sua cor ao passar o mouse.

02 – Sub-menus e derivados que formam uma cascata. Quando clicados levam até a página requisitada.

03 – Recurso *hint*, utilizado como uma breve explicação sobre o que fará a opção selecionada.

## 6.5 MÓDULO DE USUÁRIOS

Este módulo permite o cadastro dos usuários do *proxy* e SquidHelp, bem como a manutenção destas contas.



Figura 16 - Configuração de usuários

### 6.5.1 Novo usuário

Esta interface permite o cadastro dos usuários ao *proxy*, utilizando validação dinâmica em seus campos. Define o *login* e senha do usuário, bem como seu grupo de privilégios de acesso ao *proxy*, além da liberação a aplicação *web* SquidHelp.

Figura 17 - Novo usuário

#### 6.5.1.1 Descrição da interface

01 – Formulário de texto a ser preenchido o nome completo do usuário.

02 – Formulário de texto a ser preenchido o *login* escolhido pelo usuário que será utilizado para acesso ao *proxy* ou a aplicação SquidHelp.

03 – Formulário de texto a ser preenchida a senha do usuário.

04 – Formulário de texto utilizado para confirmação da senha digitada.

05 – Caixa de seleção utilizada para selecionar um dos três grupos de privilégios possíveis ao usuário, onde:

- Restrito: corresponde aos usuários que terão acesso à apenas as páginas liberadas pelo administrador, bem como apenas as extensões de arquivos permitidas por este, sendo um grupo extremamente limitado pela aplicação;
- Parcial: corresponde aos usuários que terão acesso a qualquer página com exceção das páginas bloqueadas pelo administrador. A regra para extensões de arquivo prevalece a este grupo também;
- Total: corresponde ao grupo de administradores, com permissão total para acesso, sem qualquer tipo de restrição.

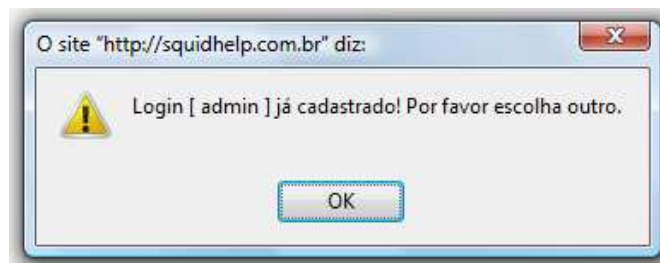
06 – Caixa de seleção utilizada para permitir ou não o acesso ao SquidHelp.

07 – Conjunto de botões destinados a submeter os dados para cadastro ou limpar os formulários preenchidos, respectivamente.

#### 6.5.1.2 Tratamento de erros

As figuras abaixo representam o tratamento de erros realizado pela aplicação SquidHelp com relação ao cadastro de usuários.

**Figura 18** - Campos obrigatórios tela novo usuário



**Figura 19** – Alerta de *login* já cadastrado

Onde:

Figura 18 – esta interface possui tratamento de erros em todos os seus campos, sendo que alguns deles possuem mais de um tipo de validação, como *login* e senha. O campo [ Repita a senha ] possui uma validação de confirmação, ou seja, verifica se o texto digitado é igual ao campo [ Senha ].

Figura 19 – Este alerta é apresentado quando ocorrer a tentativa de inserção de um *login* já existente no banco de dados, proporcionando um texto amigável ao usuário.

#### 6.5.1.3 Cadastro efetuado com sucesso

Quando o cadastro for efetuado com sucesso, será apresentada uma tela de aviso ao usuário, possibilitando o cadastro de um novo usuário ou acesso ao painel de controle.



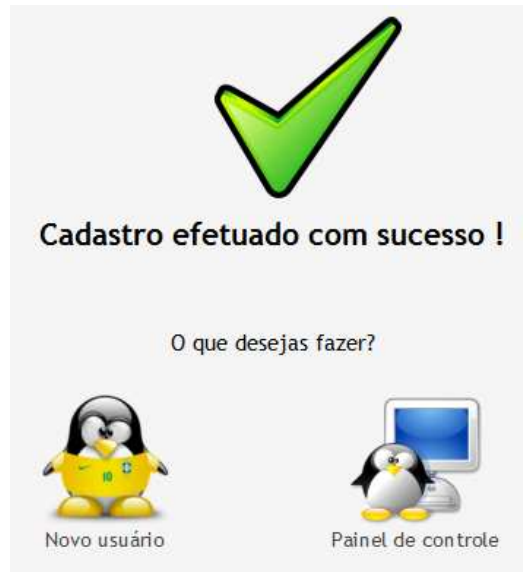


Figura 20 - Tela cadastro de usuário efetuado com sucesso

### 6.5.2 Manutenção de conta

Esta interface foi desenvolvida para possibilitar uma melhor administração dos usuários cadastrados na aplicação SquidHelp.

[ SquidHelp ] - Manutenção de conta

**Manutenção**

Login:  ✓ → 04

Nova senha:  → 06

Repita a senha:  → 07

Ativo:   Alterar senha? → 05

Grupo:  Utiliza SquidHelp:  → 10

→ 11

Login:   → 02

Código	Login	Nome usuário	Ativo	SquidHelp	Grupo	Editar
5	admin	Administrador SQH	SIM	SIM	TOTAL	<input type="button" value="Editar"/> → 03

Figura 21 - Tela manutenção de contas de usuário

### 6.5.2.1 Descrição da interface

A numeração dispersa na figura 21 representa o fluxo de funcionamento desta interface, descrito abaixo:

01 – Formulário de texto utilizado para pesquisa através do *login* do usuário, podendo este ficar em branco. Nesse caso, serão listados os dez primeiros registros encontrados no banco de dados.

02 – Botão pesquisar, utilizado para ativar a consulta realizada a partir do conteúdo do campo *login*. Pode ser substituído pela tecla [ Enter ].

03 – Tabela com o resultado da pesquisa realizada, trazendo na última coluna um ícone que possibilita a edição do usuário selecionado, passando-o para edição através do seu código, sendo este a chave primária no banco de dados.

04 – Após clicado no ícone de edição, o formulário acima será preenchido com os dados do usuário contido no banco de dados, iniciando com a caixa de texto de *login*. Este valor permite ser alterado.

05 – Este *checkbox* define se será alterada ou não a senha do usuário, sendo que apenas quando marcado habilita a digitação nos itens 06 e 07.

06 – Caixa de texto para digitação da nova senha do usuário, por padrão ela vem desabilitada, mantendo a mesma senha anterior do usuário junto ao banco de dados.

07 – Caixa de texto para confirmação de senha, por padrão também está desabilitada.

08 – Caixa de seleção para ativação ou inativação dos usuários ao *proxy* e aplicação SquidHelp. É preenchida com o valor contido no banco de dados do usuário selecionado.

09 – Caixa de seleção para alteração do grupo de privilégios de acesso, já descrito no capítulo 6.4.1.1. É preenchida com o valor contido no banco de dados do usuário selecionado. Além da alteração no banco de dados, será gravado também no arquivo correspondente ao novo grupo, para que o servidor Proxy permita os novos privilégios deste usuário.

10 – Caixa de seleção utilizada para permitir ou negar acesso a aplicação web SquidHelp. É preenchida com o valor contido no banco de dados do usuário selecionado.

11 – Conjunto de botões utilizados para submeter a alteração ou limpar os textos informados no formulário, respectivamente.

### 6.5.2.2 Tratamento de erros

A figura abaixo representa o tratamento de erros realizado pela aplicação SquidHelp com relação a manutenção de contas de usuário.

**[ SquidHelp ] - Manutenção de conta**

**M a n u t e n ç ã o**

Login:  **! Campo obrigatório!** → **01**

Nova senha:

Repita a senha:

Ativo:   Alterar senha?

Grupo:  Utiliza SquidHelp:

Login:

Código	Login	Nome usuário	Ativo	SquidHelp	Grupo	Editar
Nenhum registro encontrado! Verifique e tente novamente.						

→ **02**

**Figura 22** - Tratamento de erros da tela manutenção de usuários

Onde:

01 – No caso do usuário clicar no botão [ Alterar ] sem nenhuma conta selecionada, o sistema notificará sobre a obrigatoriedade dos campos.

02 – Caso o *login* pesquisado não seja encontrado, a aplicação notificará o usuário através de uma mensagem amigável, sugerindo uma nova tentativa.

### 6.5.2.3 Cadastro efetuado com sucesso

Quando a alteração for efetuada com sucesso, será apresentada uma tela de aviso ao usuário, como na figura 23.



Figura 23 - Alerta de dados alterados com sucesso

## 6.6 MÓDULO ADMINISTRAÇÃO DE ACL'S

Este é o módulo chave da aplicação SquidHelp, pois a partir dele será definida a política de liberações e bloqueios para as listas de controle de acesso junto ao servidor *proxy* Squid.



Figura 24 - Módulo Administrar ACL's

### 6.6.1 Descrição da interface

01 – Sites/domínios URL: através desta interface será possível realizar liberações e bloqueios de acessos sobre determinada URL.

Exemplo de formatação aceita pelo servidor Squid: www.playboy.abril.com.br, orkut.com, .youtube.

02 – Endereços IP: através desta ACL será possível realizar a liberação ou bloqueio de acessos através do endereço IP. Esta interface possui mascaramento em seu campo de texto, aceitando apenas formatações que correspondam a um endereço IP. A regra de liberação possui prioridade sobre o bloqueio, ou seja, caso um endereço IP conste nas duas situações, este terá seu acesso liberado.

Exemplo de formatação aceita pelo servidor Squid: 192.168.2.99, 10.0.0.1, 200.234.21.23.

03 – Endereços MAC: através desta ACL será possível realizar a liberação ou bloqueio de acessos através do endereço MAC. Esta interface possui mascaramento em seu campo de texto, convertendo todo texto digitado para caracteres em MAIÚSCULO e adicionando o caractere [ : ] (dois pontos) a cada dois dígitos. A regra de liberação possui prioridade sobre o bloqueio, ou seja, caso um endereço MAC conste nas duas situações, este terá seu acesso liberado.

Exemplo de formatação aceita pelo servidor Squid: 00:E0:4C:F5:A2:65, 00:17:ad:00:86:0c

04 – Palavras proibidas: nesta interface serão registradas as palavras que terão seu acesso bloqueado. Esta interface não possui a opção de liberação, apenas bloqueio.

Exemplo de formatação aceita pelo servidor Squid: orkut, xxx, sexo, teens, warez

05 – Extensões de arquivos: através desta ACL será possível realizar a liberação ou bloqueio de acessos de extensões de arquivos. O campo de texto possui mascaramento, permitindo apenas 4 dígitos. Assim como as demais regras, a regra de liberação possui prioridade sobre o bloqueio, ou seja, caso uma extensão de arquivo conste nas duas situações, esta terá seu acesso liberado.

Exemplo de formatação aceita pelo servidor Squid: avi, exe, mp3, pdf, php, htm.

Por serem interfaces exatamente idênticas e possuírem os mesmos tratamentos de erros, com exceção à interface [ Palavras proibidas ] que não possui a opção de liberação, será descrito apenas a interface [ Sites/domínios URL ], servindo como exemplo às demais opções de ACL's.

### 6.6.2 Sites/domínios URL

Através desta interface, torna-se possível a liberação ou bloqueio de sites ou domínios completos através da sua URL.



Figura 25 - Tela domínios/sites

#### 6.6.2.1 Descrição da interface

01 – Caixa de seleção que permite o bloqueio ou liberação de determinado domínio/site. Os endereços aqui liberados poderão ser acessados por todos os usuários da aplicação, pois formarão a chamada lista branca, sendo que esta será a única lista permitida aos usuários do grupo RESTRITO. Os sites aqui bloqueados formarão a chamada lista negra e só poderão ser acessados pelos usuários do grupo TOTAL.

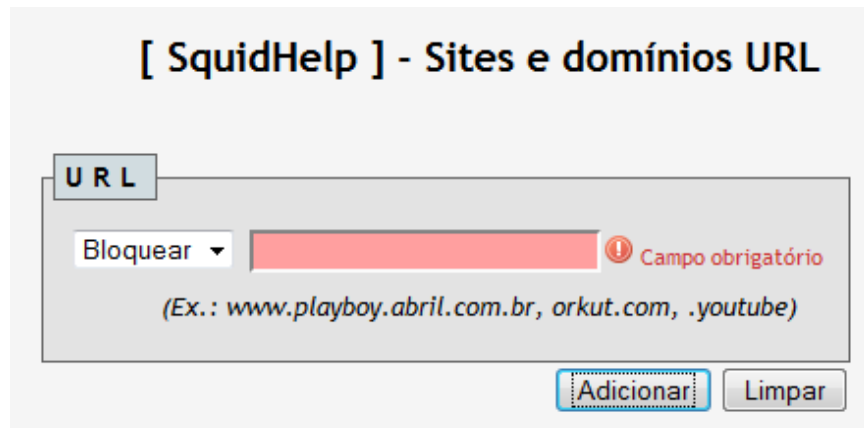
02 – Caixa de texto para digitação do endereço URL do domínio/site.

03 – Exemplos de formatação aceitos pelo servidor Squid

04 – Conjunto de botões utilizado para adicionar ou limpar o conteúdo digitado, respectivamente.

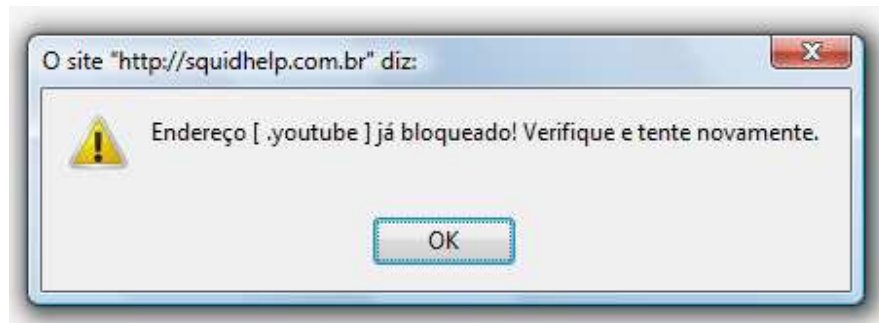
#### 6.6.2.2 Tratamento de erros

A seguir, serão apresentados as telas de tratamento de erros para a interface de Sites e Domínios URL.



The screenshot shows a web interface titled "[ SquidHelp ] - Sites e domínios URL". Below the title is a form with a label "URL" in a blue box. The form contains a dropdown menu set to "Bloquear", a red input field, and a red warning icon with the text "Campo obrigatório". Below the input field, there is an example: "(Ex.: www.playboy.abril.com.br, orkut.com, .youtube)". At the bottom right of the form are two buttons: "Adicionar" and "Limpar".

**Figura 26** - Campo obrigatório



**Figura 27** - Alerta sobre endereço já bloqueado



**Figura 28** - Alerta sobre endereço já liberado

Onde:

Figura 26 – Campo obrigatório, não será permitido a inserção caso a caixa de texto não contenha nenhum valor.

Figura 27 – Alerta emitido ao usuário na tentativa de inserção sobre um endereço já bloqueado.

Figura 28 – Alerta emitido ao usuário na tentativa de inserção sobre um endereço já liberado. Os endereços liberados são exceções junto ao servidor *proxy* Squid, ou seja, tem prioridade sobre os endereços bloqueados. Caso um endereço esteja nas duas situações, este terá seu acesso autorizado.

### 6.6.2.3 Cadastro efetuado com sucesso

Quando o bloqueio ou liberação for efetuado com sucesso, será apresentada uma tela de aviso ao usuário, como nas figuras 29 e 30.



**Figura 29** - Alerta sobre endereço bloqueado com sucesso



**Figura 30** - Alerta sobre endereço liberado com sucesso

## 6.7 MÓDULO SCRIPTS NO PROXY

Este módulo corresponde à comunicação entre a aplicação *web* SquidHelp e o servidor *proxy* Squid, permitida através de *scripts* criados em PHP com permissão de execução junto ao servidor GNU/Linux Ubuntu 8.04.



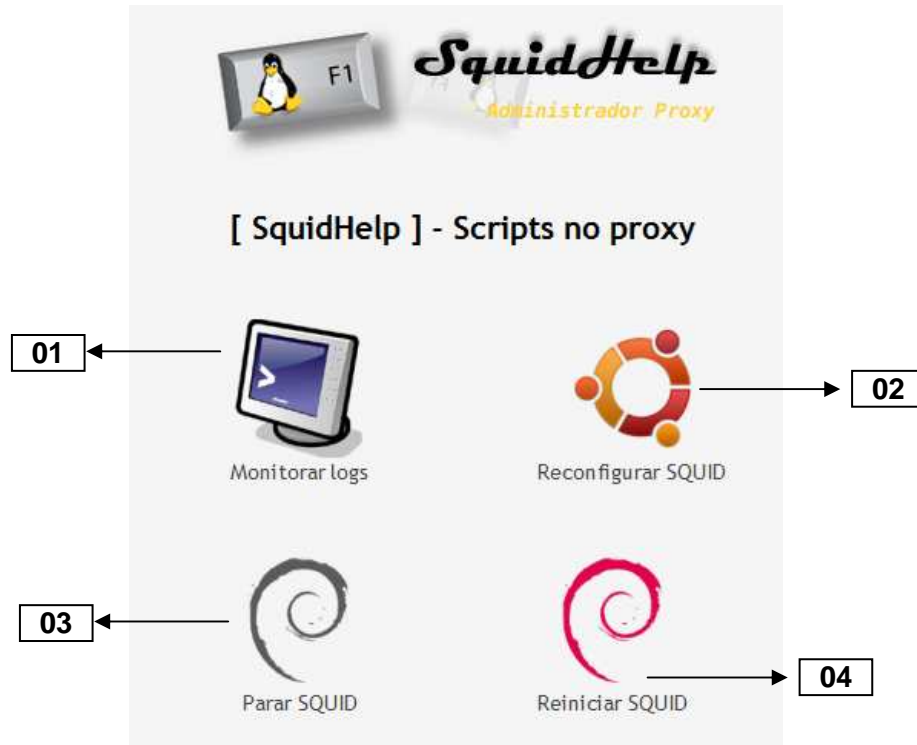


Figura 31 - Módulo de scripts no proxy

### 6.7.1 Descrição da interface

01 – Interface criada para auxiliar o administrador com relação ao monitoramento de *logs* do *proxy* Squid, permitindo a visualização das últimas dez linhas dos arquivos *access.log* e *cache.log*.



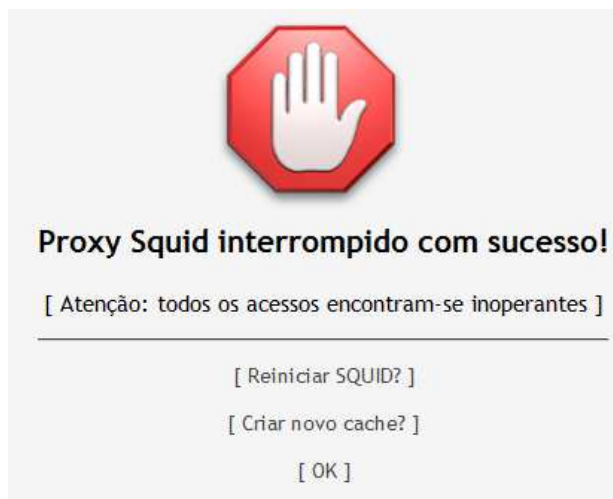
Figura 32 - Tela monitoramento dos arquivos *access.log* e *cache.log*

02 – Reconfigurar Squid: interface criada para realizar uma reconfiguração ao Squid, não precisando este parar o *proxy* para aplicar as alterações. Desta forma, os usuários que efetuarem novos acessos já receberão as alterações efetuadas. Equivale ao comando `# squid -k reconfigure` digitado em console.



**Figura 33** - Mensagem de *proxy* reconfigurado com sucesso

03 – Parar Squid: interface criada para interromper o funcionamento do Squid, tornando inoperante todos seus acessos até que seja reiniciado. Pode ser usado para manutenções no *proxy*, como a reconfiguração do seu *cache*. Equivale ao comando `# squid stop` digitado em console.



**Figura 34** - Mensagem de *proxy* interrompido com sucesso

04 – Reiniciar Squid: interface criada para realizar um reinício ao Squid, realizando um `stop` e `start` no *proxy*. Pode ser usado quando ocorrer um funcionamento inesperado do *proxy* ou para aplicação de novas alterações nas ACL's, sendo que executa uma nova leitura ao arquivo `squid.conf`. Equivale ao comando `# squid restart` digitado em console.



Figura 35 - Mensagem de *proxy* reiniciado com sucesso

## 6.8 MÓDULO DE RELATÓRIOS

Este módulo foi criado para proporcionar ao usuário da aplicação SquidHelp um melhor controle sobre a utilização do *proxy* Squid. Foi dividido em duas etapas que comportam:

1. Relatório de acessos dos usuários sobre as páginas navegadas, interagindo com o sistema SARG;
2. Relatórios dos dados cadastrados na aplicação SquidHelp, buscados a partir do banco de dados MySQL.



Figura 36 - Módulo de relatórios

### 6.8.1 Relatórios de acessos

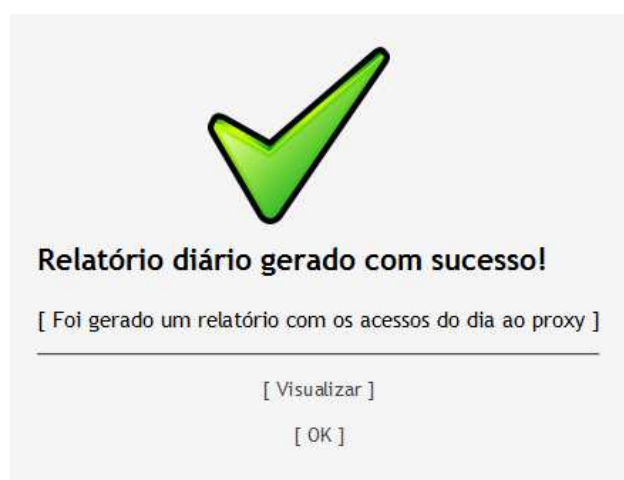
Nesta seção, será possibilitado ao usuário a emissão de relatórios sobre os acessos ao Proxy Squid, através da execução de scripts em PHP e Shell script. São oferecidos relatórios pré-definidos ou customizado.



**Figura 37** - Relatórios de acessos - SARG

#### 6.8.1.1 Descrição da interface

01 – Opção de relatório pré-definido para geração dos acessos do dia atual ao Proxy, com a possibilidade de posterior visualização através do programa SARG.



**Figura 38** - Relatório com os acessos do dia ao proxy

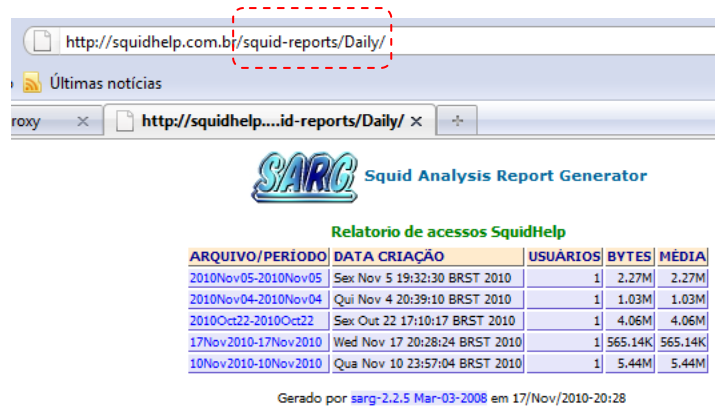


Figura 39 - Visualização de relatórios gerados no diretório *Daily*

02 – Opção de relatório pré-definido para geração dos acessos de uma semana, correspondentes a sete dias atrás até o dia atual, com a possibilidade de posterior visualização através do programa SARG.

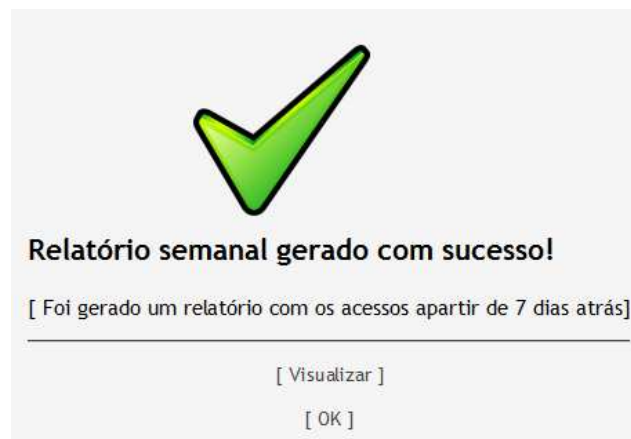


Figura 40 - Relatório com os acessos de uma semana ao *proxy*

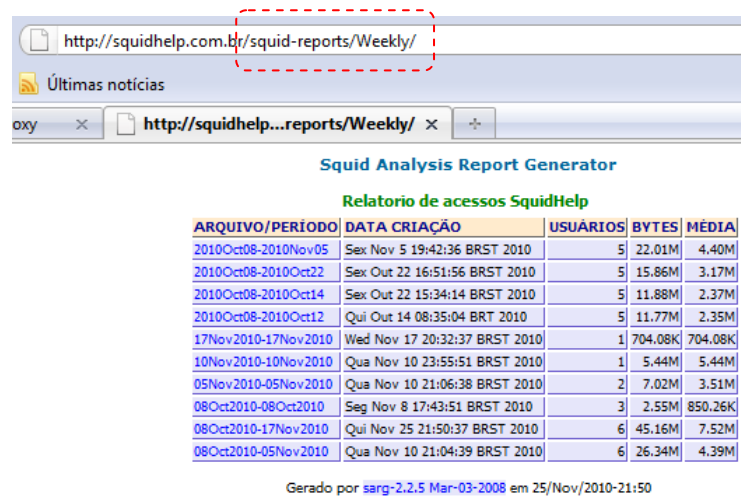


Figura 41 - Visualização de relatórios gerados no diretório *Weekly*

03 – Opção de relatório pré-definido para geração dos acessos de uma mês, correspondentes a trinta dias atrás até o dia atual, com a possibilidade de posterior visualização através do programa SARG.

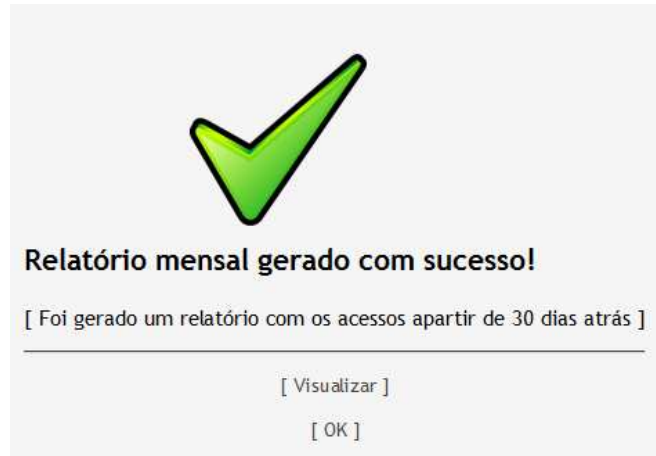


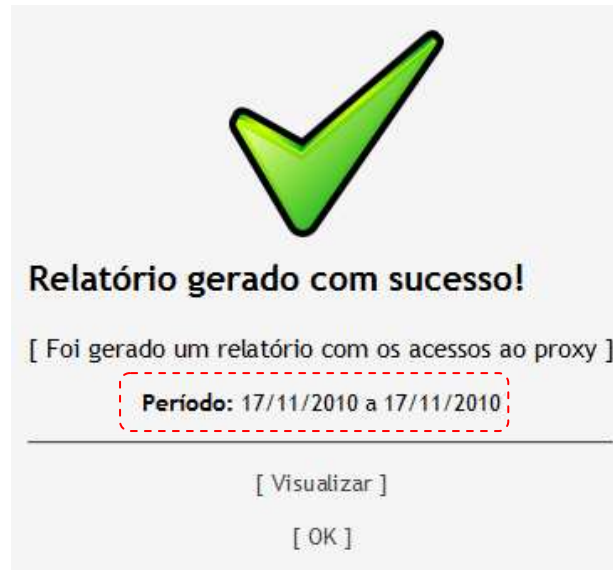
Figura 42 - Relatório com os acessos de um mês ao proxy

ARQUIVO/PERÍODO	DATA CRIAÇÃO	USUÁRIOS	BYTES	MÉDIA
2010Oct08-2010Nov05	Sex Nov 5 19:42:36 BRST 2010	5	22.01M	4.40M
2010Oct08-2010Oct22	Sex Out 22 16:51:56 BRST 2010	5	15.86M	3.17M
2010Oct08-2010Oct14	Sex Out 22 15:34:14 BRST 2010	5	11.88M	2.37M
2010Oct08-2010Oct12	Qui Out 14 08:35:04 BRT 2010	5	11.77M	2.35M
17Nov2010-17Nov2010	Wed Nov 17 20:32:37 BRST 2010	1	704.08K	704.08K
10Nov2010-10Nov2010	Qua Nov 10 23:55:51 BRST 2010	1	5.44M	5.44M
05Nov2010-05Nov2010	Qua Nov 10 21:06:38 BRST 2010	2	7.02M	3.51M
08Oct2010-08Oct2010	Seg Nov 8 17:43:51 BRST 2010	3	2.55M	850.26K
08Oct2010-17Nov2010	Qui Nov 25 21:50:37 BRST 2010	6	45.16M	7.52M
08Oct2010-05Nov2010	Qua Nov 10 21:04:39 BRST 2010	6	26.34M	4.39M

Figura 43 - Visualização de relatórios gerados no diretório Monthly

04 – Opção de relatório customizado, onde o usuário informa o período do relatório a ser gerado pelo programa SARG. Este formulário possui tratamento de erros com relação a campos obrigatórios e formatação de data.

Figura 44 - Tratamento de erros formulário de relatório customizado



**Figura 45** - Relatório com os acessos num período informado pelo usuário

### 6.8.2 Relatórios de políticas SquidHelp

Nesta seção, será possibilitado ao usuário a emissão de relatórios sobre os cadastros efetuados junto a aplicação web SquidHelp. São oferecidas algumas opções de filtro, gerando documentos no formato pdf.



**Figura 46** - Relatórios de políticas SquidHelp

### 6.8.2.1 Descrição da interface

01 – Relação de sites e domínios URL, com filtro para a situação de liberados, bloqueados ou todos.



Figura 47 - Relação de sites/domínios URL

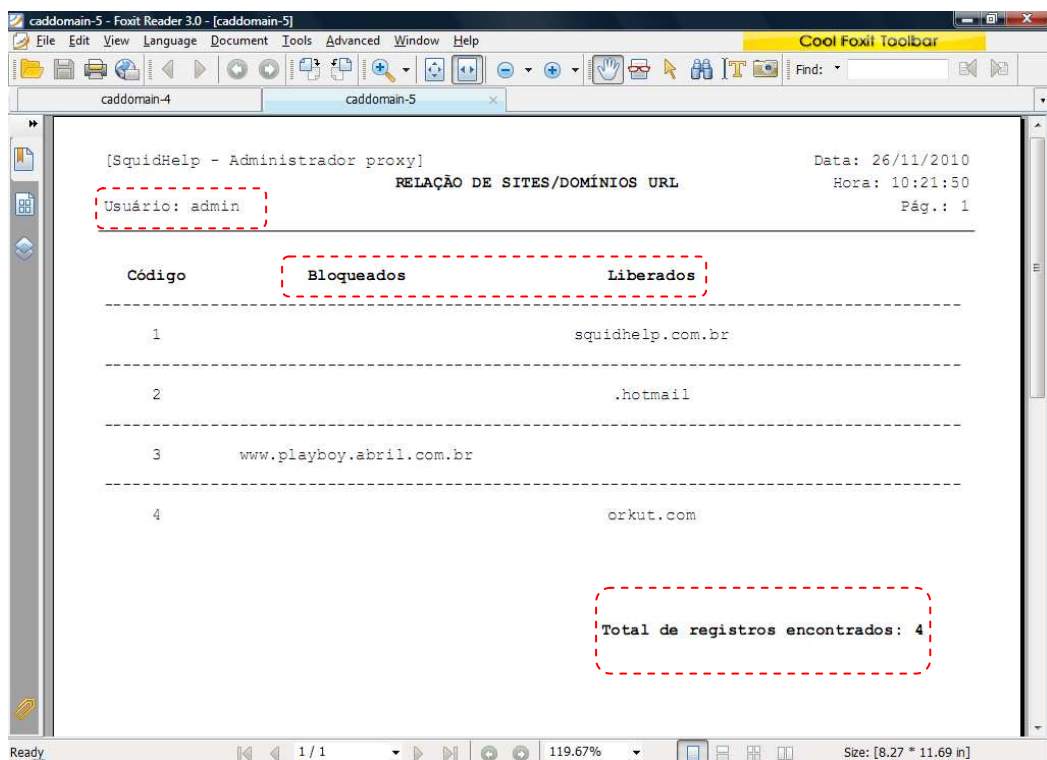


Figura 48 - Relatório gerado em pdf dos sites/domínios URL contido no banco de dados



02 – Relação de endereços IP, com filtro para a situação de liberados, bloqueados ou todos.



Figura 49 - Relação de endereços IP

Código	Bloqueados	Liberados
6	200.234.21.23	
7		10.58.115.249
8	192.168.2.99	
9	192.168.2.100	
10	192.168.2.101	

Total de registros encontrados: 5

Figura 50 - Relatório gerado em pdf dos endereços IP contido no banco de dados

03 – Relação de endereços MAC, com filtro para a situação de liberados, bloqueados ou todos.



Figura 51 - Relação de endereços MAC

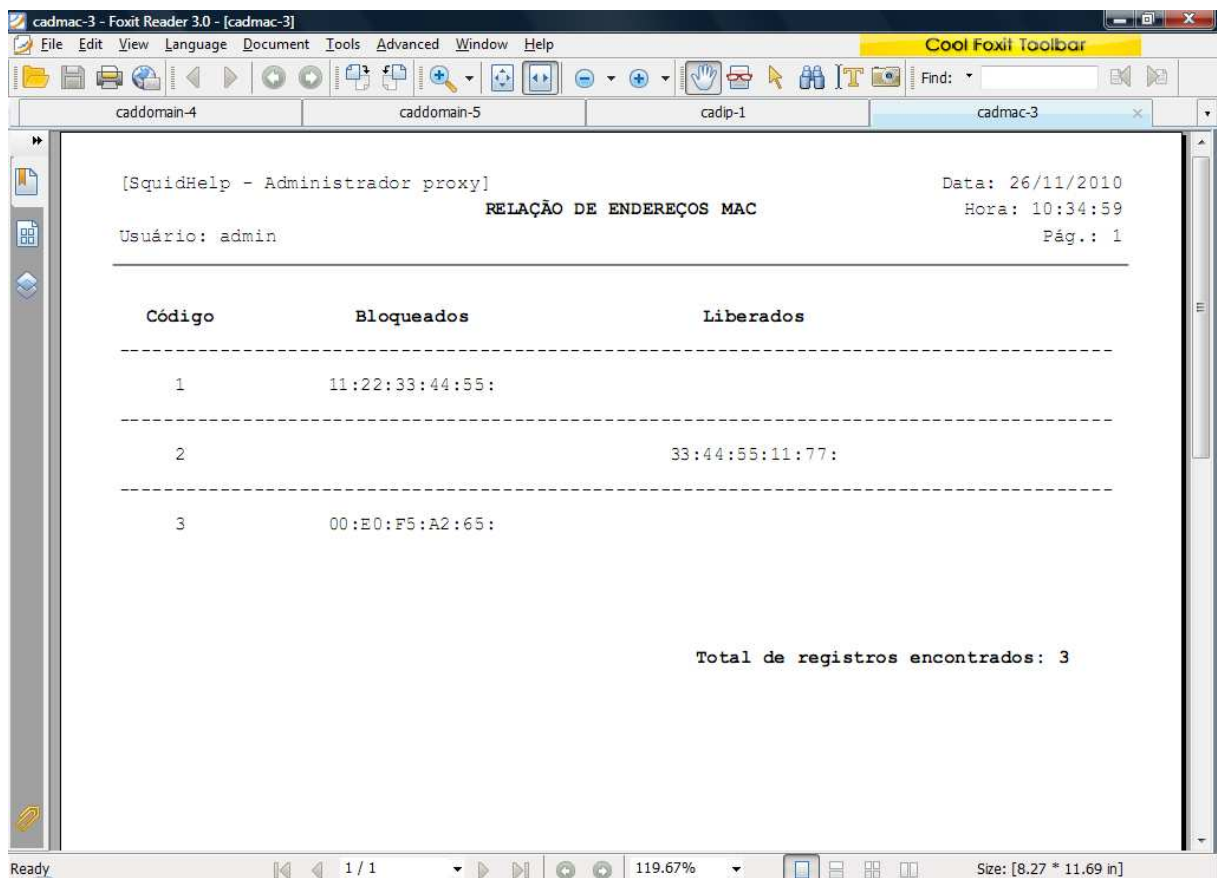


Figura 52 - Relatório gerado em pdf dos endereços MAC contido no banco de dados

04 – Relação de palavras proibidas cadastradas na aplicação SquidHelp.



Figura 53 - Relação de palavras proibidas

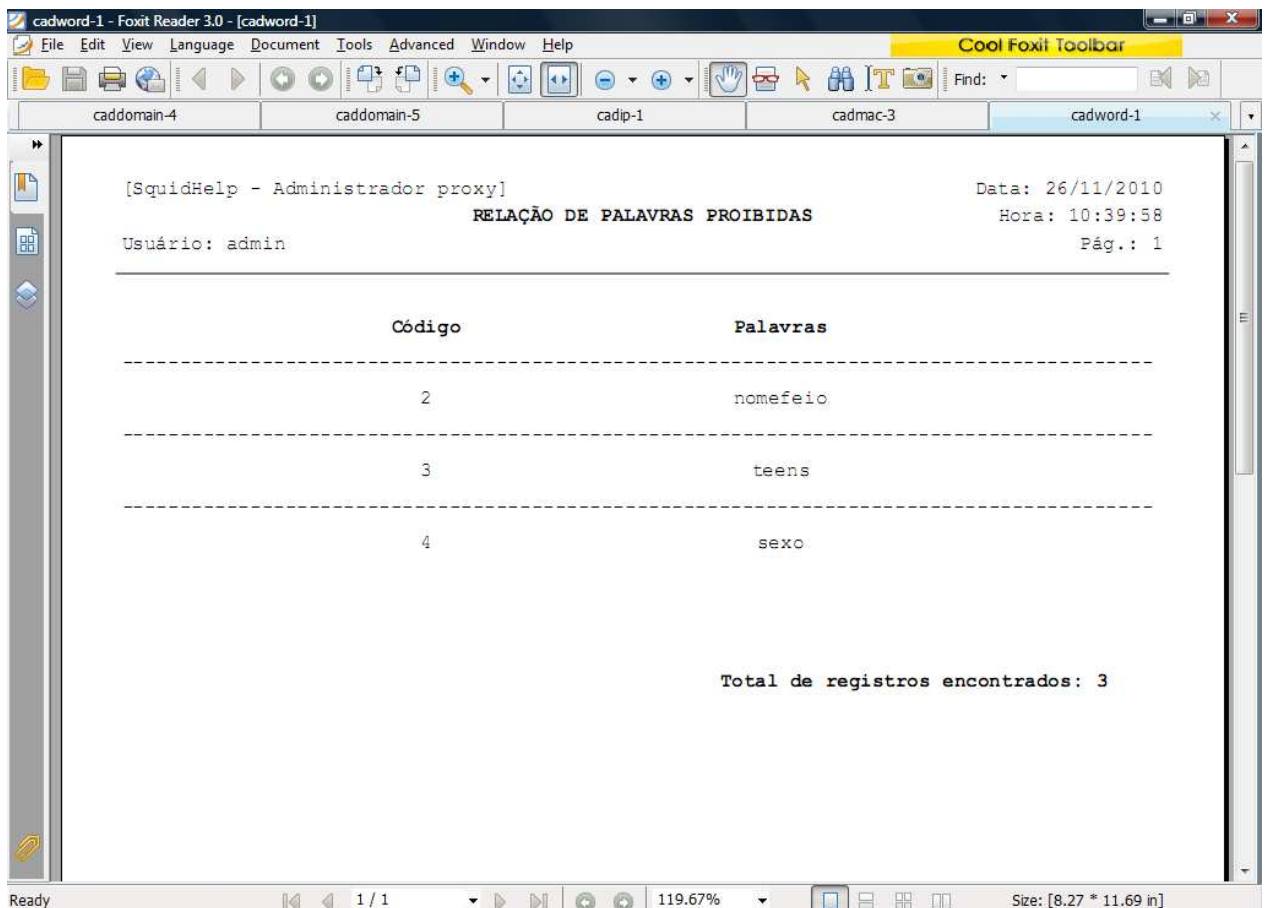


Figura 54 - Relatório gerado em pdf das palavras proibidas contido no banco de dados

05 – Relação de extensões de arquivos, com filtro para a situação de liberados, bloqueados ou todos.



Figura 55 - Relação de extensões de arquivos

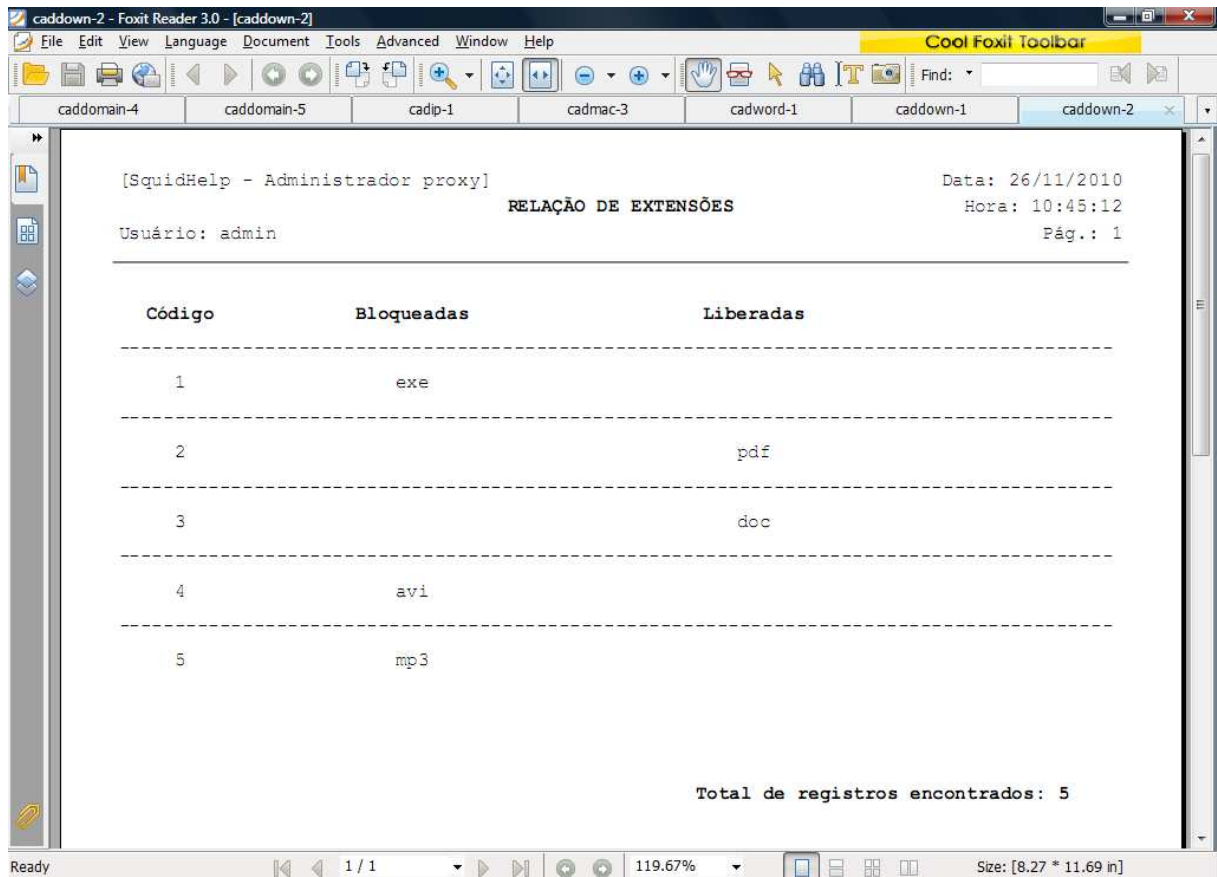


Figura 56 - Relatório gerado em pdf das extensões de arquivos contido no banco de dados

06 – Relação de usuários, com filtro para a situação de ativos, grupo e utilizadores da aplicação SquidHelp.



Figura 57 - Relação de usuários

Código	Nome do usuário	Login	Ativo	Squidhelp?	Grupo
5	Administrador SQH	admin	SIM	SIM	TOTAL
28	Cezar Luiz Krause	krause	SIM	SIM	TOTAL
29	Denise Michels	denise	NÃO	SIM	TOTAL
30	Trabalho de conclusao	tec123	SIM	NÃO	RESTRITO

Total de registros encontrados: 4

Figura 58 - Relatório gerado em pdf dos usuários contido no banco de dados

## 6.9 SOBRE

Alerta criado para identificar o autor do programa e a instituição de ensino.

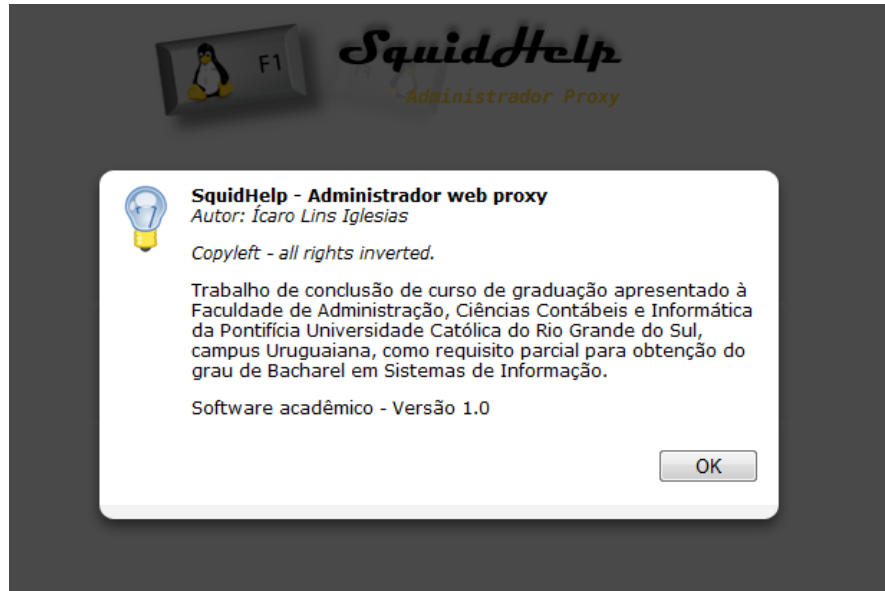


Figura 59 - Alerta Sobre

## 6.10 SAIR

Alerta criado para abandonar a aplicação SquidHelp, com confirmação ao usuário.



Figura 60 - Alerta Sair

## 7 Conclusão

Este trabalho teve o objetivo de demonstrar o desenvolvimento de uma aplicação *web* voltada para auxiliar a administração do servidor *proxy* Squid.

Durante o desenvolvimento do projeto surgiram diversas dificuldades, com destaque para as telas de manutenção de contas de usuário, relacionado à edição de usuários que não desejem alterar sua senha e geração de relatórios customizados através do programa SARG, onde foi necessário realizar alguns ajustes nas permissões do usuário do servidor Apache (*www-data*) junto ao sistema GNU/Linux Ubuntu Server 8.04. Outra dificuldade encontrada foi a manipulação dos arquivos utilizados pela aplicação: usuários, regras e *logs*.

A aplicação *web* SquidHelp foi desenvolvida atentando para as heurísticas de IHC, utilizando ícones grandes e atrativos, rótulos e formulários com tamanho padronizado em todas as interfaces, tratamento de erros com *feedback* ao usuário, exemplos de formatação aceita para que o servidor *proxy* consiga entendê-las, além da combinação de cores escolhida que tornou o sistema visualmente atrativo.

A usabilidade também foi uma das preocupações consideradas durante o desenvolvimento do projeto, onde foi desenvolvido um *menu* para acesso rápido às opções do sistema, estando este presente em todas as telas, com uma breve explicação sobre cada uma de suas funcionalidades.

Nos testes realizados em laboratório, o sistema obteve resultados satisfatórios, mostrando ser um eficiente sistema, atendendo aos objetivos propostos.

Por fim, conclui-se que a aplicação *web* SquidHelp pode ser uma ferramenta de grande serventia para usuários que não possuam maiores conhecimentos sobre sistemas GNU/Linux e conseqüentemente seus comandos em modo texto, sendo uma opção ideal para controlar de forma eficaz os acessos à internet.

### 7.1 TRABALHOS RELACIONADOS

Para o desenvolvimento deste trabalho foi realizado uma pesquisa na internet sobre temas relacionados, onde se obteve como resultado o trabalho de conclusão de curso do bacharel Vanderson Clayton Siewert, graduado em Ciências da

Computação pela Universidade Regional de Blumenau - Santa Catarina, no ano de 2007, com o título: *Ferramenta Web para Administração do Servidor Proxy Squid*. Também foi utilizado como referência o trabalho de Pós-graduação em Segurança de Redes de computadores, sob a autoria de Rogério Pereira Gualberto e Rommel Baia Silva, da Faculdade Salesiana de Vitória, também do ano de 2007, intitulado *Proposta de Controle Eficaz do Acesso à Internet*.

Os dois trabalhos utilizam o Squid como servidor *proxy*, sendo que a aplicação *web* desenvolvida difere-se devido ao uso de banco de dados, possibilitando relatórios sobre os cadastros efetuados, enquanto a aplicação citada utiliza um controle baseado apenas em arquivos.

## 7.2 TRABALHOS FUTUROS

Como sugestões para continuidade do projeto, destacam-se trabalhos futuros como:

- Possibilidade de ser criado e editado o arquivo de configuração `squid.conf` via aplicação *web*;
- Possibilidade de utilização das demais configurações do *proxy* Squid;
- Possibilidade de escolha de configuração entre *proxy* transparente ou autenticado;
- Permitir criar outros grupos de privilégios de acesso, adaptando as necessidades dos usuários;
- Criação de perfis de acesso à aplicação, citando como exemplo um acesso no nível de usuário com possibilidade de alteração em sua senha.



## REFERÊNCIAS

- ALVAREZ, Miguel A. **Introdução a Javascript.**, [2004]. Disponível em: <http://www.criarweb.com/artigos/156.php>>. Acesso em 24 nov. 2010.
- BR. Cert. **Sobre o CERT.br.**, ([2010?]). Disponível em <<http://www.cert.br/>> Acesso em 24 nov. 2010.
- CARMONA, Tadeu. **Universidade Linux.** São Paulo: Digerati Books, 2007. 383 p.
- CONTEÚDO aberto. **In: Wikipédia: a enciclopédia livre: HTML.** Disponível em: <<http://pt.wikipedia.org/wiki/HTML>> Acesso em: 24 nov. 2010.
- CONTEÚDO aberto. **In: Wikipédia: a enciclopédia livre: Proxy.** Disponível em: <<http://pt.wikipedia.org/wiki/proxy>> Acesso em: 24 nov. 2010.
- CONTEÚDO aberto. **In: Wikipédia: a enciclopédia livre: Squid.** Disponível em: <<http://pt.wikipedia.org/wiki/squid>> Acesso em: 24 nov. 2010.
- FERREIRA, Elcio. **Ajax para quem só ouviu falar.**, ([2010?]). Disponível em: <http://tableless.com.br/artigos/ajaxdemo/> Acesso em 24 nov. 2010.
- FURMANKIEWICZ, E.; FIGUEIREDO, J. **Segurança máxima – o guia de um hacker para proteger seu site na internet e sua rede.** Rio de Janeiro: Campus, 2000.
- MAUJOR. **O que é CSS.**, ([2010?]). Disponível em: <<http://maujor.com/>>. Acesso em 24 nov. 2010.
- MORIMOTO, Carlos E. **Servidores Linux, guia prático.** Porto Alegre: Sul Editores, 2009. 735 p.
- NIEDERAUER, Juliano. **Web interativa com Ajax e PHP.** São Paulo: Novatec editora, 2007. 288p.
- SOARES, Wallace. **AJAX - Guia Prático para Windows.** São Paulo: Érica, 2006. 240p.
- ORSO, P. **SARG: Squid Analysis Report Generator. [S.I.]**, ([2010?]). Disponível em: <<http://sarg.sourceforge.net>>. Acesso em: 24 nov. 2010.
- VENETIANER, Tomas. **HTML: desmistificando a linguagem da Internet.** São Paulo: Makron Books, 1996. 289p.
- VESPERMAN, J. **Autenticação e o squid. [S.I.]**, [2001]. Disponível em: <<http://br.geocities.com/cesarakg/AuthenticationAndSquid.html>>. Acesso em: 24 nov. 2010.
- WELLING, Luke; THOMSON, Laura; **PHP e MySQL, Desenvolvimento Web.** Rio de Janeiro: Elsevier, 2005. 712p

## GLOSSÁRIO

*Cache* – local onde os arquivos requisitados pelo servidor *proxy* são armazenados e repassados posteriormente para os clientes, que são as estações de trabalho da rede interna.

*Framework* – framework é um conjunto de classes, uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica.

*Gateway* – máquina intermediária geralmente destinada a interligar redes, separar domínios de colisão, ou mesmo traduzir protocolos.

*Hash* – *hash* é uma sequência de bits geradas por um algoritmo de dispersão, em geral representada em base hexadecimal, que permite a visualização em letras e números (0 a 9 e A a F), representando 1/2 byte cada.

*Hint* – opção de colocar dicas de tela, descrição sobre o link.

*Firewall* – nome dado ao dispositivo de uma rede de computadores que tem por função regular o tráfego entre redes distintas e impedir a transmissão e recepção de dados nocivos ou não autorizados de uma rede a outra.

*Kernel* – núcleo do sistema, responsável pela execução dos programas, escalonamento das tarefas e acesso aos recursos de hardware

*Log* – arquivo que guarda informações passadas (registros). Existem registros/logs de erros, páginas visitadas ("Histórico"), entre outros, que podem ajudar na identificação e solução de problemas.

*Login* – identificação para acesso em sistema computacional restrito.

*Minix* – sistema operacional Unix-like (semelhante ao UNIX), gratuito e com o código fonte disponível.

*Open source* – software que respeita as quatro liberdades definidas pela FSF (Free Software Foundation).

*Proxy* – software que centraliza o acesso de todas as estações de uma rede à Internet, podendo assim realizar várias tarefas como: armazenar localmente cópias das páginas mais acessadas para economizar banda; normatizar o acesso por horário, usuário e tipo de página acessada, entre outras possibilidades.

*Script* – linguagens de programação executada do interior de programas e/ou de outras linguagens de programação, não se restringindo a esses ambientes. As linguagens de *script* servem para estender a funcionalidade de um programa e/ou controlá-lo.

*Sexyalertbox* – script desenvolvido em JavaScript, utilizando o *framework* Mootools, onde levamos um nível acima as nossas janelas de alert(), tornando essa janela mais amigável ao usuário.

*Spry* – frameworks para Ajax baseado em uma biblioteca JavaScript. Proporciona aos *webdesigners* a habilidade de desenvolver sites e aplicativos mais ricos e interativos.

*Tags* – estruturas de linguagem de marcação que consistem em breves instruções, tendo uma marca de início e outra de fim.

*Unix-like* – sistema parecido com o Unix, não estando necessariamente de acordo com sua especificação.