

MARLON FRANCISCO DOS SANTOS

**IMPLEMENTAÇÃO DE PAINEL DE ADMINISTRAÇÃO PARA
GERENCIAMENTO DO *SQUID***

**FACULDADES UNIFICADAS DE TEÓFILO OTONI
TEÓFILO OTONI - MG**

2017

MARLON FRANCISCO DOS SANTOS

**IMPLEMENTAÇÃO DE PAINEL DE ADMINISTRAÇÃO PARA
GERENCIAMENTO DO SQUID**

Monografia apresentada ao Curso de Sistemas de Informação das Faculdades Unificadas de Teófilo Otoni, como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.

Área de Concentração: Desenvolvimento Web.
Orientador: Prof. Luiz Fernando Alves Souza.

**FACULDADES UNIFICADAS DE TEÓFILO OTONI
TEÓFILO OTONI – MG**

2017

FOLHA DE APROVAÇÃO

A monografia intitulada: *Implementação de painel de administração para gerenciamento do squid,*

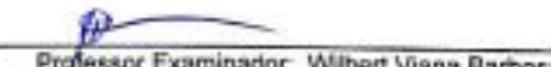
elaborada pelo aluno Marlon Francisco dos Santos,

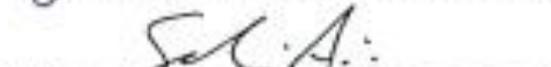
foi aprovada por todos os membros da Banca Examinadora e aceita pelo curso de Sistemas de Informação das Faculdades Unificadas de Teófilo Otoni, como requisito parcial da obtenção do título de

BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Teófilo Otoni, 20 de novembro de 2017


Professor Orientador: Luiz Fernando Alves Souza


Professor Examinador: Wilbert Viana Barbosa


Professor Examinador: Salim Ziad Pereira Aouar

ABREVIATURAS E SIGLAS

AD - Active Directory

BD - Banco de dados

BIND - Berkeley Internet Name Domain (Nome de domínio de internet)

CIFS - Common Internet File System (Sistema comum de arquivos da Internet)

CSS - Cascading Style Sheets (Folha de estilo em cascata)

DC – Domain Controller (Controlador de domínio)

DHCP - Dynamic Host Configuration Protocol (protocolo de configuração dinâmica de host)

DNS - Domain Name System (Sistema de nomes de domínios)

DOS - Disk Operating System (Sistema operacional em disco)

FAQ - Frequently Asked Questions, (Perguntas mais frequentes)

Fstab - File systems table (Tabela de sistemas de arquivos)

FTP - File Transfer Protocol (Protocolo de Transferência de Arquivos)

GIT - Controlador de versão

GNU – Licença pública geral (General Public License)

HTTP - Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto)

IP - Internet Protocol (Protocolo de internet)

LAN - Local area network (Rede de área local)

LDAP - Lightweight Directory Access Protocol (Protocolo de acesso aos diretórios leves)

MVC - Model View Controller (Modelo Visão Controlador)

NetBIOS - Network Basic Input Output System (Saída de entrada básica de sistema de rede)

NFS - Network File System (Sistema de Arquivo de Rede)

NMBD - Database Network Neighborhood Database (Banco de dados do endereço de rede)

NTP - Network Time Protocol (Protocolo para sincronização dos relógios)

ORM - Object Relational Mapping (Mapeamento Objeto Relacional)

PHP - Personal Home Page

SMB - Server Message Block (Servidor de Bloco de mensagens)

SMTP - Simple Mail Transfer Protocol (Protocolo de transferência de correio simples)

SO - Sistema Operacional

SQL - Structured Query Language (Linguagem de Consulta Estruturada)

SSL - Secure Sockets Layer

Swat - Samba Web Administration Tool (Ferramenta de administração da Web Samba)

WAN - Wide Area Network (Rede de longa distância)

WINS - Windows Internet Name Serviço (Serviço de Nome da Internet do Windows)

RESUMO

O presente trabalho de conclusão de curso tem como objetivo desenvolver um painel para plataforma web, utilizando o framework Laravel, para o gerenciamento do AD (Active Directory) e do Squid. Será elaborado de forma simples, utilizando de software livre, possibilitando desta forma a redução de custo, permitindo assim realizar ações básicas no Active Directory tais como, criar grupos, usuários, redefinir senha e deleta-los. Possibilitando o gerenciamento do Squid, definindo quais sites cada grupo ou usuário pode acessar com nível de acesso. Cada usuário utilizará o mesmo login e senha criado no AD, para autenticar no momento de acessar a internet. Em busca de criar uma aplicação ágil e eficiente, permitindo assim que os gestores de informática encontrem mais tempo para outras atividades e contribuir de forma produtiva para o ambiente organizacional corporativo.

Palavras-Chave: Active Directory; Squid; Proxy; Samba; GNU/Linux; Framework Laravel.

SUMÁRIO

INTRODUÇÃO	9
1. REFERENCIAL TEÓRICO	11
1.1 Samba.....	11
1.1.1 História do Samba.....	12
1.1.2 Versões	13
1.1.3 Funcionalidades	15
1.2 Active Directory	18
1.3 História Do Squid	20
1.3.1 Squid	21
1.3.2 Funcionamento.....	22
1.4 Sarg	23
1.5 Banco De Dados	24
1.6 Linguagem De Marcação – HTML	25
1.7 Cascading Style Sheets - CSS	26
1.8 Personal Home Page - PHP	26
1.9 Framework	27
1.9.1 Laravel.....	27
1.9.1.1 MVC (Model View Controller)	28
1.9.1.2 <i>Eloquent ORM</i>	29
1.9.1.3 <i>Composer</i>	29

1.9.2 Bootstrap	30
1.10 JavaScript	30
1.11 Sistemas Operacionais	31
1.11.1 GNU/Linux	31
2. DESENVOLVIMENTO	32
2.1 Configuração do Servidor	32
2.1.1 Configuração de Rede.....	33
2.1.2 Configuração do DHCP	34
2.1.3 Instalando o Samba 4.....	35
2.1.4 Configuração do Squid.....	38
2.1.5 Desenvolvimento da ferramenta.....	43
CONSIDERAÇÕES FINAIS	47
REFERÊNCIAS.....	49

INTRODUÇÃO

A presente monografia foi realizada com intuito da obtenção do título de bacharel em Sistemas de Informação do curso das Faculdades Unificadas de Teófilo Otoni - MG com o tema "implementação de painel de administração para gerenciamento do Squid". Concentra-se na área do desenvolvimento web e redes, na qual aborda o gerenciamento, configuração de servidores e aplicação web.

O projeto tem como foco realizar a instalação e configuração do Samba 4, como controlador de domínio e definir o nível de acesso à internet com autenticação através do Squid. Utilizando de ferramentas Open Source (Código aberto), visando um custo baixo sem a necessidade de licenciamento. A aplicação realizará funções básicas como controlador de domínio, criar usuário, redefinir senha, desativar, mover de grupo e excluir, criar ou excluir grupos. No Squid é possível definir regras gerais para todos os grupos e usuários, liberado ou bloqueado. As regras específicas serão aplicadas para todos no grupo, possibilitando também definir regras em particular para um usuário, sobrescrevendo as demais regras. Para realizar este tipo de nível de regras, foi necessário configurar o servidor proxy de forma não transparente, pois somente assim é possível trabalhar com autenticação, na qual vai utilizar o mesmo login e senha fornecido pelo AD, sem a necessidade de criar um novo login e senha para o acesso à internet.

Atualmente no mercado existem ferramentas que contempla esta funcionalidade infelizmente com um manuseio complexo. Em busca de uma maneira para facilitar o gerenciamento do Squid, foi desenvolvido um software capaz de realizar esta tarefa. Foram levantadas as seguintes hipóteses:

H0: Os meios realizados no ambiente corporativo seriam eficazes, não necessitando de uma nova ferramenta, pelo custo e segurança e suporte oferecido pelas ferramentas disponíveis no mercado.

H1: O servidor proxy ajudaria na velocidade da internet, pois poderia ser feito o cache no servidor no qual seriam armazenados arquivos pequenos, como páginas e imagens, que seriam entregues instantaneamente para os clientes sem precisar buscar todo o conteúdo do site novamente.

H2: Com a implementação do servidor o funcionário poderia realizar sua função sem distração em sites alheios.

H3: Com a ferramenta poderia verificar o conteúdo da internet, que o funcionário está acessando e realizando a liberação ou bloqueio do conteúdo de forma simples e rápida.

H4: A ferramenta não teria despesas com licenças sendo assim não haveria custo alto para empresa.

Este trabalho foi dividido em 2 capítulos, na qual o primeiro descreverá sobre os conceitos da aplicação e ferramentas.

O segundo capítulo descreve o desenvolvimento, detalhando o processo de instalação e configuração do software.

1. REFERENCIAL TEÓRICO

1.1 Samba

O Samba é um Software Livre licenciado sob a GNU *General Public License*, publicado em 1992. O Samba fornece serviços de compartilhamento de arquivo e impressão, de forma segura, estável e rápida, para todos os clientes que usam o protocolo SMB (*Server Message Block*) ou Bloco de Mensagens do Servidor / CIFS (*Common Internet File System*) ou Sistema Comum de Arquivos da Internet. Como todas as versões dos sistemas DOS (*Disk Operating System*) ou sistema operacional em disco e Windows, OS/2 (Sistema Operacional), GNU/Linux e muitos outros SO's. O Samba é um componente importante para integrar perfeitamente servidores GNU/Linux / Unix em ambientes *Active Directory* (AD). Outro serviço disponível é a funcionalidade de controlador de domínio (DC).¹

Filho descreve o Samba como um serviço famoso por implementar o protocolo SMB e protocolo CIFS no GNU/Linux. Esse protocolo é responsável por permitir que a máquina Windows compartilhe arquivos e impressoras, além de outras tarefas inerentes as redes Microsoft Windows. O Samba pode ser implementado em GNU/Linux e Unix em geral permitindo a interatividade com o Microsoft Windows. (FILHO, 2012, p.823)

¹ Disponível em <<https://www.samba.org/samba>>
Tradução livre do autor: Marlon Francisco dos Santos.

1.1.1 História do Samba

O Samba teve como criador Andrew Tridgell e surgiu devido a necessidade que ele encontrava em montar um espaço em disco no seu computador DOS a partir de um servidor Unix. Na verdade este não era o problema, porque ele tinha um cliente NFS (*Network File System*) um sistema de Arquivo de Rede para o DOS que funcionava bem. Porém, a aplicação precisava de suporte ao protocolo NetBIOS (*Network Basic Input Output System*) ou Sistema de Saída de Entrada Básica de rede, porém infelizmente não suportava NFS.

A solução escolhida por Andrew foi escrever um código sniffer², para aplicar a engenharia reversa do protocolo SMB e implementar no Unix. Com isso, ele criou uma comunicação entre o sistema de arquivo Windows com o seu servidor Unix, que lhe permitiu montar sistemas de arquivos compartilhados do servidor Unix ao mesmo tempo que ele executa aplicativos NetBIOS.

Andrew publicou seu código *nbserver*³ no início de 1992. Seu lançamento ocorreu de forma prematuro e com vários erros. Sendo assim, ele decidiu deixar o projeto de lado. Após um longo tempo, quase dois anos depois, decidiu religar o computador com o sistema operacional Windows de sua esposa ao seu computador com o sistema GNU/Linux. A falta de melhores opções, ele usou seu próprio código na qual continha muitas falhas e erros. Através de várias pesquisas descobriu que as documentações dos protocolos SMB e *NetBIOS* estavam atualizadas e com este novo apanhado de informações voltou a dedicar-se ao projeto.

Algum tempo depois ele foi contatado por uma empresa que reivindicava a marca registrada no nome que ele havia escolhido para o seu software. Ao invés de causar um barulho, Andrew fez uma pequena procura em um dicionário "smb" e acabou encontrando o termo "Samba" na qual mudou o nome da sua aplicação solucionado assim o problema.

² Ferramentas que interceptam e analisam o tráfego de uma rede.

³ Nome da aplicação criada pelo Andrew

O projeto Samba cresceu poderosamente e desde então não parou mais. Andrew agora tem toda uma equipe de programadores espalhada pelo mundo para ajudar no desenvolvimento do *software*. Quando uma nova versão é anunciada, milhares de cópias são baixadas em poucos dias. Disponível de forma *OpenSource* para qualquer pessoa que poderá utilizar, entender e modificar o código, sobre licença GNU.⁴

1.1.2 Versões

A primeira versão foi escrita no ano de 1992. Foi um dos primeiros programas de *sockets*, a conversar com o NT4 (Edição do sistema operacional Microsoft Windows NT 4.0) para obter uma listagem de diretórios fora dele.

Em meados de 1993 surgiu a versão *Nbserver* 1.5, nela foi incluído o "*nmbserver*" agora conhecido como *nmbd* (*Database Network Neighborhood Database*, em português banco de dados do endereço de rede), *smbclient* (*Server Message Block* para cliente, em português Bloco de Mensagem de Servidor em clientes) é a renomeação do domínio primário para "*smbserver*". Esta foi a primeira versão que foi influenciada por uma especificação SMB. Também teve a capacidade de atuar como um servidor de impressão.

Na versão *Nbserver* 1.5.20 teve o primeiro patch (Atualização para correção de erros) disponibilizados pelo Lee Fisher, funcionário da Microsoft à depuração do *SMBcreate* na qual foi financiada é paga pelo seu próprio bolso. Após os ajustes conseguiu conectar os servidores NT graças as correções.

Com o lançamento da versão 1.5.21, o arquivo "*smb.conf*" nasceu devido ao árduo trabalho de Karl Auer. Isso na época foi considerado bastante revolucionário para o projeto em sua grande parte. Com isso, a estrutura do código facilitou a criação de novas opções de configurações e do tempo de execução.

⁴ Disponível em <<https://www.samba.org/samba/docs/SambaIntro.html>>
Tradução livre do autor: Marlon Francisco dos Santos.

A versão 1.6 lançada em março de 1994, oferecia suporte de nomes longos em arquivos graças a Jeremy, além de uma melhor documentação devido ao trabalho de Karl.

Em abril de 1994, o nome do pacote foi alterado para "Samba", depois de ter recebido uma carta de um gerente da empresa Syntax Corp, dizendo que eles tinham uma marca registrada no nome "smbserver". Sendo assim, Andrew teve que alterar o nome para Samba e foi lançada a versão 1.6.05.

A versão 1.6.09 foi a mais popular dentre as versões anteriores, na qual teve diversos downloads. Com isso, chegavam diversos feedbacks de seus usuários. Esta também foi a primeira versão a suportar a segurança no nível de usuário, o que o tornou um concorrente muito mais sério para uso corporativo.

O Samba 1.7 foi lançado em julho de 1994, versão que iniciou suporte para o uso de nomes longos e corrigiria uma grande quantidade de bugs (Falha e erro no *software*). Também foi a primeira versão que foi feita o pré-lançamento *alpha*. Ao invés de simplesmente despejar o que estava no seu diretório pessoal em um site FTP (*File Transfer Protocol* em português, Protocolo de Transferência de Arquivos) disponibilizava a cada poucos dias. Sendo também a primeira versão a ter um FAQ (*Frequently Asked Questions*, que pode ser traduzido por Perguntas Mais Frequentes), graças novamente ao maravilhoso trabalho de Karl Auer.

O Samba 1.8 obteve uma nova estrutura de arquivos e de código fonte. Essa mudança trivial provocou uma grande melhoria no desenvolvimento. Esta versão também foi a primeira a adicionar suporte primitivo para idiomas estrangeiros.

Em janeiro de 1994, com o lançamento do Samba 1.9, ele trouxe consigo algum suporte de navegação decente e até permitiu que o Samba fosse o navegador principal de uma rede. Pequenos aparelhos começaram a surgir usando Samba como GNU/Linux ou FreeBSD fornecendo o armazenamento instantâneo para as redes Windows.

Na versão 2.0 lançada em janeiro de 1999, teve inúmeros downloads de diversas partes do mundo. John Blair lançou o primeiro livro sobre Samba. Este foi um enorme marco e tornou o Samba muito mais acessível aos usuários comuns. Foi uma grande surpresa para todos, quando este livro superou os gráficos de vendas por

vários meses, o que levou imediatamente dezenas de outras editoras a anunciar o livro do Samba. Atualmente existe mais de 20 livros em várias linguagens possibilitando um maior acesso aos interessados no assunto.

Com a versão do Samba 2.2 já é possível fazer um controlador de domínio (DC) primário para domínios NT4, suportando Windows 2000 e uma infraestrutura de impressão muito melhorada graças ao suporte da HP entre outras coisas.

O código cresceu em uma escala de fator 100 vezes mais, desde do primeiro lançamento em 1992, O samba 3.0 suporta o *Active Directory* (AD) e tem muitas outras melhorias estruturais que proporcionam uma base muito mais sólida para os próximos anos.⁵

Na versão 4.0 foi introduzido suporte para a tecnologia *Microsoft Active Directory* através da implementação e combinação de serviço LDAP (*Lightweight Directory Access Protocol* - Protocolo de acesso aos diretórios), e serviço de autenticação *Heimdal Kerberos*, serviço DNS (*Domain Name System*, traduzido sistema de nomes de domínios). Ele é capaz de realizar todos procedimento de um controlador de domínio do *Active Directory* disponível pela Microsoft.

1.1.3 Funcionalidades

O Samba consiste em dois programas-chave, *smbd* (serviços de compartilhamento de arquivos e impressão para clientes do Windows) e *nmbd* (serviço de nomes de *NetBIOS* sobre IP). Esses serviços oferecem as seguintes funcionalidades para implantar os serviços do samba:

- Serviços de arquivo e impressão;
- Autenticação e autorização;
- Resolução de nomes;

⁵ Disponível em <<https://www.samba.org/samba/docs/10years.html>>
Tradução livre do autor: Marlon Francisco dos Santos.

- Anúncio de serviço (navegação);

Estas funções são fornecidas pelo `smbd` Daemon `smb`. `Smbd` é lida como autenticação e autorização de "modo compartilhamento" e "modo usuário". Ou seja, pode-se proteger serviços ou arquivos e impressoras compartilhadas, exigindo senhas. No modo compartilhamento, o esquema mais simples e menos recomendado, uma senha pode ser atribuída a um diretório ou impressora compartilhada simplesmente chamado de "compartilhamento". Esta senha única é então disponibilizada a todos que estão autorizados a usar o compartilhamento.

Com a autenticação do modo de usuário, cada usuário possui seu próprio nome de usuário e senha, e o administrador do sistema pode conceder ou negar o acesso individualmente.

O sistema de domínio do Windows NT fornece um nível adicional de refinamento de autenticação. A ideia básica é o usuário ter acesso fazendo o login uma única vez para ter acesso a todos os serviços autorizados na rede. O sistema NT *Domain* lida com um servidor de autenticação chamado de controlador de domínio, sendo NT um grupo de máquinas que partilham o mesmo controlador de domínio.

O sistema NT Domain merece um reconhecimento especial porque até o lançamento do Samba versão 2, apenas o código de propriedade da Microsoft conseguia implementar os protocolos de autenticação do domínio NT. Com a versão 2, o Samba introduziu o primeiro código de autenticação NT Domain não derivado da Microsoft. O objetivo final é imitar completamente um controlador de domínio do Windows NT.

Esses dois serviços envolvem basicamente o gerenciamento e a distribuição de listas de nomes NetBIOS.

A resolução do nome tem duas formas: (1) transmissão de ponto a ponto, onde uma máquina pode usar um ou ambos os métodos, dependendo da configuração. A resolução mais próxima ao mecanismo NetBIOS original. Um exemplo básico, presume-se que um cliente que procura um serviço chamado *Trillian* chama "Yo! Trillian! Where are you?" (Onde está você?) e aguarda que a máquina com esse nome responda com um endereço IP, isso pode gerar um pouco de tráfego de transmissão, mas é restrito à LAN local, por isso não causa muitos problemas.

O outro tipo de resolução de nome envolve a (2) utilização de um NBNS *NetBIOS* atualmente chamada de WINS (*Windows Internet Name Service*). O funcionamento do NBNS ocorre da seguinte forma: Os clientes enviam seus nomes NetBIOS e endereços IP para o servidor NBNS, que mantém as informações em um banco de dados simples. Quando um cliente quer conversar com outro cliente, ele envia o nome do outro cliente para o servidor NBNS. Se o nome estiver na lista, o NBNS entrega um endereço IP, sendo assim, ele tem o nome e procura o número.

Os clientes em diferentes sub-redes podem compartilhar o mesmo servidor NBNS. Então, ao contrário da transmissão, o mecanismo ponto-a-ponto não está limitado à LAN (*local area network* pode ser traduzido por Rede de área local). Em muitos aspectos, o NBNS é semelhante ao DNS, mas a lista de nomes NBNS é quase completamente dinâmica e existem poucos controles para assegurar que somente clientes autorizados possam registrar nomes. Os conflitos podem ocorrer com bastante facilidade.

Em uma LAN, os computadores participantes mantêm uma eleição para decidir qual deles se tornará o Navegador Mestre Local (LMB). O "vencedor" então identifica-se reivindicando um nome *NetBIOS* especial (além de outros nomes que possa ter). O trabalho do LMBs é manter uma lista de serviços disponíveis e esta é a lista que aparece no momento da consulta no Windows "*Network Neighborhood*".

Além do LMBs, existe o *Domain Master Browsers* (DMBs). Os DMBs coordenam listas de navegação em domínios NT, mesmo em redes roteadas. Usando o NBNS, um LMB localizará seu DMB para trocar e combinar listas de navegação. Assim, a lista de procura é propagada para todos os hosts no domínio NT. Infelizmente, os tempos de sincronização são separados um pouco, podendo demorar mais de uma hora para uma alteração em uma sub-rede remota para aparecer nas sub redes.

De forma geral, o Samba vem com uma variedade de utilitários, sendo os mais utilizados:

- ✓ Smbclient:

Simplesmente como uma interface semelhante à do utilitário FTP. Ele pode ser usado a partir de um sistema Unix para se conectar a um compartilhamento SMB remoto, usado para transferir arquivos, enviá-los e ainda fazer o compartilhamento de impressoras remotas.

✓ *Nmblookup*:

Pode ser usado para encontrar nomes *NetBIOS* em uma rede, pesquisar seus endereços IP e consultar uma máquina remota para a lista de nomes que a mesma máquina acredita possuir.

✓ SWAT

SWAT (*Samba Web Administration Tool*) permite configurar diversos serviços e arquivos complexos do Samba usando um navegador web de forma remota, além disso, permitindo que um administrador procure facilmente os efeitos de quaisquer alterações nas configurações. O SWAT é executado a partir de *inetd* (*Internet service daemon*, um tipo de servidor antigo disponível para sistemas operacionais UNIX e Linux).

O software utilitário para o gerenciamento do samba também foi criado por Andrew Tridgell, disponível de forma *Open Source*.^{6 7}

1.2 Active Directory

O *Active Directory* (AD) é um serviço da Microsoft disponível para Windows Server (2000, 2003, 2003R2, 2008, 2008R2, 2012 e 2012R2).⁸

Lançado em 1999, foi primeiramente disponibilizado para o Windows 2000. Após o seu nascimento assumiu a liderança dos serviços de diretório, utilizando como

⁶ Disponível em <<https://www.samba.org/samba/docs/SambaIntro.html>>

⁷ Disponível em <<https://www.samba.org/samba/docs/man/manpages-3/swat.8.html>>

Tradução livre do autor: Marlon Francisco dos Santos.

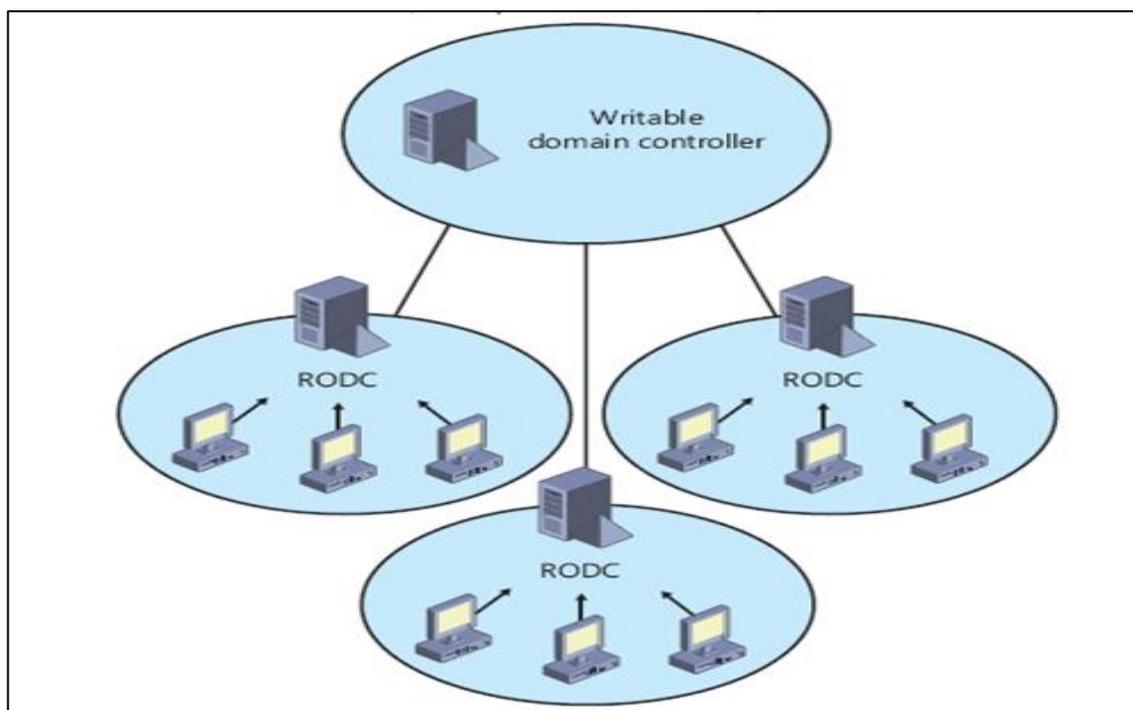
⁸ Disponível em <<http://comoaprenderwindows.com.br/wp-content/uploads/2015/04/22-Passos-Para-Instalar-o-AD-com-Sucesso.pdf>>

base o LDAP e a comunicação através de replicação. Vários atributos foram lançados e principalmente ferramentas para facilitar o gerenciamento de informações nas empresas pelo seu desempenho, segurança e principalmente disponibilidade.

A administração e o gerenciamento são feitos de forma centralizada. Existe uma estrutura que armazena informação sobre usuários e computadores da organização.

Pode-se usar para interligar domínio *Domain Controller* (DC) com matriz (Flores) e filias (Árvores) de forma que possa trocar informações, consiste no agrupamento de uma ou mais árvores de domínio. O primeiro domínio da floresta é chamado domínio raiz da floresta (mestre). O nome desse domínio faz referência as florestas. Ver o exemplo na figura 01.

Figura 01: Domain Controller (mestre)



Fonte: Disponível em <<http://microsoftgeek.com/?p=2239>>

Também é possível configurá-lo como replicação de informação em caso de pane, sendo assim, outra máquina assume o serviço como mestre, ver o exemplo na figura 02.

Figura 02: Replicação Active Directory



Fonte: Disponível em <<https://technet.microsoft.com/pt-br/library/jj206711.aspx?f=255&MSPPEror=-2147217396>>

O AD disponibiliza diversas ferramentas e recursos para o gerenciamento das estações de trabalho, tais como permissões de acesso, cota de armazenamento e outros serviços.⁹

1.3 História Do Squid

Antes do surgimento do Squid os servidores da CERN (*Conseil Européenne pour la Recherche Nucléaire* ou Conselho Europeu de Pesquisa Nuclear) funcionavam como servidor HTTP (*Hypertext Transfer Protocol* sendo em português, Protocolo de Transferência de Hipertexto) que desempenhava a função de armazenamento em cache hoje realizada pelo Squid. Na qual foi escrito por Ari Luotonen em 1994. No mesmo ano foi iniciado o projeto Harvest pela IRTF-RD (Internet Research Task Force Group on Resource Discovery), que pode ser diferenciado como um conjunto de ferramentas, para reunir, extrair, organizar, procurar, armazenar em cache e replicar as informações na Internet. O cache Harvest foi utilizado pela maioria das pessoas como mecanismo de busca ou distribuição pois tinha melhorias sobre o cache da CERN, sendo mais rápido, sistema de arquivo eficiente e hierarquias através do ICP (Internet Cache Protocol). Os principais autores do projeto Harvest foi o Peter Danzig

⁹ Disponível em <<https://technet.microsoft.com/pt-br/library/jj206711.aspx?f=255&MSPPEror=-2147217396>>

e Anawat Chankhunthod, transformado em um produto comercial. No início de 1996 o projeto foi financiado pela *National Science Foundation*, onde o projeto foi renomeado para *Squid* e liberou-o sob a GNU (*General Public License* ou Licença Pública Geral). Desde então, *Squid* cresceu em tamanho, características e agora suporta uma série de novos recursos tais como: redirecionamento de URL, modelagem de tráfego, controles de acesso sofisticados, autenticação numerosa, módulos, opções avançadas de armazenamento em disco, interceptação de HTTP entre outros. O financiamento do projeto terminou em julho de 2000. Hoje vários voluntários continuam a desenvolver e apoiar o *Squid*. Seja por apoio financeiro ou por outros tipos de suporte de empresas que se beneficiam do *Squid*. Pensando na posteridade, o *Squid* foi reescrito em linguagem C++ e ao mesmo tempo corrigindo inúmeros problemas. (WESSELS, 2004, p.36)

1.3.1 Squid

Souza, Pinto e Barbosa (2007), descrevem em seu livro o *Squid* como um software sobre licença GPL, na qual pode ser utilizado para qualquer propósito, estudado e/ou aperfeiçoado e distribuído livremente sem custo de licenciamento. O *Squid* está entre o programa mais popular do mundo, com uma ótima opção para plataformas GNU/Linux e Windows com suporte aos protocolos FTP (*File Transfer Protocol* em português Protocolo de Transferência de Arquivos), HTTP (*Hypertext Transfer Protocol* em português Protocolo de Transferência de Hipertexto), SSL (*Secure Sockets Layer*), WAIS e *Gopher*.

Assim como outros *proxy's*, possui como características: operações de filtro de sites, gerar logs de acesso, tradução de endereços NAT e implementação de mecanismo de autenticações que atua na camada 7 do modelo OSI¹⁰ (*Open Systems Interconnect*, tradução Interconexão de sistemas abertos), com o *proxy* para um ou mais serviços, podendo ser utilizado com os serviços: TELNET, FTP, SMTP (*Simple Mail Transfer Protocol* em português "Protocolo de transferência de correio simples"),

¹⁰ Modelo padrão para protocolos de comunicação entre diversos tipos de sistema, criada pela ISO (Organization for Standardization, em português Organização Internacional para Padronização)

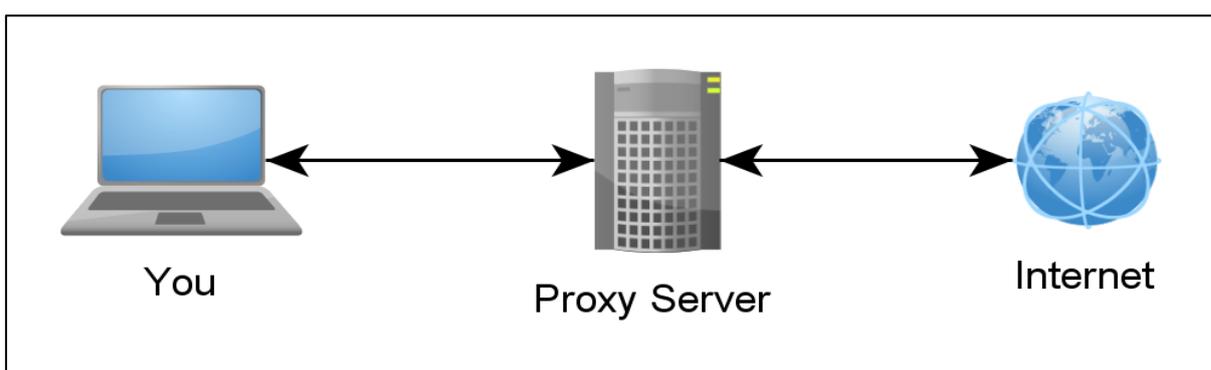
DNS (*Domain Name System* ou sistema de nomes de domínios), HTTP e também outros, utilizando como um *firewall*.

O *Squid* está sendo cada vez mais usado em arquiteturas de entrega de conteúdo, para entregar vídeo e áudio estático e de transmissão de dados para usuários de *Internet* em todo o mundo. (SOUZA; PINTO; BARBOSA, 2017, p.5)

1.3.2 Funcionamento

O Squid permite compartilhar a conexão entre vários computadores, servindo como um intermediário entre eles e a Internet. O servidor apenas repassa as requisições recebidas, como um garoto de recados. O *proxy* é como uma ponte que não se limita apenas 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, tais como *log* e *cache*. Ver na figura 03 o funcionamento do proxy.

Figura 03: Funcionamento do Proxy



Fonte: Disponível em <<https://www.tech.com.pk/how-to-fix-unable-to-connect-to-proxy-server-error/>>

Para usar os recursos do *proxy* é preciso configurar a máquina que vai prover o serviço, os navegadores e aplicativo que forem acessar a Internet em cada computador da rede. Tal configuração é denominada como *proxy* não transparente pois é feita de forma manual. Esta forma de configuração é uma tarefa trabalhosa pois

toma-se tempo aplicando a configuração do *proxy* individualmente em cada estação de trabalho.

Outra forma de fazer o *proxy* é de maneira transparente. Este recurso é muito útil para evitar que os usuários burlem o *proxy* removendo as configurações sem a necessidade de configurar cada estação de trabalho. Estas configurações são feitas automaticamente de forma transparente. Outro recurso do *proxy* é a opção de autenticação por usuário na qual é criado um usuário e senha para cada um utilizar no momento de acessar a *Internet* sendo assim possibilitando fazer regras de bloqueio por usuário ou grupo. A autenticação também pode ser feita através da integração com AD que esteja utilizando na rede.

O *Squid* permite bloquear sites indesejados de forma bastante simples. Ele permite que crie *acl's* (*Access Control List* ou Lista de controle de Acesso) contendo endereços de sites, palavras, ip's e usuários que devem ser bloqueados ou permitidos. (MORIMOTO, 2008, p.37)

Marcelo (2005, p.09) afirma que:

o arquivo *squid.conf* é o responsável por todas as configurações. Ali dentro é que serão criadas as listas de acesso *acl's* é onde poderemos inserir/modificar parâmetros importantes no sistema tais com porta de funcionamento, diretório de log e cache e etc.

1.4 Sarg

Um interpretador de *logs* para o *Squid* disponível de forma *Open Soure*, na qual gera relatórios HTML com informações. Sempre que é executado ele cria um conjunto de páginas, divididas por dia ou mês com uma lista de todas as páginas que foram acessadas e a partir de que máquina da rede veio cada acesso podendo ver qual usuário, IP, hora e data. Também é possível ver gráfico com o uso de banda e ranking de sites mais acessados. (MORIMOTO, 2008, p.s/n)

1.5 Banco De Dados

O banco de dados surgiu da necessidade de obter um controle mais efetivo dos dados e garantir a integridade da informação, com isso, é possível armazenar dados de maneira estruturada podendo ter um melhor controle. O BD (banco de dados) pode ser relacional ou não relacional. O BD relacional é feito de forma estruturada na qual contém diversas tabelas para diferentes informações, separando por exemplo, fornecedor e produtos, capaz de relacionar e extrair informações de várias tabelas ao mesmo tempo através de instrução SQL (*Structured Query Language*, tradução Linguagem de Consulta Estruturada) *select* (Selecionar), *update* (Atualizar), *insert* (Inserir) e *drop* (Deletar) dentre outros. O BD não relacional não contém esquemas de tabela fixa e geralmente não suportam instruções e operações de junção SQL. Em <<https://mariadb.org/about/>> define banco de dados relacional como MariaDB:

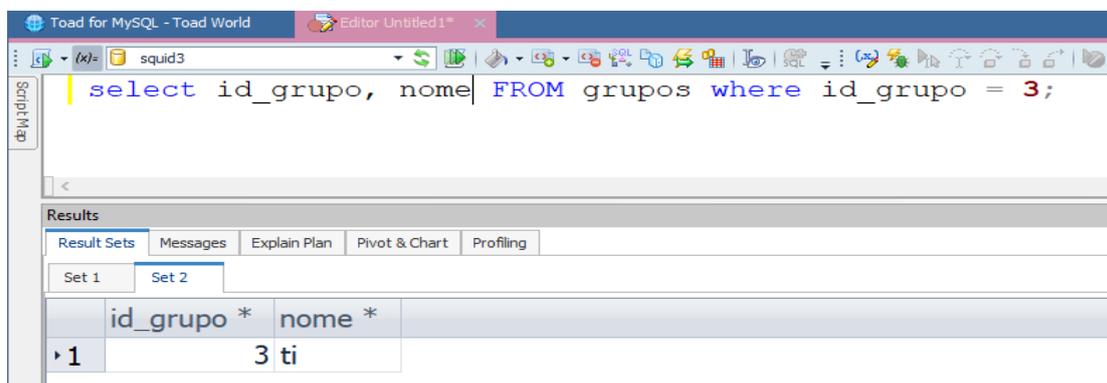
MariaDB *Server* é um dos servidores de banco de dados mais populares do mundo. Ele é feito pelos desenvolvedores originais do MySQL e garantido para ficar de fonte aberta.

MariaDB transforma dados em informações estruturadas em uma ampla gama de aplicativos, que vão desde bancos a sites. É um substituto melhorado para o MySQL. MariaDB é usada porque é rápida, escalável e robusta, com um rico ecossistema de mecanismos de armazenamento, plugins e muitas outras ferramentas que o tornam muito versátil para uma ampla variedade de casos de uso.

MariaDB é desenvolvido como software de código aberto e como um banco de dados relacional que fornece uma interface SQL para acessar dados.

Veja na figura 04 um exemplo do comando *select* para busca o grupo com a ID 3.

Figura 04: Comando SQL



Fonte: Do autor

1.6 Linguagem De Marcação – HTML

Silva (2015, p.19) fala que o HTML (*HyperText Markup Language*), significa Linguagem de Marcação de Hipertexto, sendo uma estrutura da página. Ela não é um tipo de linguagem direcionada para programação e sim para marcação de texto. Quando o cliente acessa um site na web a aplicação pode ser feita em PHP, ASP, JAVA ou outra linguagem de programação *web* e o retorno feito via HTML apenas se aplica no caso do texto com marcações, pois o navegador não interpreta linguagens de programações. Silva afirma que “podemos resumir hipertexto como todo o conteúdo inserido em um documento para a web”. Na figura 05 mostra um exemplo simples de uma página HTML.

Figura 05: Estruturada HTML

```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Título da Página</title>
6   </head>
7
8   <body>
9     <header>
10      CABEÇALHO
11    </header>
12    <section>
13      CONTEÚDO DA PÁGINA
14    </section>
15    <footer>
16      RODA-PÉ
17    </footer>
18  </body>
19 </html>
```

Fonte: Disponível em <<http://worldwildwebdesign.blogspot.com.br/2014/01/iniciando-em-html-5-parte-1-codigo.html>>

1.7 Cascading Style Sheets - CSS

CSS (*Cascading Style Sheets*) significa folhas de estilo em cascata, que assim como o HTML não é um tipo de linguagem de programação sendo apenas o mecanismo para adicionar estilo, formatação, espaçamento e cores aos elementos da página. A formatação é adicionada através das marcações feitas pelas *tags* do HTML. Silva (2015, p.129) define como.

O mecanismo de posicionamento CSS permitem ao autor alterar o comportamento padrão, não só alterando a ordem como também posicionamento elementos nível de bloco um ao lado do outro.

Na figura 06 mostra um exemplo simples de CSS, para estilização da tag 'h1' definindo fonte, tamanho e cor.

Figura 06: Estruturada do CSS

```

1  /*CSS*/           Arquivo: estilo.css
2
3  h1{
4      font-family: arial;
5      font-size: 20px;
6      color: #FFFF66;
7  }
```

Fonte: Disponível em <<http://andersonvania5ciclo.blogspot.com.br/2013/03/tipos-de-seletores-css-avancado.html>>

1.8 Personal Home Page - PHP

O PHP é um tipo de linguagem direcionada para o desenvolvimento *web*. Com ela podemos realizar funções lógicas como somas de números, laços de condições,

comparação de caracteres, manipulação e tratamento de dados e fazer conexão com diversos bancos de dados. Welling (2005, p.2) descreve o php como:

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

1.9 Framework

São estruturas que possuem o código pré-criado com funcionalidades pré-configuradas desenvolvidas por toda uma equipe, na qual permite que o usuário desenvolva o seu sistema de forma ágil com muitas funções já criadas ou trechos de códigos já implementados. Segundo Turine (2015, p.8) “Eles nos ajudam e muito a agilizar o processo de desenvolvimento, de forma organizada, evitando repetições de código e muito mais”.

Atualmente contém diversos frameworks para cada tipo de linguagem de programação, cada um com o seu ponto forte na qual cabe ao desenvolvedor escolher qual deles vai atender melhor a sua demanda, ajudando a organizar o código da aplicação.

1.9.1 Laravel

O Laravel é um *framework* PHP criado em 2011 para desenvolvimento *web* que utiliza a arquitetura MVC, lançado sob licença MIT Costa^{11,12}. O Laravel tem como um dos pontos fortes a organização e padronização do código, com ele é possível fazer integração em outras aplicações apenas adicionando outras bibliotecas na aplicação de forma simples através do *Composer*, melhorando a estrutura prevista na aplicação.

¹¹ Conhecida como licença X ou de licença X11, criada pelo Massachusetts Institute of Technology. permite a reutilização de software licenciado em programas livres ou proprietários.

¹² Disponível em: <<https://opensource.org/licenses/MIT>>

O Laravel é composto de uma série de ferramentas que facilitam de forma prática e eficiente no processo do desenvolvimento sendo uma ajuda a mais ao desenvolvedor. Uma dessas ferramentas são:

1.9.1.1 MVC (*Model View Controller*)

Com o MVC é possível organizar o código da aplicação, de forma que cada função do código tenha o seu lugar. Onde cada camada com sua responsabilidade organizacional possibilita um trabalho muito mais centralizada e modularizado. De acordo com Turine (2015, p. 21).

Ele cuida de atender as requisições e enviá-las para o local correto, no caso os nossos controllers. Os controllers, por sua vez, decidem o que fazer com as requisições, passando pela camada de model (que fazem acesso ao banco, executam as regras de negócio etc.) e logo em seguida delegam pra view que será exibida como resposta no navegador do cliente.

Shafik e Ramsey define MVC Como:

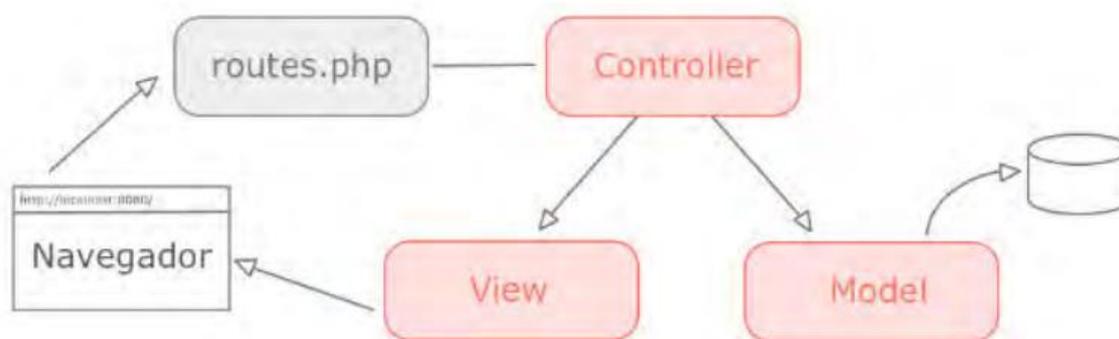
Model: O modelo fornece acesso aos dados para o controlador e notifica a *view* das mudanças.

View: A exibição exibe o modelo para o usuário final e fornece interfaces para aceitar entrada para o controller.

Controller: O controlador aceita a entrada do usuário e executa a lógica da aplicação.

Na figura 07 é possível ver um exemplo do fluxo do MVC.

Figura 07: Fluxo do MVC



Fonte: TURINI, 2015, p.21

1.9.1.2 Eloquent ORM

ORM (*Object Relational Mapping*) ou em português Mapeamento Objeto Relacional o qual faz a relação direta entre as diversas classes do Model, com as variadas tabelas do banco de dados. Schmitz e Georgii falam que:

Eloquent é a biblioteca de mapeamento objeto-relacional do Laravel. Através da implementação de um padrão de projeto chamado Active Record, que corresponde ao Model de uma aplicação, o *Eloquent* torna possível que possam ser realizadas operações no banco de dados sem a necessidade de escrever SQL. Ao invés do SQL, usa-se objetos para persistir dados nas tabelas do banco. (SCHMITZ; GEORGII, 2016, p.118)

Com isso podemos consultar dados nas tabelas, bem como inserir novos registros, atualizar, deletar de forma eficaz e simples, evitando a criação de instruções SQL, pois o Laravel realiza esta função facilitando a manipulação do banco de dados.

1.9.1.3 Composer

Nos últimos anos, surgiram gerenciadores de pacotes, dentre eles temos o Composer, no qual permite que instalemos diversos *frameworks* em PHP através de linhas de comando (*shell*), muito semelhante aos sistemas apt-get do Linux/Debian e seus derivados. O Composer funciona de uma forma diferenciada em relação aos

demais gerenciadores de pacotes existentes, pois ele não é instalado de forma global no sistema operacional, mas sim de forma local apenas no projeto. “Composer é uma ferramenta para o gerenciamento de dependências em PHP. Ele permite que você declare as bibliotecas de seu projeto depende e ele irá gerenciar (instalar / atualizar) -los para você” ¹³.

1.9.2 Bootstrap

O Bootstrap é um *framework* para desenvolvimento de sistema *front end*¹⁴ que utiliza CSS, JavaScript e HTML que tem como foco ajudar no desenvolvimento de forma rápida para múltiplos tipos de tela. Foi criada em meados de 2010 pelo *designer* e um desenvolvedor do Twitter de forma *open source* (Código livre). Silva (Bootstrap, 2015, p.20) define o *Bootstrap* como *Framework*:

Para desenvolvimento de sites e aplicações *web* responsivas e alinhadas com a filosofia *mobile first*. Torna o desenvolvimento front-end muito mais rápido e fácil. Indicado para desenvolvedores de todos os níveis de conhecimento, dispositivos de todos os tipos e projetos de todos os tamanhos.

1.10 JavaScript

O *JavaScript* é uma linguagem de programação desenvolvida por Brendan Eich na década de 90. Esta linguagem serve para criar interação nas páginas *web* gerando efeitos, alertas, dentre outras funcionalidades dinâmicas. Com ela podemos manipular todo o conteúdo e interagir com formulários feitos em HTML, realizando a comunicação com outras linguagens de forma mais dinâmica. Silva (2010, p.23) fala que:

¹³ Disponível em < <https://getcomposer.org/doc/00-intro.md>>

¹⁴ Páginas responsável por coletar a entrada de dados do usuário, formatando, e adequando os campos para diversos tipos de dispositivos de maneira organizada e estratégica

Foi criado pela Netscape em parceria com a Sun Microsystems com a finalidade de fornecer um meio de adicionar interatividade a um página *web*. A primeira versão, denominada JavaScript 1.0 foi lançada em 1995 e implementada em março de 1996 no *Netscape Navigator*.

1.11 Sistemas Operacionais

Os S.O. (Sistemas Operacionais) podem ser definidos como um conjunto de *softwares* usados para executar as tarefas solicitadas pelos operadores ou auxiliar nos serviços que serão realizados no decorrer do seu funcionamento em contato com o usuário comum, permitindo por exemplo, que o mesmo consiga realizar o seu manuseio possibilitando a realização de tarefas rotineiras como: ouvir música, ler textos, salvar e/ou gravar um arquivo em mídias. Tanebaum define:

Os sistemas operacionais executam basicamente duas funções não-relacionadas e dependendo de quem está falando, você ouve mais uma coisa ou outra. (TANEBAUM, 2000. p.18)

Popularmente no mercado os S.O mais conhecido são, *Microsoft Windows*, Linux, Mac OS, para dispositivo móvel *Android*, IOS e Windows Phone.

1.11.1 GNU/Linux

O sistema GNU/Linux tem sua origem no Unix, um sistema operacional multitarefa e multiusuário que tem a vantagem de rodar em uma grande variedade de computadores com diversos tipos de *Hardwares* (Equipamento fixo). Filho (2012, p.69), define como:

O Linux é apenas um kernel. Ele foi criado por Linus Torvalds, mas os aplicativos utilizados, deste o começo, foram feitos pela FSD para o projeto GNU. Assim sendo, o nome correto do sistema operacional é GNU/Linux.

2. DESENVOLVIMENTO

2.1 Configuração do Servidor

Para o desenvolvimento do ambiente do servidor foi utilizado o sistema operacional GNU/Linux Debian 8, após a instalação e devidas atualizações já é possível realizar a instalação dos pacotes para o funcionamento do servidor tais como, DHCP (*Dynamic Host Configuration Protocol* em português protocolo de configuração dinâmica de *host*), editor de texto VIM, NTP (*Network Time Protocol* - Protocolo para sincronização dos relógios), Instalação da versão 4 do Samba e *Squid* versão 3. No terminal logado com o usuário *root* (Administrador), foi executado o comando para realizar a instalação das bibliotecas e pacotes para o funcionamento do servidor como mostra na figura 08.

Figura 08: Comando para instalação dos pacotes.

```
root@gateway:~# apt-get install -y build-essential python-dev pkg-config libcups2root
@gateway:~# apt-get install -y build-essential python-dev pkg-config libcups2-dev cup
s krb5-user docbook-xsl libldap2-dev libattr1-dev libacl1-dev attr python-dnspython l
ibreadline-dev libbsd-dev libblkid-dev libpopt-dev cups acl gdb xsltproc libsasl2-dev
libaio-dev libpam-dev valgrind resolvconf autoconf ntp isc-dhcp-server vim squid3
```

Fonte: Do autor

Para o desenvolvimento da ferramenta é necessário configurar o arquivo `fstab` (*File systems table*, tradução tabela de sistemas de arquivos) que ficou da seguinte forma, como mostra na figura 09.

Figura 09: Arquivo `fstab`

```
root@gateway:/home/linux# grep CONFIG_EXT4_FS /boot/config-`uname -r`
CONFIG_EXT4_FS=m
CONFIG_EXT4_FS_POSIX_ACL=y
CONFIG_EXT4_FS_SECURITY=y
root@gateway:/home/linux#
```

Fonte: Do Autor

2.1.1 Configuração de Rede

O servidor tem duas placas de rede conectadas a `eth0` WAN (*Wide Area Network*, ou Rede de Longa Distância) com o IP fornecido pelo provedor de *Internet*, e `eth1` LAN (*Local Area Networks* ou Redes Locais) com o endereço IP estático, placa de rede responsável na propagação dos serviços para os clientes da rede. Na Figura 08 mostra como ficou o arquivo com as configurações de redes.

Figura 10: Configuração da placa de rede.

```
1 # This file describes the network interfaces available on your system
2 # and how to activate them. For more information, see interfaces(5).
3
4 source /etc/network/interfaces.d/*
5
6 # The loopback network interface
7 auto lo
8 iface lo inet loopback
9
10 # The primary network interface
11 allow-hotplug eth0
12 iface eth0 inet dhcp
13
14 # Placa de rede LAN
15 iface eth1 inet static
16     address 192.168.2.1
17     netmask 255.255.255.0
18     dns-search server.com
19     dns-domain server.com
20     dns-nameservers 192.168.2.1
```

Fonte: Do Autor

2.1.2 Configuração do DHCP

A configuração do DHCP foi realizada da seguinte forma: adicionando o conteúdo da figura 11 no arquivo *dhcpd.conf* localizado em */etc/dhcp/dhcpd.conf*.

Figura 11: Configuração do DHCP

```
1 ddns-updates on;
2 option domain-name "server.com";
3 option domain-name-servers 192.168.2.1, 192.168.2.1;
4 option netbios-name-servers 192.168.2.1;
5 option ntp-servers 192.168.2.1;
6 default-lease-time 600;
7 max-lease-time 7200;
8 log-facility local7;
9 subnet 192.168.2.0 netmask 255.255.255.0 {
10 range 192.168.2.10 192.168.2.30;
11 option routers 192.168.2.1; }
```

Fonte: Do Autor

2.1.3 Instalando o Samba 4

A instalação do Samba 4 foi realizada primeiramente baixando a versão 4 disponível no *Git*¹⁵ (Controlador de versão) na qual contém a última versão disponível e estável. Após baixar o código-fonte do Samba 4 foi necessário configurar o software através do comando: `./configure --enable-debug --enable-selftest`. Como mostra a figura 12 o resultado final do comando:

Figura 12: Configurar o Samba4

```

Checking for ftruncate extend : ok
Checking for header sys/sendfile.h : yes
Checking for linux sendfile support : ok
getcwd takes a NULL argument : ok
Checking for library gen : not found
Checking for getsnam : ok
Checking for header sys/fs/vx_quota.h : no
Checking for header sys/quota.h : yes
Checking for header ufs/ufs/quota.h : no
Checking for header xfs/xqm.h : no
for XFS QUOTA in <sys/quota.h> : not found
Checking for member dqb_fsoftlimit in struct dqblk : not found
Checking for member dqb_curbytes in struct dqblk : not found
Checking for header rpcsvc/rquota.h : yes
Checking for member getquota_rslt_u in struct getquota_rslt : ok
Checking for header ctdb.h : not found
building without cluster support: ctdb.h is required for cluster support
Checking whether we can compile with __attribute__((destructor)) : ok
Checking whether seekdir returns void : ok
Checking for pthread_attr_init : ok
Checking for header gpfs_gpl.h : no
Checking linker accepts -Wl,-no-undefined : yes
Checking linker accepts ['-undefined', 'dynamic_lookup'] : no
Checking linker accepts -Wl,--as-needed : yes
Checking for -lc not needed : ok
Checking configure summary : ok
Checking compiler accepts -g : yes
Checking compiler for PIE support : yes
configure finished successfully (1m12.404s)
root@ratonny: /home/linux/workspace/samba4#

```

Fonte: Do Autor

Após finalizar o comando anterior foi executado o comando `make && echo para` copilar o samba. A figura 13 mostra o resultado final do comando.

¹⁵ Disponível em <[git://git.samba.org/samba.git](https://git.samba.org/samba.git)>

Figura 13: Comando make

```
[3911/3923] Generating man/ldbrename.1
Note: Writing ldbrename.1
[3912/3923] Generating ndrdump.1
Note: Writing ndrdump.1
[3913/3923] Generating man/regdiff.1
Note: Writing regdiff.1
[3914/3923] Generating man/regpatch.1
Note: Writing regpatch.1
[3915/3923] Generating man/regshell.1
Note: Writing regshell.1
[3916/3923] Generating man/regtree.1
Note: Writing regtree.1
[3917/3923] Generating man/ntlm_auth4.1
Note: Writing ntlm_auth4.1
[3918/3923] Generating man/olschema2ldif.1
Note: Writing olschema2ldif.1
[3919/3923] Generating man/smbtorture.1
Note: Writing smbtorure.1
[3920/3923] Generating man/gentest.1
Note: Writing gentest.1
[3921/3923] Generating man/masktest.1
Note: Writing masktest.1
[3922/3923] Generating man/locktest.1
Note: Writing locktest.1
[3923/3923] Generating man/nmblookup4.1
Note: Writing nmblookup4.1
Waf: Leaving directory `/home/linux/workspace/samba4/bin'
'build' finished successfully (11m50.828s)
OK
root@gateway:/home/linux/workspace/samba4#
```

Fonte: Do Autor

Para realizar a instalação do samba em seu destino final (diretório */usr/local/*) foi executado o seguinte comando: *make install*. Este comando copiará todos os arquivos necessários para seu destino final, ou seja, para o sistema de arquivo correto. Na figura 14 mostra a saída do comando.

Figura 14: Saída do comando *make install*

```

rnal/subunit/tests/test_chunked.py
* installing lib/subunit/python/subunit/tests/test_details.py as /usr/local/samba/lib/python2.7/site-packages/samba/exte
rnal/subunit/tests/test_details.py
* installing lib/subunit/python/subunit/tests/test_progress_model.py as /usr/local/samba/lib/python2.7/site-packages/sam
ba/external/subunit/tests/test_progress_model.py
* installing lib/subunit/python/subunit/tests/test_run.py as /usr/local/samba/lib/python2.7/site-packages/samba/external
/subunit/tests/test_run.py
* installing lib/subunit/python/subunit/tests/test_subunit_filter.py as /usr/local/samba/lib/python2.7/site-packages/sam
ba/external/subunit/tests/test_subunit_filter.py
* installing lib/subunit/python/subunit/tests/test_subunit_stats.py as /usr/local/samba/lib/python2.7/site-packages/samb
a/external/subunit/tests/test_subunit_stats.py
* installing lib/subunit/python/subunit/tests/test_subunit_tags.py as /usr/local/samba/lib/python2.7/site-packages/samba
/external/subunit/tests/test_subunit_tags.py
* installing lib/subunit/python/subunit/tests/test_tap2subunit.py as /usr/local/samba/lib/python2.7/site-packages/samba/
external/subunit/tests/test_tap2subunit.py
* installing lib/subunit/python/subunit/tests/test_test_protocol.py as /usr/local/samba/lib/python2.7/site-packages/samb
a/external/subunit/tests/test_test_protocol.py
* installing lib/subunit/python/subunit/tests/test_test_results.py as /usr/local/samba/lib/python2.7/site-packages/samba
/external/subunit/tests/test_test_results.py
* installing bin/default/pidl/pidl.1p as /usr/local/samba/share/man/man1/pidl.1p
* installing bin/default/pidl/Parse::Pidl::Dump.3pm as /usr/local/samba/share/man/man3/Parse::Pidl::Dump.3pm
* installing bin/default/pidl/Parse::Pidl::Wireshark::Conformance.3pm as /usr/local/samba/share/man/man3/Parse::Pidl::Wi
reshark::Conformance.3pm
* installing bin/default/pidl/Parse::Pidl::Util.3pm as /usr/local/samba/share/man/man3/Parse::Pidl::Util.3pm
* installing bin/default/pidl/Parse::Pidl::NDR.3pm as /usr/local/samba/share/man/man3/Parse::Pidl::NDR.3pm
* installing bin/default/pidl/Parse::Pidl::Wireshark::NDR.3pm as /usr/local/samba/share/man/man3/Parse::Pidl::Wireshark:
:NDR.3pm
Maf: Leaving directory `/home/linux/workspace/samba4/bin'
'install' finished successfully (2m44.496s)
root@gateway:/home/linux/workspace/samba4#

```

Fonte: Do Autor

Para prover o serviço de AD e DC fornecidos pelo Samba4 foi executado o comando da figura 15 na qual encarregará de fazer as devidas configurações para os serviços.

Figura 15: Configurando o domínio.

```

root@gateway:/home/linux# samba-tool domain provision --realm=SERVER.COM --domain=SERVER --adminpass='Senha@123' --serve
r-role='domain controller'

```

Fonte: Do Autor

Após finalizar o comando anterior, foi obtido a saída da figura 16 com as informações do AD.

Figura 16: Saída do comando samba-tool.

```
Looking up IPv4 addresses
More than one IPv4 address found. Using 192.168.2.1
Looking up IPv6 addresses
No IPv6 address will be assigned
Setting up share.ldb
Setting up secrets.ldb
Setting up the registry
Setting up the privileges database
Setting up idmap db
Setting up SAM db
Setting up sam.ldb partitions and settings
Setting up sam.ldb rootDSE
Pre-loading the Samba 4 and AD schema
Adding DomainDN: DC=server,DC=com
Adding configuration container
Setting up sam.ldb schema
Setting up sam.ldb configuration data
Setting up display specifiers
Modifying display specifiers
Adding users container
Modifying users container
Adding computers container
Modifying computers container
Setting up sam.ldb data
Setting up well known security principals
Setting up sam.ldb users and groups
Setting up self join
Adding DNS accounts
Creating CN=MicrosoftDNS,CN=System,DC=server,DC=com
Creating DomainDnsZones and ForestDnsZones partitions
Populating DomainDnsZones and ForestDnsZones partitions
Setting up sam.ldb rootDSE marking as synchronized
Fixing provision GUIDs
A Kerberos configuration suitable for Samba 4 has been generated at /usr/local/samba/private/krb5.conf
Once the above files are installed, your Samba4 server will be ready to use
Server Role:      active directory domain controller
Hostname:         gateway
NetBIOS Domain:  SERVER
DNS Domain:       server.com
DOMAIN SID:      S-1-5-21-3152886210-701144793-2481743429
root@gateway:/home/linux#
```

Fonte: Do Autor

2.1.4 Configuração do Squid

Para realizar a configuração do Squid para o projeto apresentado, é necessário acessar a pasta do Squid através do comando da figura 17.

Figura 17: Diretório do squid

```
root@gateway:~# cd /etc/squid3/
```

Fonte: Do Autor

No arquivo *squid.conf* contém as configurações do *Squid*, tais como: porta diretório de log, acl e etc.

Para o funcionamento do *Squid*, de acordo com as necessidades do projeto foram adicionadas as seguintes linhas no arquivo *squid.conf* como mostra na figura 18. Das linhas de 3 a 10 são as configurações gerais, para o devido funcionamento do *squid*, tais como porta, tamanho do cache, tamanho da memória máxima utilizada pelo *squid*, diretório de cache, e-mail do administrador, nome do host e diretório da mensagem para páginas bloqueadas. As linhas de 14 a 17 são respectivas para realizar a autenticação do usuário criado no AD. As demais linhas são referentes as regras de acesso e inclusão dos arquivos adicionais.

Figura 18 – Configuração do Squid

```

1
2 ### CONFIGURAÇÕES GERAIS ###
3 http_port 3128
4 cache_mem 16 MB
5 maximum_object_size_in_memory 4 MB
6 cache_dir ufs /var/spool/squid3 100 16 256
7 access_log /var/log/squid3/access.log squid
8 cache_mgr marlon-20-12@hotmail.com
9 visible_hostname gateway.server.com
10 error_directory /usr/share/squid3/errors/pt-br
11
12 ### AUTENTICAÇÃO DO ADD ###
13 ### provide basic authentication via ldap for clients not authenticated via kerberos/ntlm
14 auth_param basic program /usr/lib/squid3/basic_ldap_auth -R -b "dc=server,dc=com" -D "cn=administrator,cn=Users,dc=server,dc=com" -W /etc/squid3/password.txt -f sAMAccountName=%s -h 192.168.2.1
15 auth_param basic children 15
16 auth_param basic realm Autentique-se com o usuario e senha do AD
17 auth_param basic credentialsttl 1 hours
18 #acl password proxy_auth REQUIRED
19 #auth_param basic casesensitive off
20
21 ### Controles de Acesso (ACL) ###
22 acl all src all
23 acl localhost src 127.0.0.0/32
24
25 ##### IP's Liberados #####
26 acl ip_liberado src "/etc/squid3/arquivos/ip/IpLiberados.txt"
27 http_access allow ip_liberado
28
29 ##### IP's Bloqueados #####
30 acl ip_bloqueado src "/etc/squid3/arquivos/ip/IpBloqueados.txt"
31 http_access deny ip_bloqueado
32
33 ##### Liberação para todos #####
34 #acl liberados url_regex -i "/etc/squid3/arquivos/liberados.txt"
35 #http_access allow liberados
36
37 ##### Arquivo com as configurações dos grupos é usuario #####
38 include /etc/squid3/body.conf
39
40 acl liberados url_regex -i "/etc/squid3/arquivos/liberados.txt"
41 http_access allow liberados
42
43 ##### Bloqueados para todos #####
44 acl bloqueados url_regex -i "/etc/squid3/arquivos/bloqueados.txt"
45 http_access deny bloqueados
46
47 ### Outros
48 http_access allow localhost
49 #http_access allow all
50 ##### Arquivo com as configurações final #####
51 include /etc/squid3/footer.conf
52

```

52,0-1

Fim

Fonte: Do Autor

Para melhor organização dos arquivos foi criado uma pasta com o nome “arquivos” dentro da diretório /etc/squid/. Dentro da mesma foi criado pastas e arquivos para a utilização da aplicação. Como mostra na figura 19

Figura 19 – Criar pasta e arquivo para organização do squid

```
root@gateway:/etc/squid3# cd arquivos/  
root@gateway:/etc/squid3/arquivos# mkdir grupo url users && touch bloqueados.txt liberados.txt boby.conf && echo "OK";  
OK  
root@gateway:/etc/squid3/arquivos#
```

Fonte: Do Autor

No arquivo ‘bloqueados.txt’ contém os sites bloqueados para todos os grupos e usuários, no arquivo ‘liberados.txt’ já contém os sites liberados para todos. As pastas criadas são grupos, *url*, *users* *ip*. Na pasta grupos contém arquivos com a seguinte nomenclatura *g_nomedogrupo.txt*. Nele contém o nome dos usuários que pertencem ao grupo. Na pasta url contém arquivos com as regras para os grupos com a seguinte nomeação: a primeira letra ‘r’ de regras em seguida da letra ‘g’ de grupo, ‘l’ ou ‘b’ de liberado ou bloqueado com o caráter underline ‘_’ nome do grupo. Ex.: *rgl_nomedogrupo.txt*, *rgb_nomedogrupo.txt*. Na pasta *users* contém as regras dos usuários, primeira letra ‘r’ de regras em seguida da letra ‘u’ de usuário, ‘l’ ou ‘b’ de liberado ou bloqueado com o caráter underline ‘_’ nome do usuario. Ex.: *rul_nomedousuario.txt*, *rub_nomedousuario.txt*. Na pasta *ip* contém dois arquivos ‘IpBloqueados.txt e IpLiberados.txt’ na qual contém os endereços IP dos clientes que poderá ter liberação sobre tudo ou bloqueado sobrescrevendo as demais regras. Na figura 20 mostra uma listagem dos arquivos criados nas pastas citadas.

Figura 20 – Arquivos das pastas

```

root@gateway:/etc/squid3/arquivos# ls grupos/
g_Aluno.txt  g_Professor.txt
root@gateway:/etc/squid3/arquivos# ls users/
rub_marlon.txt  rul_luiz.txt
root@gateway:/etc/squid3/arquivos# ls url/
rgl_Professor.txt
root@gateway:/etc/squid3/arquivos#

```

Fonte: Do Autor

No arquivo *body.conf*, irá conter as relações das regras para os grupos e usuário. Vale ressaltar que todo o conteúdo do arquivo é criado de forma automática pela aplicação sempre que houver alguma alteração no banco de dados. Veja o exemplo da estrutura do arquivo na figura 21.

Figura 21 – Arquivo body.conf

```

root@gateway:/etc/squid3# cat body.conf
acl u_fabiano proxy_auth fabiano
acl u_luiz proxy_auth luiz
acl u_marlon proxy_auth marlon
acl g_Aluno proxy_auth Aluno "/etc/squid3/arquivos/grupos/g_Aluno.txt"
acl g_Professor proxy_auth Professor "/etc/squid3/arquivos/grupos/g_Professor.txt"
acl rgl_Aluno url_regex -i "/etc/squid3/arquivos/url/rgl_Aluno.txt"
acl rgl_Professor url_regex -i "/etc/squid3/arquivos/url/rgl_Professor.txt"
acl rgb_Aluno url_regex -i "/etc/squid3/arquivos/url/rgb_Aluno.txt"
acl rgb_Professor url_regex -i "/etc/squid3/arquivos/url/rgb_Professor.txt"
acl rul_fabiano url_regex -i "/etc/squid3/arquivos/users/rul_fabiano.txt"
acl rul_luiz url_regex -i "/etc/squid3/arquivos/users/rul_luiz.txt"
acl rul_marlon url_regex -i "/etc/squid3/arquivos/users/rul_marlon.txt"
acl rub_fabiano url_regex -i "/etc/squid3/arquivos/users/rub_fabiano.txt"
acl rub_luiz url_regex -i "/etc/squid3/arquivos/users/rub_luiz.txt"
acl rub_marlon url_regex -i "/etc/squid3/arquivos/users/rub_marlon.txt"
http_access allow u_fabiano rul_fabiano
http_access allow u_luiz rul_luiz
http_access allow u_marlon rul_marlon
http_access deny u_fabiano rub_fabiano
http_access deny u_luiz rub_luiz
http_access deny u_marlon rub_marlon
http_access allow g_Aluno rgl_Aluno
http_access allow g_Professor rgl_Professor
http_access deny g_Aluno rgb_Aluno
http_access deny g_Professor rgb_Professor
root@gateway:/etc/squid3#

```

Fonte: Do Autor

2.1.5 Desenvolvimento da ferramenta

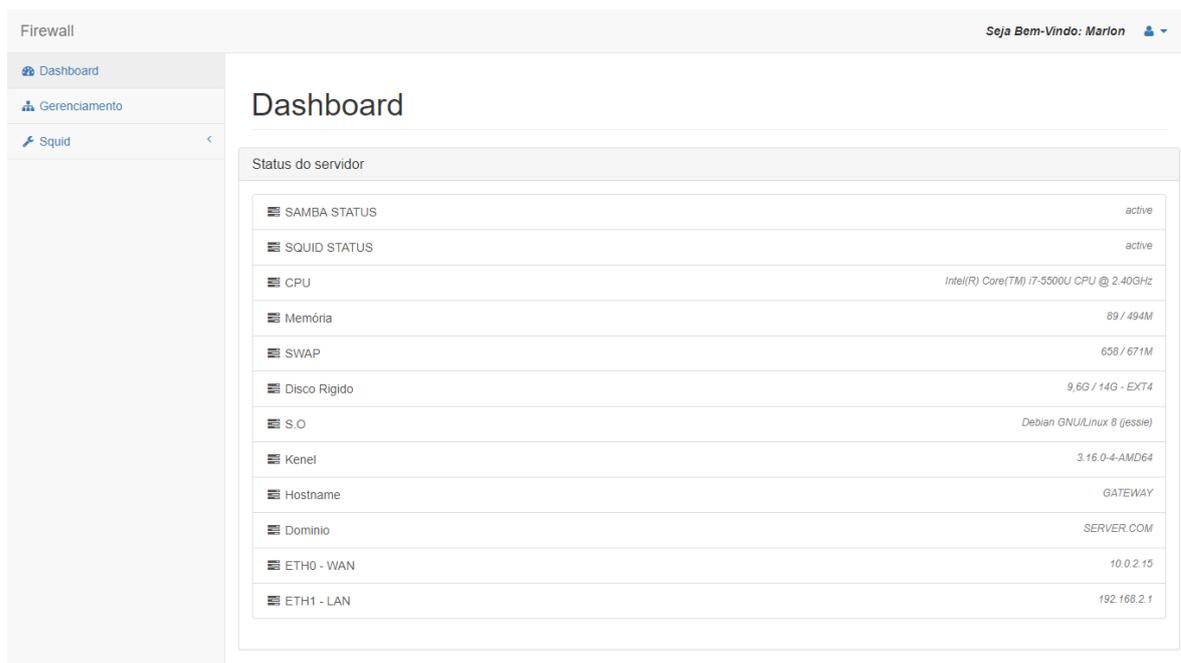
Para o desenvolvimento da ferramenta foi utilizado o Sistema Operacional GNU/Linux Debian 8, *Framework* Laravel versão 5.0 com o template (Tema) o *SB Admin 2 Bootstrap*, PHP versão 5, Banco de dados MariaDB, como editor a IDE *JetBrains PhpStorm*, em um ambiente virtualizado através do *VirtualBox*.

Através da ferramenta é possível realizar funções básicas no controlador de domínio, criar usuário, redefinir senha, desativar, mover de grupo e excluir, criar ou excluir grupos. Também é possível definir regras de acesso à *Internet*, qual *site* poderá ser acessado de forma geral para todos os grupos e usuários, liberado ou bloqueado. Ou regras específicas para um grupo ou usuários.

A aplicação encarregará de criar os arquivos com as informações dos grupos e dos usuários e regras para dentro do *Squid*, na qual estes arquivos sempre vão estar em concordância com o banco de dados.

Na tela principal da ferramenta contém informações do servidor tais como status do Squid e do Samba, consumo de memória e outros. Como mostra na figura 22.

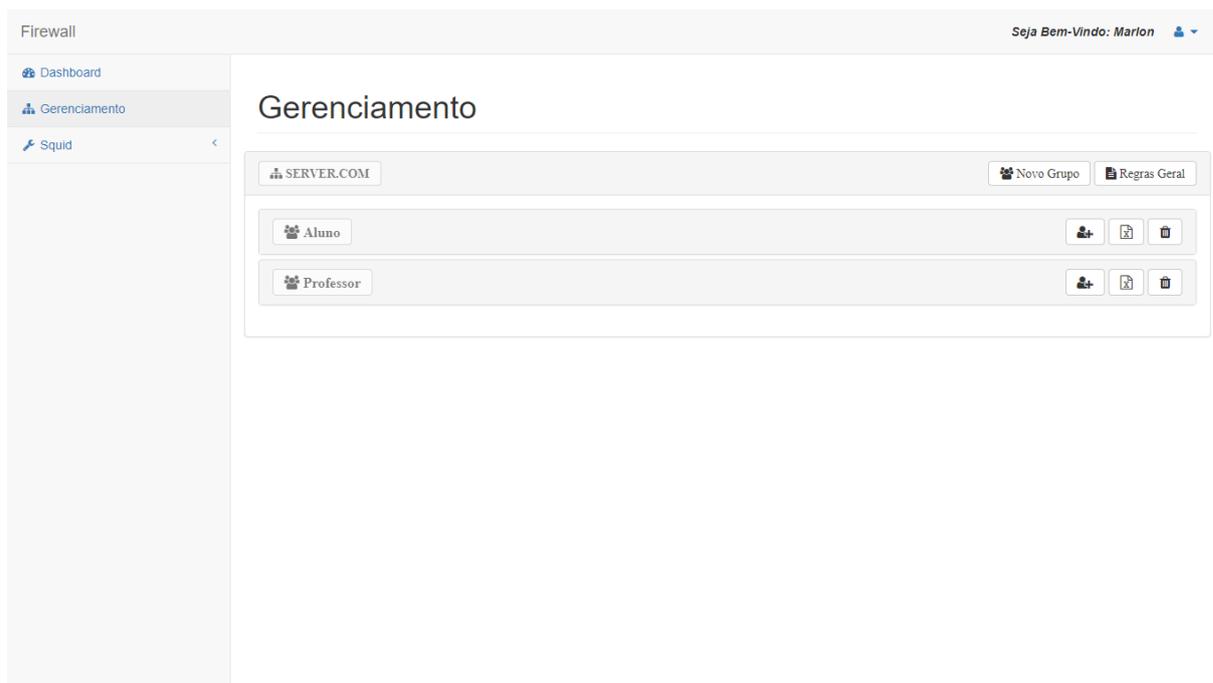
Figura 22: Tela principal.



Fonte: Do Autor

Na opção gerenciamento é possível criar grupos e usuários definindo quais *sites* eles poderão acessar e definir regras gerais para todos os grupos e usuários.

Figura 23: Tela de Gerenciamento



Fonte: Do Autor.

Figura 24: Página para criar novo usuário.

The screenshot shows a web interface for managing a firewall. A modal window titled "Novo Usuário" is open, displaying a form with the following fields:

- Nome Do Usuário:** Input field with placeholder "Nome do Usuário".
- Nome De Login:** Input field with placeholder "Nome do Login".
- Senha:** Input field with placeholder "Nova Senha".
- Confirme A Senha:** Input field with placeholder "Confirme a Senha".
- Grupo:** Dropdown menu with "Aluno" selected.

At the bottom right of the modal are two buttons: "Salvar" (green) and "Fechar" (red). The background interface shows a sidebar with "Dashboard", "Gerenciamento", and "Squid" options, and a main area with "Novo Grupo" and "Regras Geral" buttons.

Fonte: Do Autor

Figura 25: Página para definir regras geral.

The screenshot shows the "Regras do Geral" page in the Firewall management interface. The page title is "Regras do Geral". Below the title, there is a form for adding a new rule:

www. dominio .* Liberado [Salvar]

Below this form is a table of existing rules:

http.	Domínio	Tipo	Ação
www.	adx.doctum.edu.br	Liberado	[Atualizar] [Deletar]
www.	facebook.com	Bloqueado	[Atualizar] [Deletar]

Fonte: Do Autor

Figura 26: Página para defini regras para um usuário específico.

Fonte: Do Autor

Figura 27: Código para criar grupos no Samba e salva no BD.

```

43     * @return Response
44     */
45     public function store(Request $request) {
46
47         $nome = $request->all();
48         $this->validate($request, [
49             'nome' => 'required|unique:grupos|filled'
50         ]),
51         [
52             'nome.unique' => 'O grupo <strong>'. $nome['nome'] . '</strong> já está em uso.'
53         ]);
54     ];
55
56     $saida = shell_exec( cmd: "/usr/local/samba/bin/samba-tool group add " . $nome['nome'] . ". "; echo $? );
57     $saida = substr( string: "$saida", start: -2);
58     if( $saida != 0 ){
59         return back()->with('error', 'Alguma coisa deu errado no samba para criar o grupo \'' . $nome['nome'] . '\'' entre em contato com o administrador');
60     }
61
62     Grupo::create($nome);
63     return back()->with('success', 'Grupo \'' . $nome['nome'] . '\'' criado com sucesso');
64 }
65
66 /**
67  * Display the specified resource.
68  *
69  * @param int $id
70  * @return Response
71  */

```

Fonte: Do Autor

CONSIDERAÇÕES FINAIS

Com o término do trabalho foi adquirido conhecimento acadêmico para enfrentar o mercado de trabalho, pois cada dia a tecnologia cresce de forma veloz, com isso a questão de controle de acesso à *Internet* é uma alternativa fundamental para o bom funcionamento de qualquer empresa.

A proposta apresentada neste trabalho forneceu resultados satisfatórios, considerando que atendeu ao objetivo proposto. Após realizar pesquisas bibliográficas foi possível construir a ferramenta e validar as hipóteses levantadas no decorrer do projeto.

H0: Os meios realizados no ambiente corporativo são eficazes, na qual não necessitam de uma nova ferramenta, pelo custo e segurança e suporte oferecido pelas ferramentas disponíveis no mercado.

A Hipótese H0 não foi validada pois o suporte a integridade da aplicação é semelhante ao fornecidos pelas demais empresas disponíveis no mercado de forma segura. E de acordo com os resultados da aplicação a mesma apresentou ser fácil de utilizar e eficaz.

H1: O servidor proxy ajudará na velocidade da internet, pois pode ser feito o cache no servidor na qual são armazenados arquivos pequenos, como páginas .html e imagens, que serão entregues instantaneamente para os clientes sem precisar buscar todo o conteúdo do site novamente.

Uma configuração realizada no *Squid* é a função de cache de arquivos na qual melhorar na velocidade da internet sendo assim validando a hipótese.

H2: Com a implementação do servidor o funcionário poderá realizar sua função sem distração em sites alheios.

Poderá realizar bloqueio de sites que contém conteúdo impróprio para o ambiente de trabalho, validando a, pois, a ferramenta realizar bloqueio de sites.

H3: Com a ferramenta, podemos verificar o conteúdo da internet, que o funcionário está acessando e realizando a liberação ou bloqueio do conteúdo de forma simples e rápida.

Válida, porque através do SARG ferramenta para criar relatório dos sites acessados por cada usuário é possível realizar o bloqueio do mesmo caso for necessário.

H4: A ferramenta não terá despesas com licenciamentos sendo assim não haverá custo alto para empresa.

Válida, visto que todos os softwares utilizados na ferramenta são disponibilizados sobre licença GNU na qual não terá custo com licenciamento.

Conclui-se o principal objetivo traçado foi alcançado o que viabilizou o ganho acadêmico, profissional e pessoal.

Com a conclusão desse trabalho é com os objetivos alcançados mostrou ser uma aplicação vantajosa auxiliando o administrador realizar ações básica facilitando no seu dia-a-dia no controle de acesso à internet.

Para trabalhos futuros pode-se adicionados novos recursos relacionados no gerenciamento do AD com políticas de grupos, script de configurações e no *Squid* funcionalidade como definir regras pelo horário, controle de banda, configurações do *Squid* definindo porta, tamanho de memória e diretório de log.

REFERÊNCIAS

DAVEY, Shafik; Ben, Ramsey. *Zend PHP 5 Certification Study Guide* 3ª. Ed. Leanpub, 2015.

FILHO, João Eriberto Mota. *Descobrimdo o Linux*. 3ª. Ed. São Paulo: Novatec Editora, 2012.

MARCELO, Antonio. *Squid: configurando o proxy para Linux*. 4ª. Ed. Rio de Janeiro: Brasport, 2005.

MORIMOTO, Carlos E. Morimoto. *Servidores Linux, Guia Prático*. 1ª.Ed. São Paulo GDH Press e Sul Editores, 2008.

SCHMITZ, Daniel; GEORGII, Daniel Pedrinha. *Laravel e Angular JS*. Ed. Leanpub, 2016.

SILVA, Maurício Samy. *Fundamentos de HTML5 e CSS3*. 1ª. Ed. São Paulo: Novatec Editora, 2015.

SILVA, Maurício Samy. *HTML 5: A linguagem de marcação que revolucionou a web*. São Paulo: Novatec, 2011.

SOUZA, Welber Amaro Santos; PINTO, Cláudio José; BARBOSA, Dalva de Souza. *Guia de Consulta Rápida*. 1º. Ed. São Paulo: Novatec Editora, 2007.

TANENBAUM, Andrew Stuart. *Sistemas Operacionais: Projeto e implementação*. 2ª. Ed. Porto Alegre: Bookman. 2000.

TURINI, Rodrigo. *PHP e Laravel: Crie aplicações web como um verdadeiro artesão*. 1ª Ed. São Paulo: Casa do Código, 2015.

WELLING, Luke. *PHP e MySQL Desenvolvimento Web*. 6ª. Ed. Rio de Janeiro: Elsevier 2005

WESSELS, Duane. *Squid: The Definitive Guide*. 1º. Ed. O' Reilly, 2004.

10 anos de Samba. Disponível em: <https://www.samba.org/samba/docs/10years.html>. Acesso em: 11 ago. 2017.

Instalar do Active Directory. Disponível em: <http://comoaprenderwindows.com.br/wp-content/uploads/2015/04/22-Passos-Para-Instalar-o-AD-com-Sucesso.pdf>. Acesso em: 15 nov. 2017

O que é Active Directory. Disponível em: <https://technet.microsoft.com/pt-br/library/jj206711.aspx?f=255&MSPPErr=-2147217396>. Acesso em: 11 nov. 2017

Samba uma introdução. Disponível em: <https://www.samba.org/samba/docs/SambaIntro.html>. Acesso em: 10 ago. 2017.

Sobre o Samba. Disponível em: <https://www.samba.org/samba>. Acesso em: 11 ago. 2017.

SWAT. Disponível em: <https://www.samba.org/samba/docs/man/manpages-3/swat.8.html>. Acesso em: 11 ago. 2017.

The MIT License. Disponível em: <https://opensource.org/licenses/MIT>. Acesso em: 8 nov. 2017.