Vanderlei Freitas Junior
Vitoria Rodrigues dos Santos
Guilherme R. de Campos

TECNOLOGIA E REDES DE COMPUTADORES

5ª edição

INSTITUTO FEDERAL
Catarinense
Campus Avançado Sombrio

Vanderlei Freitas Junior
Vitoria Rodrigues dos Santos
Guilherme R. de Campos

TECNOLOGIA E REDES DE COMPUTADORES

5ª edição

2019
Instituto Federal Catarinense

Associação Brasileira
das Editoras Universitárias

INSTITUTO FEDERAL
Catarinense
Campus Avançado Sombrio

Direção Editorial	Vanderlei Freitas Junior
Capa e Projeto Gráfico	Ricardo Dal Pont
Editoração Eletrônica	Guilherme Rodrigues de Campos
	Vitória Rodrigues dos Santos
Comitê Editorial	Armando Mendes Neto
	Cleber Luiz Damin Ferro
	Guilherme Klein da Silva Bitencourt
	Jéferson Mendonça de Limas
	Joédio Borges Junior
	Marcos Henrique de Morais Golinelli
	Matheus Lorenzato Braga
	Sandra Vieira
	Vanderlei Freitas Junior
	Victor Martins de Sousa
Revisão	Gilnei Magnus dos Santos
Organizadores	Vanderlei Freitas Junior
	Guilherme Rodrigues de Campos
	Vitória Rodrigues dos Santos

INSTITUTO FEDERAL
Catarinense
Campus Avançado Sombrio

T255 Tecnologias e Redes de Computadores / Vanderlei de Freitas Junior; Guilherme Rodrigues de Campos; Vitória Rodrigues dos Santos (Orgs.). – 5. Ed. -- Sombrio: Instituto Federal Catarinense, 2019.

317 f.:il. color.

ISBN: 978-85-5644-044-0

1. Redes de Computadores. 2. Tecnologia da Informação e Comunicação.I. Freitas Junior, Vanderlei de. II. Rodrigues de Campos, Guilherme. III. Rodrigues dos Santos, Vitória. IV. Instituto Federal Catarinense. V. Título.

CDD: Ed. 21 -- 004.6

INSTITUTO FEDERAL
Catarinense
Campus Avançado Sombrio

Esta é uma publicação do
Curso Superior de

REDES DE COMPUTADORES

INSTITUTO FEDERAL
Catarinense
Campus Avançado Sombrio

Sumário

Sumário de Autores

Alta Disponibilidade para Serviços de Rede: Balanceamento de um Repositório *Cache* sob Demanda

Ruandreilton de Almeida Sousa[1], Marcos Henrique de Morais Golinelli[2]

[1]Acadêmico do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88.960-000 – Sombrio – SC – Brasil

[2]Docente do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88.960-000 – Sombrio – SC – Brasil

xslackx@live.fr,marcos.golinelli@ifc.edu.br

Abstract: *The high availability of systems has become indispensable in the online environment, so this work seeks to present a high availability solution for network services by applying the HAproxy software as a load balancer, being applied with the system cache package Apt-Cacher-NG. A bibliographic search was carried out from books and online documentation, respectively. As its nature is classified as applied research. Three tests were performed, one without caching, one with the caches in operation and the other with a cache off. As a result of the tests, we can observe the efficiency in the WAN bandwidth utilization using the cache, and the efficiency of the HAproxy software in performing the load balancing. Even with the simulation of defect in one of the cluster nodes, the service remained working.*

Resumo: *A alta disponibilidade de sistemas tornou-se indispensável no ambiente online, assim, este trabalho busca apresentar uma solução de alta disponibilidade*

para serviços de rede, utilizando o software HAproxy como balanceador de carga, sendo aplicado com o sistema de cache de pacotes Linux Apt-Cacher-NG. Foi realizada uma pesquisa bibliográfica a partir de livros e documentações online, respectivamente. Quanto sua natureza classifica-se como pesquisa aplicada. Foram realizados 3 testes, sendo um sem utilização do cache, um com os caches em funcionamento e outro com um cache desligado. Como resultado dos testes, pode-se observar a eficiência na economia de banda WAN utilizando o cache, e a eficiência do software HAproxy em realizar o balanceamento de carga. Mesmo com a simulação de defeito em um dos nós do cluster, o serviço se manteve em funcionamento.

1. Introdução

Em ambientes heterogêneos de sistemas operacionais, há uma demanda de acesso continuo à Internet, que resulta em uma sobrecarga na utilização da banda larga. Isso ocorre pois a largura de banda interna da LAN (*Local Area Network*), quase sempre é superior a largura de banda externa WAN (*Wide Area Network*), ocasionando o congestionamento da conexão (NEMETE, SNYDER, HEIN 2009), além do gasto do tráfego quando dados externos são solicitados continuamente, como por exemplo páginas Web, consultas de resolução de nomes para domínios, sincronização de dados, troca de mensagens ou atualizações de sistemas. Entretanto, conteúdos estáticos podem ser armazenados internamente em *cache* na rede LAN, diminuindo o tempo de espera e melhorando a eficiência interna da rede (TANENBAUM, 2011).

As aplicações ou serviços de rede localizam-se no topo da pilha TCP/IP. Os esforços de entrega dos dados (IP) e de

transporte (TCP e UDP) são executados para que os serviços de rede da camada de aplicação sejam capazes de enviar e receber dados (ALVES, 2008).

A característica habitual de uma aplicação proxy é a intermediação da conexão com a Internet ou com outros serviços de rede, onde as solicitações de acesso são norteadas pelo proxy e o retorno dos dados solicitados são obtidos pelo proxy e encaminhado ao solicitante (SOUSA, 2009).

Em casos onde sistemas operacionais baseados em Unix e Linux são utilizados, há a possibilidade de armazenar os pacotes de softwares localmente, agilizando as requisições futuras, diminuindo a latência de acesso a estes arquivos e economizando largura de banda da conexão WAN.

A necessidade de manter sistemas atualizados ou com o mínimo de indisponibilidade dos serviços, garantindo integridade e diminuição da quantidade de erros provocados por inconsistência da conectividade da rede, faz com que a implementação de um proxy *cache* seja viável, desde ambientes de TI que possuem pequeno porte até grandes Datacenters (TANENBAUM, 2011).

O presente trabalho tem como proposta implementar um ambiente alto disponível e escalável, utilizando softwares de código aberto, com intuito de disponibilizar um serviço de rede para repositório *cache* sob demanda. Durante sua realização, foram utilizados softwares como: Docker Engine, para gerenciar os contêineres; o HAProxy, para balanceamento de carga e o Apt-Cacher-NG, como solução de *proxy cache*, a implementação do ambiente e replicado em equipamentos diferentes, o que aumenta a redundância contra falhas ocorridas por *software* ou *hardware.* Desse modo, é possível escalar horizontalmente e verticalmente o ambiente proposto.

O artigo está organizado em cinco seções, sendo: introdução, referencial teórico, metodologia do trabalho, resultados e discussões e considerações finais. Após a parte textual encontra-se o Apêndice com o código utilizado nos testes.

2. Referencial Teórico

Neste capítulo são apresentados os fundamentos teóricos e ferramentas computacionais utilizadas neste trabalho, ressalta-se que tais fundamentos e ferramentas são abordados aqui de forma sucinta.

O tópico a seguir apresenta as definições para serviços altos disponíveis, utilizando advento da virtualização, agrupamento de equipamentos para se conseguir poder computacional ou micro segmentações de serviços balanceados.

2.1 Cluster

Conforme Hwang (2012), há dois ambientes em que *clusters* podem ser gerenciados: os virtuais e os físicos, ambos com diversas vantagens. O *cluster* físico é o agrupamento de máquinas físicas conectadas por uma rede física, já os *clusters* virtuais podem ser máquinas virtuais distribuídas em servidores físicos ou até mesmo em *clusters* físicos. A interligação das máquinas virtuais em um *cluster* ocorre em uma rede conectada logicamente sobre uma rede virtual.

Segundo Pitanga (2008), os clusters podem ser divididos em duas categorias: Alta Disponibilidade (HA – *High Availability*) e Alto Desempenho de Computação (HPC – *High Performance Computing*), cuja finalidade do HA é manter um serviço acessível de forma segura o maior tempo possível e do HPC é fornecer alto desempenho computacional.

Moraes (2010) e Pitanga (2008) concordam que para um serviço ter alta disponibilidade deve possuir mecanismos de segurança, como replicação do serviço implementado e a divisão em servidores diferentes. Isto possibilita que a aplicação se torne redundante à falhas, pois caso um nó do *cluster* pare os outros continuam a carga de trabalho sem uma parada de serviço.

Pitanga (2008) cita algumas vantagens na utilização de um *cluster* de computadores:

- **Alto desempenho** – Resolução de problemas muitos complexos utilizando processamento paralelo;
- **Escalabilidade** – Novos elementos podem ser adicionados, conforme a demanda de trabalho;
- **Tolerância a falhas** – A parada parcial de um sistema não o afeto em todo;
- **Baixo custo** – Maior poder de processamento realizando agregação de equipamentos;
- **Independência de fornecedores** – Possibilidade de utilizar equipamentos heterogêneos ou homogêneos com ou sem restrições de uso de software.

2.2 Proxies

Por definição, um proxy é um serviço com capacidade de fornecer acesso a outro ambiente. O uso mais comum para tal serviço é o fornecimento de acesso à internet por partes de máquinas que estão em uma LAN (ALVES, 2013).

A RFC[1] (*Request For Comments*) apresenta algumas terminologias de Proxy que são os mais utilizados, essas são citadas abaixo:

[1]As RFCs são documentos técnicos desenvolvidos e mantidos pelo IETF

- **HTTP cache** – As solicitações realizadas pelos usuários são armazenadas com o objetivo de reduzir o tempo de resposta e consumo de largura de banda em futuras requisições, sendo que o *cache* é compartilhado entre os navegantes, os objetos das mensagens HTTP ficam armazenados localmente e a aplicação *cache* controla a validade dos dados, a recuperação e a exclusão dos dados (R. FIELDING et al, 2014);

- **Proxy transparente** – Utilizado para alterar os cabeçalhos do protocolo HTTP, sendo estes o cabeçalho REMOTE_ADDR que contém o endereço IP do solicitante, o HTTP_VIA utilizado por proxies de encaminhamento e reversos, com o fim de evitar *loops* causados por requisições e identificação da versão do protocolo suportado pelo proxy e o HTTP_X_FORWARDED_FOR que engloba o endereço IP do solicitante e o endereço do proxy ou balanceador de carga (A. PETERSSON, M. NILSSON, 2014);

- **Proxy auto configurável –** Os métodos de auto configuração são aplicáveis para que o usuário que utiliza a conexão não necessite configurar seu acesso para navegar. as técnicas mais comuns para clientes autoconfiguráveis são o PAC (*Proxy Auto Configurarion*) e o WPAD (*Web Proxy Auto-Discovery Protocol).* o PAC é um script que determina por onde a conexão deve passar, já o WPAD utiliza um conjunto de proxies Web pré-existentes para executar a descoberta automática do proxy Web (I. COOPER et al, 2001);

- **Proxy reverso** – O proxy reverso funciona como um *gateway* ou túnel. Antes do cliente acessar o servidor

[1](Internet Enginnering Task Force), instituição que especifica os padrões que serão implementados e utilizados em toda a internet.

> original, sua conexão é analisada pelo servidor reverso que tem a função de encaminhar os pedidos de requisição para outros servidores de origem. Desse modo, o cliente comunica-se somente com o proxy delegado como se estivesse acessando o servidor original (I. COOPER et al, 2001).

2.3 Apt-Cacher-NG

Conforme Eduard (2011), "Apt-Cacher-NG é um Proxy *cache* para pacotes de softwares que são baixados por mecanismos de distribuições Unix / Linux a partir de servidores espelhos via HTTP."

As vantagens listadas pelo desenvolvedor da aplicação Apt-Cacher-NG são:

- **Implementação leve** – Permite uso em sistemas com pouca memória e processamento;
- **Concorrência nativa** – Evitando uso de recursos internos do sistema operacional para realizar essa função;
- **Suporte HTTP pipeline** – Usando um cliente interno com controle de fluxo nativo. "Pipelining é o processo para enviar solicitações sucessivas, sobre a mesma conexão persistente, sem esperar pela resposta. Isso evita a latência da conexão" (TELLES, 2017);
- **Confiabilidade –** Previamente os pacotes são armazenados localmente, o que garante a integridade dos dados enviados.

2.4 Balanceadores

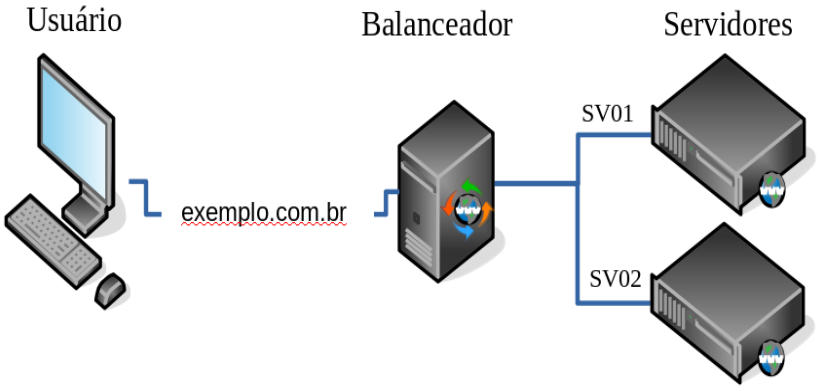
Balanceadores de carga tem por finalidade repartir a carga de trabalho entre componentes computacionais, sejam estes:

servidores, enlaces de rede, conjunto redundante de discos independentes, replicação de serviços ou outros recursos. Essa divisão na carga de trabalho limita a possibilidade de sobrecarga em um único sistema, evitando paradas por falhas.

O balanceamento de carga consiste na agregação de vários componentes para alcançar capacidade total de processamento acima da capacidade individual de cada componente. (HAProxy, 2018).

Na Figura 1, observa-se que quando o usuário acessa o endereço “exemplo.com.br”, a requisição da conexão é processada pelo balanceador, que encaminha para os servidores disponíveis, seguindo critérios configurados, por métricas como algoritmos, pesos ou checagem de saúde.

Figura 1 - Balanceamento de carga



Fonte: O autor, 2018.

A lista abaixo apresenta alguns softwares que são utilizados como balanceadores de carga:

- **LINUX HA –** Os projetos de softwares que são disponibilizados

pela Linux-HA são amplamente utilizados e importantes em muitas soluções de alta disponibilidade. Estão entre os melhores pacotes de softwares de alta disponibilidade para qualquer plataforma. “Estima-se que atualmente existam mais de trinta mil instalações em uso de missão crítica no mundo real desde 1999. ” (BEEKHOF, 2009);

- **LVS –** O LVS permite o compartilhamento de um endereço IP virtual entre um grupo de servidores, formando-se um *cluster*. Caso um participante do *cluster* falhe o outro nó assume, sem percepção por parte do usuário que acessa os servidores (MARTÍNEZ, RODRÍGUES, 2007).

- **HAPROXY –** O acrônimo HA, significa Alta Disponibilidade (*High Availability*), funcionando como um intermediador de uma conexão. O software HA Proxy é totalmente código aberto que funciona como uma solução de balanceamento de carga TCP ou HTTP, possuindo compatibilidade com Linux, Solaris e FreeBSD. É bastante utilizado por melhorar o desempenho e a confiabilidade, distribuindo a carga de trabalho para vários servidores (HAProxy, 2018).

2.5 Docker

Conforme Silva (2016), o Docker possibilita criar aplicações empacotadas com suas dependências completas. A ferramenta foi desenvolvida pela dotCloud com intuito de simplificar a administração de seu *PaaS (Plataform as a Service*[2]*)*. Dessa forma, seus funcionários de desenvolvimento poderiam criar

[2]É um ambiente de desenvolvimento e implantação completo na nuvem, com recursos que permitem fornecer tudo, de aplicativos simples baseados em nuvem a sofisticados aplicativos empresariais habilitados para a nuvem.

imagens (*Deploy*[3]) das aplicações sendo executadas dentro de contêineres Linux

Atualmente, o Docker possui duas versões, a *Community Edition* (CE) e a *Enterprise Edition* (EE), ambas versões possuem compatibilidade com os sistemas operacionais Mac, Windows e Linux.

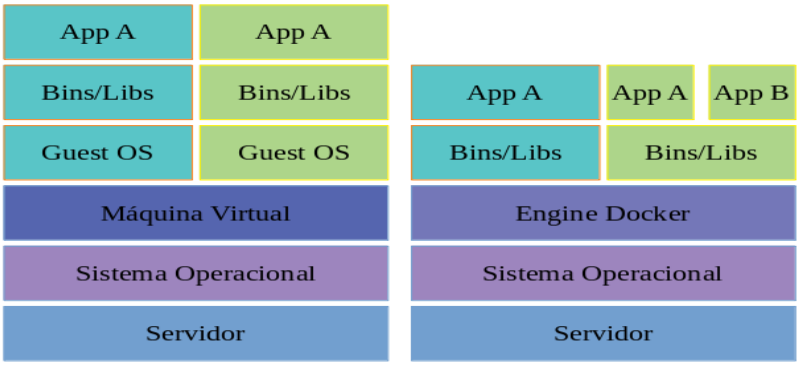
"O Docker, uma tecnologia *open source,* é atualmente o projeto e o método mais famoso para implantar e gerenciar contêineres Linux." (REDHAT, 2018).

A Figura 2 demonstra as diferenças da arquitetura entre aplicações, sendo executadas em um ambiente virtualizado e outro que utiliza a *Engine Docker*. Nota-se que a camada que faz a interface entre *Guest OS*[4] e o sistema operacional é a máquina virtual, que usa os recursos disponibilizados pelo sistema operacional e virtualiza o *hardware* para o *Guest OS*. Já os contêiners, precisam apenas dos binários das aplicações, uma vez que compartilham os recursos do Kernel utilizado pelo Sistema Operacional.

[3]Colocar em disposição, compilação, teste, implantação, gerenciamento e atualização.

[4]Sistema operacional convidado.

Figura 2 - Balanceamento de carga



Fonte: Adaptado de Silva, 2016.

2.6 Contêiner

Conforme Silva (2016), "Contêineres Linux são como os sistemas operacionais convidados, mas eles compartilham recursos como o Kernel do sistema operacional, rodam dentro de Cgroups." O *Cgroups* é uma abreviação para *control groups*[5], sendo um recurso do Kernel Linux desde sua versão 2.6.24 lançado em 24 de janeiro de 2008, seu desenvolvimento foi proposto por dois engenheiros do Google.

Abaixo uma lista de métodos de conteinerização que utilizam o conceito de *cgroups* para isolar aplicações por grupos:

- **CHROOT** – *Change root directory,* abreviado para chroot, é uma chamada de sistema (*System Call*) que tem por finalidade, "mudar o diretório raiz do processo chamado para outro especificado no caminho. Este

[5]Grupos de Controle fornecem um mecanismo para agregar e particionar conjuntos de tarefas, todos os seus processos filhos, em grupos hierárquicos com comportamento especializado.

diretório indicado se torna a pseudo raiz do sistema. O diretório raiz é herdado por todos os filhos do processo chamado" (KERRISK, 2017);

- **JAIL –** As *jails* são similares ao conceito de ambiente disponibilizado pelo uso do *chroot* no Linux, porém *jails* são de exclusividade de sistemas operacionais baseados em Kernel Unix/Like como FreeBSD, que desde a versão 4.X, possui j*ails*. Apesar das similaridades com o *chroot*, em um ambiente *chroot* tradicional os processos são limitados apenas na parte do sistema de arquivos que eles podem acessar, o restante dos recursos do sistema, os usuários do sistema, os processos em execução e o subsistema de rede são compartilhados pelos processos *chroot* e pelos processos do sistema *host* (RIONDATO, 2018);

Dentro deste contexto, Riondato (2018), afirma que "Os *jails* ampliam esse modelo virtualizando o acesso ao sistema de arquivos ao conjunto de usuários e ao subsistema de rede. Isso permite que o ambiente isolado tenha uma camada de isolamento do sistema *host*"

- **LXC –** O LXC permite um ambiente virtual com seu próprio processo e espaço de rede. Usando *namespaces* para reforçar o isolamento do processo e alavancar a própria funcionalidade de grupos de controle (cgroups) do Kernel, o recurso limita, contabiliza e isola CPU, memória, E/S[6] de disco e uso de rede de um ou mais processos. Este framework userspace pode ser considerado como uma forma muito avançada de chroot (KOUTOUPIS, 2018);

[6]Abreviação para entrada e saída.

- **DOCKER CONTÊINER –** Pode se considerar que o contêiner Docker, “é um pacote de software leve, autônomo e executável que inclui tudo o que é necessário para executar um aplicativo: código, tempo de execução, ferramentas e bibliotecas do sistema e configurações.” (DOCKER, 2018);
- **UNIKERNEL –** Diferentes dos diversos tipos de tecnologias contêineres disponíveis para sistemas operacionais, “Unikernels são Kernels de sistemas operacionais especializados que são escritos em uma linguagem de alto nível e atuam como componentes de software individuais.” (MADHAVAPEDDY, SCOTT, 2014).

2.7 Automação

Automação tem sua origem do latim *Automatus*, que é definido como mover-se por si (MAHONEY, 2014). Implementações em softwares estão se tornando cada vez mais autônomas, isto em resposta a necessidade de ter sistemas distribuídos geograficamente com o mínimo de tempo necessário para sua instalação e manutenção, evitando esforços repetitivos.

Alguns utilitários podem ser utilizados em conjunto para automatizar tarefas cotidianas, como é o caso das ferramentas apresentadas a seguir:

- **SHELL –** Conforme Ward (2015), “O shell é uma das partes mais importantes de um sistema Unix”. Pode-se executar comandos do sistema, além de possuir funções internas e procedimentos como outras linguagens de programação. “Tudo isso para tornar um pouco mais "esperta" e flexível essas chamadas de comandos feitas pelo usuário. ” (JARGAS, 2018);

- **YAML –** "(*Ain't Markup Language*), É uma linguagem de serialização de dados projetada para ser amigável para o homem e funciona bem com as modernas linguagens de programação para tarefas cotidianas comuns." (YAML, 2009);

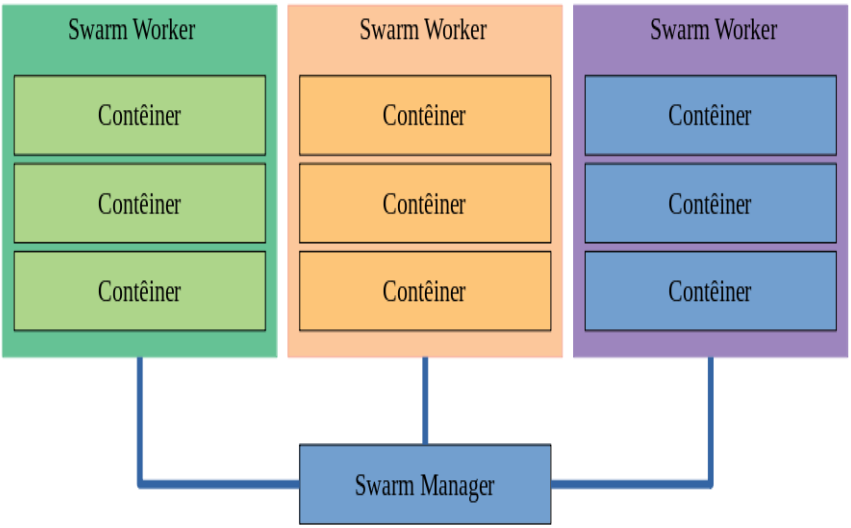
- **DOCKER COMPOSE –** Compose é uma ferramenta para definir e executar aplicativos Docker com vários contêineres. Com o Compose, utiliza-se um arquivo em YAML para configurar os serviços do seu aplicativo. Então, com um único comando, pode se criar e iniciar todos os serviços da aplicação conteinerizada (DOCKER, 2016);

- **DOCKER SWARM** – Conforme Silva (2016), "O Docker Swarm é uma ferramenta que possibilita agrupar diversos Docker hosts montando um *cluster* onde podem rodar vários contêineres com diversas aplicações dentro."

A Figura 3 ilustra o Swarm Manager que gerencia todos os Swarm Workers que fazem parte do *cluster*, a comunicação entre o gerenciador e os contêiners utiliza ambos protocolos TCP e UDP, sendo que a porta TCP 2377 é utilizada para o gerenciamento do *cluster*, para o tráfego da rede *overlay*[7] (sobreposta) o UDP porta 4789 e para comunicação entre os nós TCP e UDP porta 7946.

[7]Utiliza o VXLAN (*Virtual eXtensible Local Area Network*) para interconexão das redes sobrepostas, utilizando os serviços de rede da camada *Ethernet 2*, que trabalha sobre as limitações das VLANs, provisionando extensibilidade e flexibilidade.

Figura 3 - Exemplo de um cluster de Docker Swarm



Fonte: Adaptado de Docker, 2018.

2.8 Redes Virtualizadas

Segundo Fátima (1996), "Uma rede virtual é construída sobre várias tecnologias de *switching*[8], ao contrário das redes compartilhadas que são baseados em cabos compartilhados interconectados através de roteadores." A tecnologia VLAN (*Virtual Local Area Network*) descrita pelo padrão IEEE[9] 802.1Q[10], possibilita que uma rede física seja logicamente repartida. Essa segmentação da rede conecta-se a adaptadores

[8]É o processo de interligação de dois pontos em uma rede, seja por circuitos ou por pacotes.

[9](**Instituto de Engenheiros Eletricistas e Eletrônicos**) é uma organização profissional que estabelece questões de interconectividade e interação dos padrões do instituto com o modelo OSI.

[10]Estabelece um método padrão para a marcação de quadros *Ethernet*.

Ethernet[11] virtuais. O tráfego proveniente de VLANs são encaminhados a partir de comutadores virtuais.

"Essa separação é obtida pela identificação de pacotes *Ethernet* com suas informações de associação da VLAN e, em seguida, restringindo a entrega aos membros dessa VLAN." (IBM, 2017).

A Libnetwork implementa o modelo de rede de contêiner (CNM) que formaliza as etapas necessárias para fornecer rede para contêineres enquanto fornece uma abstração que pode ser usada para suportar vários drivers de rede (DOCKER, 2018).

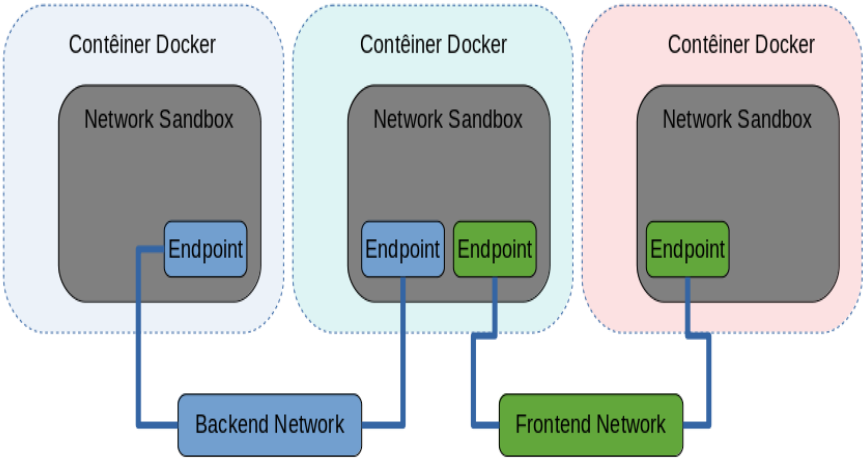
2.8.1 CNM (*Container Network Model*)

Conforme Silva (2016), o modelo CNM possui três entidades:

- O Network Sandbox contém a configuração da pilha de rede de um contêiner.
- O Endpoint é responsável por conectar uma interface de rede a uma rede. Podendo ter mais de uma Endpoint em cada Network Sandbox.
- A Network é a rede de fato onde podemos conectar e desconectar os Endpoints dos contêineres.

[11]O termo *Ethernet* refere-se à família de produtos de rede local (LAN) cobertos pelo padrão IEEE 802.3.

Figura 4 - Modelo CNM



Fonte: Adaptado de Silva, 2016.

3. Materiais e métodos

Esta sessão apresenta a classificação deste trabalho, bem como os materiais e procedimentos realizados para implementação dos softwares.

3.1. Métodos

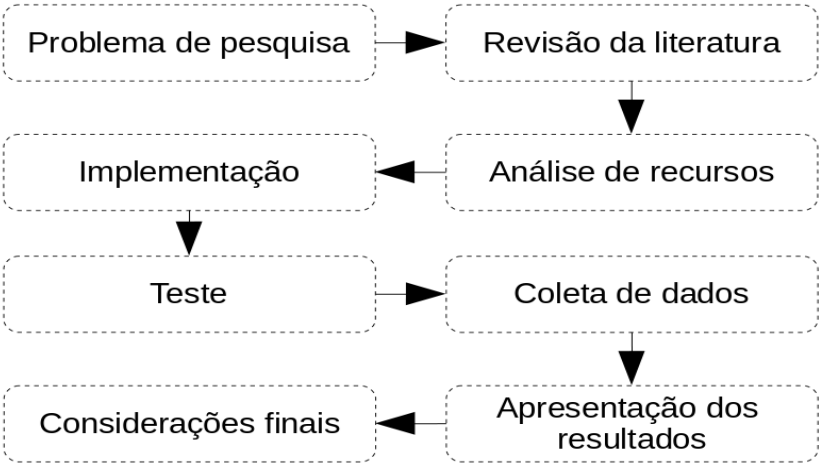
Este trabalho utilizou uma metodologia bibliográfica, a qual de acordo com Lima e Mioto (2007, p.38) se estabelece por intermédio de "um conjunto ordenado de procedimentos de busca por soluções, ligado ao objeto de estudo". Aplicando-se, também, à pesquisa tecnológica, sendo que, de acordo com Freitas Junior *et al* (2014, p. 12) "O conhecimento científico é limitado pela teoria, enquanto que o conhecimento tecnológico pela tarefa."

Quanto sua natureza, este trabalho classifica-se como pesquisa aplicada, que tem como objetivos a aplicação de seus

resultados à prática, tornando o problema mais claro, os fatos e fenômenos relacionados (FLEURY, WERLANG, 2017).

As etapas de desenvolvimento seguem o fluxo da figura 5.

Figura 5. Fluxo de metodologia



Fonte: O autor, 2018.

Dois problemas foram levantados, a implantação de sistema de proxy de pacotes Linux e o balanceamento de carga para alta disponibilidade de serviços. A partir disso foi realizada a pesquisa na literatura em busca de conceitos, tecnologias e softwares necessários para realização de tais tarefas, resultando na análise de recursos.

Para implementação utilizou-se *scripts* escritos em *Shell* e YAML na construção do ambiente. Os testes foram organizados em três modos, sem uso do *cache*, com uso do *cache* e sem um nó do *cluster*.

Após os testes, foi realizado a coleta de dados por meio dos arquivos gerados pelos *scripts*, que na sequência foram

devidamente tratados para extração dos dados para criação de gráficos.

3.2 Materiais

Para implementação deste ambiente se fez necessário a utilização de três computadores para formação do *clusters,* sendo um como balanceador com o HAProxy, e dois como nós com o Apt-Cacher-Ng e outros três computadores foram utilizados para testes como clientes. O quadro abaixo descreve detalhadamente o *hardware,* software e os sistemas operacionais de cada componente do ambiente.

Quadro 1. Software e *Hardware*

Sistema Operacional	**CPU**	**Mem. RAM**	**Software**	**Quantidade**
Fedora 28	*4x Intel® Core™ i5-4300U CPU @ 1.90GHz*	*8GB*	Docker Engine 18.09.0 HAProxy	1
Ubuntu 18.04 LTS	*8x Intel® Core™ i7-3770 CPU @ 3.40GHz*	*8GB*	Docker Engine 18.09.0 Apt-Cacher-NG	2
Ubuntu 18.04 LTS	*8x Intel® Core™ i7-3770 CPU @ 3.40GHz*	*8GB*	Clientes	3

Fonte: O autor, 2018.

3.3 Implementação

Para o ambiente de produção são usados scripts de configuração que automatizam a construção dos serviços necessários, tendo

em vista que pudessem acompanhar a escalabilidade horizontal e a elasticidade.

A Figura 6, apresenta o conteúdo do arquivo de configuração escrito na linguagem YAML (*Ain't Markup Language*) interpretado pelo Docker que provisionará o ambiente proposto. O arquivo lista algumas opções que devem ser utilizadas para execução dos serviços. Os campos: ***version:*** define qual é a versão do arquivo de configuração; ***services:*** quais serviços serão executados; ***balanceadorha:*** é o nome do serviço que será criado; ***image:*** a imagem utilizada; ***deploy:*** especifica a configuração relacionada à implementação e execução do serviço; ***ports:*** determina as portas utilizadas na ordem *host* para contêiner; ***volumes:*** monta diretórios do *host* para o contêiner; ***networks:*** o nome da rede e suas opções.

Figura 6. *Compose File*

```
version: "3.3"
services:


 balanceadorha:
   image: haproxy:1.8
   deploy:
    replicas: 1
    endpoint_mode: vip
    resources:
     limits:
      cpus: "1"
      memory: 1GB
    restart_policy:
      condition: on-failure
      delay: 5s
      max_attempts: 3
      window: 120s
   ports:
   - 9999:9999
   - 3148:3148
   volumes:
   - '/opt/configs/ha/:/usr/local/etc/haproxy:ro'
   networks:
    proxycache:

 cacheng:
   image: sameersbn/apt-cacher-ng:3.1-1
   ports:
    - 3142:3142
   deploy:
    mode: replicated
    replicas: 2
    resources:
     limits:
      cpus: "1"
      memory: 960MB
     reservations:
      memory: 480MB
    restart_policy:
      condition: on-failure
      delay: 5s
      max_attempts: 3
      window: 120s
   networks:
    proxycache:

networks:
  proxycache:
   driver: overlay
   attachable: true
```

Fonte: O autor, 2018.

Para a implementação, o *hardware* principal deve ter outro de *backup* pronto para uso. As aplicações correntes devem estar sincronizadas entre os dois *hardwares*, mantendo a confiabilidade e consistência dos dados armazenados em *cache.*

É possível visualizar na Figura 7 o processo de inicialização do *clusters*. A saída do comando ***docker swarm***, gera um *token* que deve ser adicionado aos nós *slaves* do *cluster* criado, como demonstra na Figura 8.

Figura 7 - Inicialização do *Cluster*

```
[root@master /]# docker swarm init --advertise-addr 172
.16.38.5
Swarm initialized: current node (slmziaph4he6b40voqhrda
9cv) is now a manager.

To add a worker to this swarm, run the following comman
d:

    docker swarm join --token SWMTKN-1-5apl297u905wfx9v
ojx9ij5w0fimskiwcavcnzmk6i2f1jp5x0-8iy1mrszoizf0qknldjt
cp6qw 172.16.38.5:2377

To add a manager to this swarm, run 'docker swarm join-
token manager' and follow the instructions.
```

Fonte: O autor, 2018.

Figura 8 - Adicionando *slaves*

```
[root@slave1:/]# docker swarm join --token SWMTKN-1-5ap
l297u905wfx9vojx9ij5w0fimskiwcavcnzmk6i2f1jp5x0-8iy1mrs
zoizf0qknldjtcp6qw 172.16.38.5:2377
This node joined a swarm as a worker.
[root@slave2:/]# docker swarm join --token SWMTKN-1-5ap
l297u905wfx9vojx9ij5w0fimskiwcavcnzmk6i2f1jp5x0-8iy1mrs
zoizf0qknldjtcp6qw 172.16.38.5:2377
This node joined a swarm as a worker.
```

Fonte: O autor, 2018.

Após a inclusão dos *slaves* no *clusters,* é necessário executar o comando ***docker stack deploy,*** para o ambiente ser orquestrado e escalonado. A opção --compose-file indica onde se localiza o arquivo YAML que será interpretador, **adsr** foi o nome escolhido para dar ao *clusters* provisionado. a Figura 9 apresenta a execução do comando.

Figura 9 – Orquestrando o ambiente.

```
[root@master ha]# docker stack deploy --compose-file=/opt/configs/ha/ng.yml adsr
Creating network adsr_proxycache
Creating service adsr_cacheng
Creating service adsr_balanceadorha
```

Fonte: O autor, 2018.

No arquivo de configuração do HAproxy foram incluídas as linhas apresentadas na Figura 10, as regras definidas realizam o balanceamento de carga entre os dois nós do *clusters*.

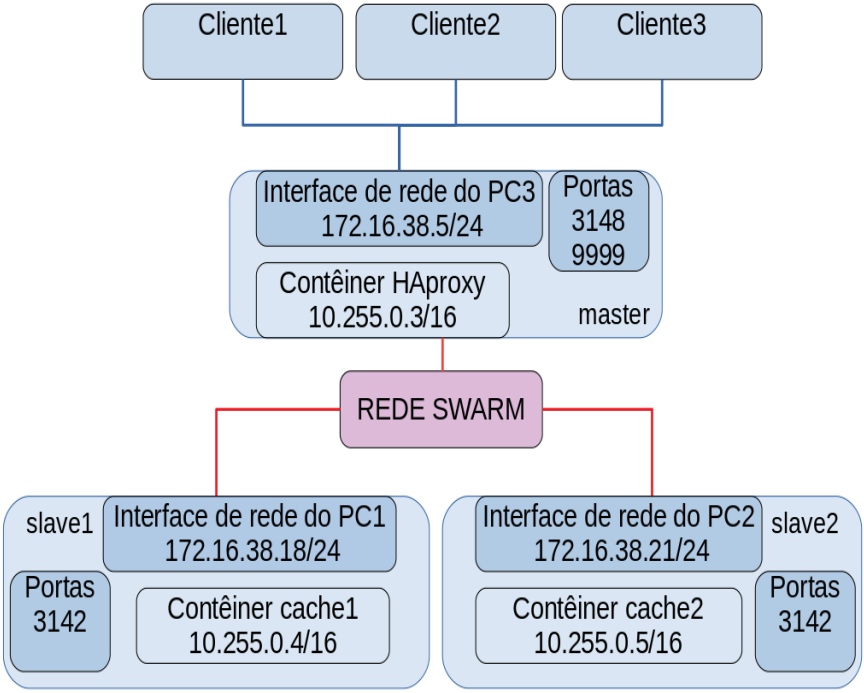
Figura 10. Configuração HAproxy.

```
frontend cache
    bind *:3148
    acl cacher hdr(host) -i tccproxycache
    use_backend aptcacherng if cacher
    default_backend aptcacherng
    option http_proxy
    option http-use-proxy-header
    option http-no-delay
backend aptcacherng
    balance roundrobin
    timeout check 8
    server cache2  172.16.38.21:3142 check fall 3 rise 2 weight 1
    server cache1  172.16.38.18:3142 check fall 3 rise 2 weight 1
    option http-no-delay
```

Fonte: O autor, 2018.

A Figura 11 detalha a topologia do ambiente proposto, onde o *master* executa o HAProxy que realiza o balanceamento de carga entre o *slave1* e o *slave2,* que por sua vez executam o *proxy cache* Apt-Cacher-NG.

Figura 11 - Topologia



Fonte: O autor, 2018.

Os testes foram realizados em três modos. Cada teste foi composto por uma sequência de *download* de softwares, que foram requisitados pelo gerenciador de pacotes *apt-get*. Somente o segundo e terceiro modos utilizaram o *cache*. No terceiro modo um dos nós do *cluster* foi desligado a fim de comprovar a alta disponibilidade do ambiente.

A coleta de dados foi realizada a partir das informações oriundas dos testes. Para isso, foi utilizado um *script* em *Shell* que pode ser visualizado no apêndice do trabalho que automatiza o processo de *download* realizado nos testes. A Figura 12

demonstra a execução do *script* denominado de teste, a saída total do comando foi suprimida.

Figura 12 - Execução do script.

```
root@cliente:/home# ./teste
TESTE SEM USO DO CACHE
Sun Dec  2 17:27:58 UTC 2018
LIMPANDO CACHE LOCAL
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
TESTE DOWNLOAD PACOTES
```

Fonte: O autor, 2018.

No terceiro teste, um nó do *cluster* foi desligado. A Figura 13 a seguir mostra como o painel de *status* do HAproxy apresenta o *slave1* desligado.

Figura 13 - Web *Status* HAproxy

Note: NOLB / DRAIN = UP with load-balancing disabled.

cache

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	8	-	0	15	10 000	218			700 854	11 754 395 568	0	0	0					OPEN							

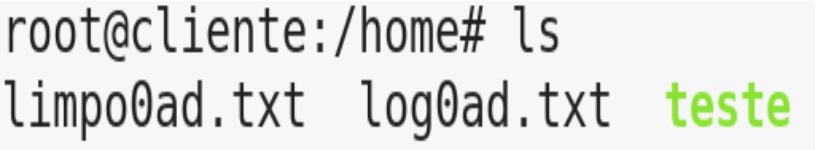
aptcacherng

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
cache1	0	0	-	0	139		0	10	-	2 293	2 293	37m10s	479 289	8 412 071 348		0		0	1	0	0	8m37s DOWN	L4TOUT in 2001ms	1	Y	-	7	3	8m37s	-
cache2	0	0	-	0	38		0	6	-	1 007	1 007	51m18s	221 565	3 342 324 220		0		0	3	0	0	11s UP	L4OK in 0ms	1	Y	-	10	4	0s	-
Backend	0	0		0	139		0	10	1 000	3 300	3 300	37m10s	700 854	11 754 395 568	0	0		0	4	0	0	11s UP		1	1	0		3	9m1s	

Fonte: O autor, 2018.

Em análise dos dados coletados, são separados os arquivos de *logs* gerados pelos testes, como demonstra a Figura 14.

Figura 14 - Logs.



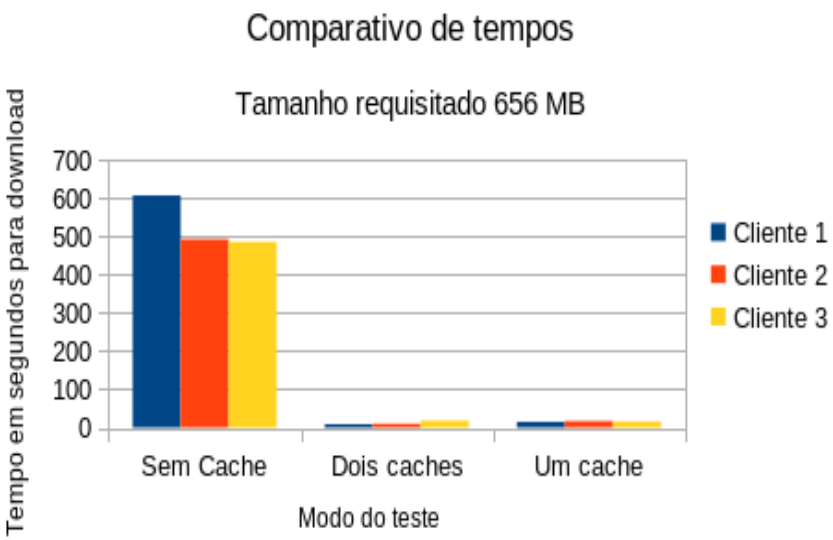
Fonte: O autor, 2018.

4. Resultados e discussões

Nesta etapa de apresentação dos resultados, os dados coletados e organizados são dispostos em gráficos que demonstram os três cenários propostos. No primeiro modo de teste não foi utilizado o *cache* e todos os dados solicitados foram obtidos a partir de repositórios disponíveis na Internet, sendo que a banda larga total foi de 30 Mbps para este teste. No segundo modo de teste com a utilização do *cache* e no último teste sem um dos nós *cache* do *cluster*. Nos testes realizados o pacote solicitado para ambos clientes foi o software 0ad, tendo tamanho total de 656MB. Este foi escolhido para os testes pois possui tamanho semelhante a uma distribuição Linux, quando este desempenha o processo de *download* dos seus pacotes de softwares no momento de sua instalação.

A figura 15 apresenta os testes em relação à quantidade de tempo que se levou para concluir o *download* do arquivo solicitado pelos clientes.

Figura 15. Comparação de tempo entre os clientes



Fonte: O autor, 2018.

Pode-se observar que no primeiro teste, sem utilização do *cache* o cliente 1 demorou 606 segundos, o cliente 2 demorou 492 segundos, já o cliente 3 concluiu o *download* em 485 segundos. Destaca-se que os três clientes realizaram o teste simultaneamente, de forma que, por meio do tamanho dos arquivos e a média do tempo gasto na transferência é possível obter a taxa média de transferência, que foi de 29,87 Mbps.

No segundo teste, utilizando os dois *caches* o cliente 1 finalizou o d*ownload* em 9 segundos, o cliente 2 terminou em 10 segundos, e o cliente 3 concluiu em 18 segundos. A taxa média de transferência foi de 1.312 Gbps.

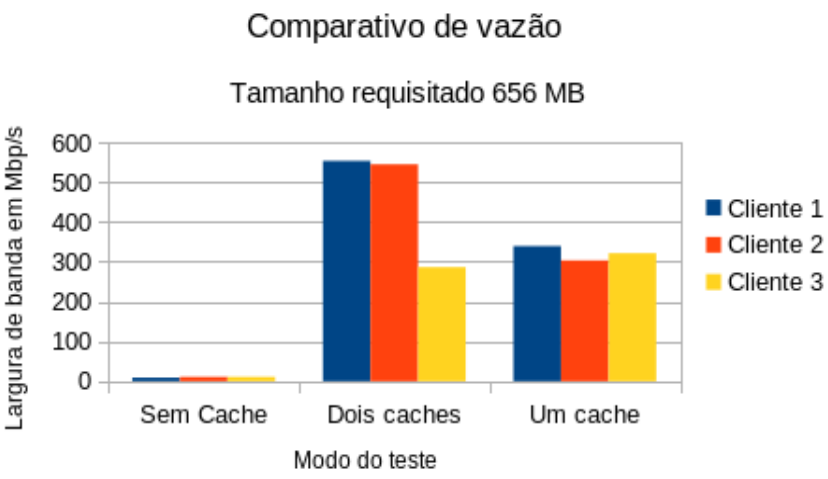
Para o último teste um dos nós *cache* do *cluster* foi desligado. Ao solicitar o arquivo o cliente 1 finalizou o

download em 15 segundos, já o cliente 2 terminou em 17 segundos e o cliente 3 acabou em 16 segundos. A taxa média de transferência foi de 984 Mbps.

A soma dos tempos nos modos de teste com utilização do *cache* e sem utilização de um dos nós do *cluster,* em comparação com o tempo gasto no primeiro teste, foi 94,64% menor do que o tempo total utilizado no primeiro teste.

A Figura 16 dispõe os resultados dos testes dos três clientes em relação à taxa de transferência obtida até a finalização do *download* dos arquivos requisitados.

Figura 16. Taxa de transferência entre os clientes



Fonte: O autor, 2018.

Observa-se que no primeiro teste, sem a utilização do *cache,* o cliente 1 atingiu 8,664 Mbp/s, o cliente 2 alcançou 10,664 Mbp/s, já o cliente 3 consumiu apenas 10,816 Mbp/s de largura de banda.

No segundo teste, com a utilização dos dois *caches* , o cliente 1 alcançou 553,6 Mbp/s, o cliente 2 obteve 544,8 Mbp/s, já o cliente 3 atingiu 286,4 Mbp/s. É possível obter a eficiência na transferência dos arquivos solicitados a partir da soma das taxas alcançadas pelos clientes e sua divisão pela capacidade nominal da interface de rede, com isso o segundo teste foi 138,48% eficaz na transferência dos arquivos solicitados.

No último teste realizado, com um nó *cache* do *clusters* desligado foi verificado que não houve perda de vazão na taxa de transferência nem aumento do tempo em relação ao primeiro teste apresentado na figura 15. Observou-se que o cliente 1 obteve 339,2 Mbp/s, o cliente 2 atingiu 303,2 Mbp/s, já o cliente 3 alcançou a marca de 321,6 Mbp/s. A partir dos dados dispostos também é possível determinar a eficiência, desta forma o segundo teste foi 96,4% eficaz.

5. Considerações finais

Devido às aplicações dos *clusters* serem muito diversificadas, sua utilização pode ser indicada para qualquer local onde há problemas computacionais em que a divisão de carga de processamento é uma vantagem.

Com base nisso, este trabalho apresentou uma alternativa para construção de um ambiente de alta disponibilidade, em que foi possível analisar a veracidade do ambiente proposto. Os testes realizados comprovaram que é possível realizar a construção de um *cluster* de computadores utilizando equipamentos de baixo custo e ferramentas de código aberto ou livres.

A utilização do proxy Apt-Cacher-NG como solução de *cache* de pacotes de software para Linux mostrou-se útil no ambiente escolar, pois gera economia de largura da banda WAN, e, mais importante que isso, no contexto educacional, é o tempo

gasto para instalação dos pacotes. Um exemplo simples deste benefício seria quando um professor solicita aos alunos que instalem determinados pacotes para realização de atividades práticas em sala de aula, algo que poderia levar muito tempo até que todos os computadores realizassem o *download* diretamente da internet, com o *proxy* o tempo de instalação diminui exponencialmente.

Com base nos dados bibliográficos e testes realizados nesta pesquisa, conclui-se que a utilização de um *cluster* na implantação de um ambiente alto disponível onde exista uma alta criticidade nas aplicações correntes, evita a parada de serviços causadas por falhas relacionadas ao *hardware* ou software. Também foi possível verificar que não houve perda de *performance* quando um dos nós do *cluster* foi desativado, o que demonstrou a existência de redundância e eficiência do ambiente proposto.

Referências

ALVES, Atos Ramos. **Administração de Servidores Linux.** 1. ed. Rio de Janeiro: Ciência Moderna, 2013.

ALVES, Maicon Melo. **Sockets Linux.** 1. ed. Rio de Janeiro: Brasport, 2008.

ANICAS, Mitchell. **An Introduction to HAProxy and Load Balancing Concepts**. Disponível em: <https://www.digitalocean.com/community/tutorials/an-introduction-to-haproxy-and-load-balancing-concepts>.Acesso: abril/2018.

BEEKHOF. **Home of the Heartbeat project.** Disponível em: <http://www.linux-ha.org/w/index.phptitle=Main_Page&oldid=80>. Acesso: novembro/2018

BONAN, Adilson Rodrigues. **Linux fundamentos, prática & certificação LPI**. Rio de Janeiro: Alta Books, 2010.

BLOCH, Eduard. **Apt-Cacher-NG User Manual**. Disponível em: <https://www.unix-ag.uni-kl.de/~bloch/acng/html/index.html>.Acesso: abril/2018.

CANALTECH. **Estudo mostra que 83% das empresas executam Linux em seus servidores**. Disponível em: <https://canaltech.com.br/linux/ Estudo-mostra-que-83-das-empresas-executam-Linux-em-seus-servidores/>. Acesso: abril/2018.

CANALTECH. **O que é um RFC.** Disponível em: <https://canaltech.com.br/internet/O-que-e-um-RFC/> Acesso: novembro/2018.

COOPER, Ian; MELVE, Ingrid; TOMLINSON, Gary. **Internet Web Replication and Caching Taxonomy.** Disponível em: <https://tools.ietf.org/html/rfc3040>. Acesso: abril/2018.

DOCKER. **Docker Containerization Unlocks the Potential f or Dev and Ops.** Disponível em: <https://www.docker.com/what-docker>. Acesso: abril/2018.

DOCSFEDORA. **Cap. 1. Introduction to RPM.** Disponível em: <https://docs-old.fedoraproject.org/ro/Fedora_Draft_Documentation/0.1/html/RPM_Guide/ch-intro-rpm.html>. Acesso: abril/2018.

DONGARRA, Jack; FOX, Geoffrey; HWANG, Kai. **Computação em nuvem: Clusters virtuais**. Disponível em: <https://technet.microsoft.com/pt-br/library/jj574501.aspx>. Acesso: abril/2018.

EVANS, Clark; NET, Ingy Döt; OREN, Ben-Kiki. **YAML Ain't Markup Language (YAML™) Version 1.2.**

Disponível em: <http://yaml.org/spec/1.2/spec.html>. Acesso: novembro/2018

FIELDING, Roy T; NOTTINGHAM, Mark; RESCHKE, Julia F. **Hypertext Transfer Protocol (HTTP/1.1): Caching.** Disponível em: <https://tools.ietf.org/html/rfc7234>. Acesso: abril/2018.

FLEURY, Maria Tereza Leme; WERLANG, Sérgio. **Pesquisa aplicada – reflexões sobre conceitos e abordagens metodológicas**. Disponível em: <https://bibliotecadigital.fgv.br/dspace/bitstream/handle/10438/18700/A_pesquisa_aplicada_conceito_e_abordagens_metodol%C3%B3gicas.pdf>. Acesso em: novembro/2018.

HAPROXY. **HAProxy Starter Guide.** Disponível em: <https://www.haproxy.org/download/1.9/doc/intro.txt>. Acesso: abril/2018.

IBM. **Redes Virtuais.** Disponível em: <https://www.ibm.com/supp ort/knowledgecenter/pt-br/POWER8/p8efd/p8efd_powervm_virtual_network_concept.htm> Acesso: novembro/2018.

JARGAS, Aurelio. **Shell Script.** Disponível em: <https://aurelio.net/shell/>. Acesso: novembro/2018.

JACKSON, Paul; LAMETER, Christoph. **CGROUPS**. Disponível em: <https://www.kernel.org/doc/Documentation/cgroup-v1/cgroups.txt>. Acesso: novembro/2018.

KERRISK, Michael. **Linux Programmer's Manual.** Disponível em: <http://man7.org/linux/man-pages/man2/chroot.2.html>. Acesso: novembro/2018.

KOUTOUPIS, Petros. **Everything You Need to Know about Linux Containers, Part II: Working with Linux Containers (LXC).** Disponível

em: <https://www.linuxjournal.com/content/everything-you-need-know-about-linux-containers-part-ii-working-linux-containers-lxc>. Acesso: novembro/2018.

MAHONEY, D. Kevin. **Latdict Latin Dictionary & Grammar Resources**. Disponível em: <http://www.latin-dictionary.net/definition/5776/automatus-automata-automatum>. Acesso: novembro/2018.

MADHAVAPEDDY, Anil; SCOTT, David J. **Unikernels: The Rise of the Virtual Library Operating System.** Disponível em: <http://unikernel.org/files/2014-cacm-unikernels.pdf>. Acesso: novembro/2018

MARTÍNEZ, Felipe Alfredo Quinde; RODRÍGUEZ, Diego Armando Quito. **Balanceo de Carga em Servidores de Correo Electrónico bajo el mismo Dominio.** Disponível em: <http://dspace.uazuay.edu.ec/bitstream/datos/2276/1/05790.pdf>. Acesso em: novembro/2018.

MARKETWATCH. **New Study Shows Linux Firmly Entrenched in the Enterprise.** Disponível em: <https://www.marketwatch.com/story/new-study-shows-linux-firmly-entrenched-in-the-enterprise-2013-08-13>. Acesso: abril/2018.

MONQUEIRO, Julio Cesar Bessa. **Guia do Ubuntu.** Disponível em: <https://www.hardware.com.br/guias/ubuntu/gerenciamento-pacotes-repositorios.html>. Acesso: abril/2018.

MCNAB, Chris. **Avaliação de segurança de redes**. 1. ed. São Paulo: Novatec 2017.

MIOTO, R. C. T.; LIMA, T. C. S. **Procedimentos metodológicos na construção do conhecimento científico:**

a pesquisa bibliográfica. *Ensaio*, Florianópolis, v. 10, n.esp, p. 37-41, 2007.

MORAES, Alexandre Fernandes. **Redes sem fio, instalação, configuração e segurança**. 1. ed. São Paulo: Erica 2010.

NEMETE, Avi.; SNYDER, Garth.; HEIN, Trent R. **Manual completo do Linux guia do administrador**. 2. ed. São Paulo: Pearson 2009.

NORMAS TECNICAS. **IEEE 802.1.** Disponível em: <https://www.normastecnicas.com/ieee/ieee-802-1/> Acesso: novembro/2018.

PETERSSON, Andreas; NILSSON, Martin. **Forwarded HTTP Extension.** Disponível em: <https://tools.ietf.org/html/rfc7239>. Acesso: abril/2018.

PITANGA, Marcos. **Construindo supercomputadores com Linux**. 3. ed. rev. Rio de Janeiro: Brasport, 2008.

REDHAT. **O que é Docker?**. Disponível em: <https://www.redhat.com/pt-br/topics/containers/what-is-docker>. Acesso: abril/2018.

RIODATO, Matteo. **Part.III.System Administration.** Disponível em: <https://www.freebsd.org/doc/handbook/jails.html>. Acesso: novembro/2018.

ROSAS, Fátima Weber. **Introdução a redes virtuais.** Disponível em: <http://penta.ufrgs.br/Fatima/rv/rv1.html>. Acesso: novembro/2018.

SOUSA, Lindeberg Barros de. **TCP/IP e Conectividade de Redes - Guia Prático**. 5. ed. São Paulo: Érica, 2009.

SILVA, Wellington Figueira. **Aprendendo Docker**. 1. ed. São Paulo: Novatec 2016.

SILVA, Gustavo Noronha. **Como usar o APT.** Disponível em: <https://www.debian.org /doc/manuals/apt-howto/ch1.pt-br.html>. Acesso: abril/2018.

SOMMERLAD, Peter. **Reverse Proxy Patterns.** Disponível em: <https://www. Research gate.net/publication/221034753_Reverse_Proxy_Patterns>. Acesso: abril/2018.

SHIMEL, Alan. **Brazil Wants To Be The Next India and Open Source Is Their Secret Weapon.** Disponível em: <https://www.networkworld.com /article/2230992/opensource-subnet/brazil-wants-to-be-the-next-india-and-open-source-is-their-secret-weapon.html>. Acesso: abril/2018.

TANENBAUM, Andrew S.; WETHERALL, David. **Redes de computadores**. 5. ed. São Paulo: Pearson 2011.

TELECO. **Redes Etherner I: Recursos de Segregação de Redes.** Disponível em: <http://www.teleco.com.br/tutoriais/tutorialethernetlog1/pagina_3.asp> Acesso: novembro/2018.

TELLES, Diego. **Gerenciamento de Conexão em HTTP/1.x.** Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Gerenciamento_de_Conexão_em_HTTP_1.x>. Acesso: outubro/2018.

WARD, Brian. **Como funciona o Linux, o que todo superusuário deveria saber**. 1. ed. São Paulo: Novatec 2015.

APÊNDICE

```
#!/bin/bash
NG_CACHE="http://172.16.38.5:3148"
FILE_CACHE="/etc/apt/apt.conf.d/00proxy"
PKG=(
"0ad"
)
clean_cache(){
 apt-get clean
}

install_pkg(){
echo "apt-get update" && date
time apt-get -qq update
 for n in ${PKG[@]}; do
  LOG="log$n${HOSTNAME}.txt"
  echo -e "Baixando ${n}\n" && date
  date >> ${LOG} && apt-get -o Debug::Acquire::http=true install -y -d  ${n} >> ${LOG}
  date && echo "Completo ${n}"
done
}

set_proxy(){
 test -e ${FILE_CACHE}
 test $? -eq "1" &&  rm -rf ${FILE_CACHE}
 test $? -eq "0" && touch ${FILE_CACHE}
 echo -e "Acquire::http::Proxy \"${NG_CACHE}\";" >> ${FILE_CACHE}
}

unset_proxy(){
 test -e ${FILE_CACHE} && rm -rf ${FILE_CACHE}
}

show_time_logs(){
 for n in ${PKG[@]}; do
  LOG="log$n${HOSTNAME}.txt"
  egrep "(Baixados|Fetched)" ${LOG} | awk '{print "Tamanho: " $2"-"$3" ""Tempo: " $5"-"$6" ""Vazao: "
$7$8$9$10}' >> limpo${n}.txt
done
}

iniciar(){
 echo "TESTE SEM USO DO CACHE"
 echo "LIMPANDO CACHE LOCAL"
  clean_cache
  unset_proxy
 echo "TESTE DOWNLOAD PACOTES" && date
  time install_pkg
 echo "LIMPANDO CACHE LOCAL"
  clean_cache
 echo "TESTE COM USO DO CACHE" && date
  set_proxy
  time install_pkg
 echo "REMOVENDO CACHE"
  time unset_proxy
 echo "AJUSTANDO DADOS COLETADOS"
  show_time_logs
}

iniciar
```

Estudo de Vulnerabilidades do IPv4 e IPv6

Maicon Rosa da Cunha[1], Valdir Cadorin Onório[1], Guilherme Klein da Silva Bitencourt[2]

[1]Acadêmicos do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88960-000 – Sombrio – SC – Brasil

[2]Docente do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88960-000 – Sombrio – SC – Brasil

{maiconrosadacunha,valdircadorin}@gmail.com,

guilherme.bitencourt@ifc.edu.br

Abstract: *Computer networks are indispensable today, and the new protocol, IPv6, is slowly being adopted. Although IPv6 has native support for some security tools and techniques, it is not completely secure. The objective of this work was to present the protocol failures and to highlight the similarities of the attacks available in IPV4 and IPV6. A virtual laboratory was implemented to perform the intrusion tests using real techniques and tools. The attacks were replicated on IPv4 networks and then on IPv6 networks. The attacks were scanning, Denial of Service and Man In The Middle. The results demonstrate that some attacks available on IPv4 can be replicated over IPv6 networks. IPv6 needs attention to security and we must be aware of the new vulnerabilities and exploit methods that may arise.*

Resumo: *As redes de computadores são indispensáveis atualmente, e o novo protocolo, o IPv6, está sendo lentamente adotado. Apesar do IPv6 possuir suporte*

nativo para algumas ferramentas e técnicas de segurança, não é totalmente seguro. O objetivo deste trabalho foi apresentar as falhas dos protocolos e evidenciar as similaridades do ataques disponíveis no IPV4 e IPV6. Foi implantado um laboratório virtual, para a realização dos testes de intrusão utilizando técnicas e ferramentas reais. Os ataques foram reproduzidos em redes IPv4 e depois em redes IPv6. Os ataques foram do tipo scanning, Denial of Service e Man In The Middle. Os resultados demonstram que alguns ataques disponíveis no IPv4 podem ser reproduzidos em redes IPv6. O IPv6 necessita de atenção à segurança e é preciso estar atento às novas vulnerabilidades e métodos de exploração que podem surgir.

1 Introdução

Os serviços disponibilizados pelas redes de computadores são ferramentas indispensáveis para diversas áreas da sociedade: financeiras, governamentais, científicas e pessoais. Devido essa intervenção na vida comum, tornaram-se alvo de indivíduos com objetivos contrários às condutas éticas e morais, sendo vítimas de roubos constantes de informações confidenciais.

O protocolo IPv4, responsável pela interligação dos dispositivos e redes atuais, possui muitas falhas em sua construção no que diz respeito à segurança das informações trafegadas, devido não ter sido inicialmente desenvolvido com estes aspectos (TANENBAUM; WETHERALL, 2011). Dessa maneira, uma nova versão foi desenvolvida, o IPv6, com serviços de segurança nativo em sua implementação, além de ser mais flexível à qualidade de serviço e estender o endereçamento da versão anterior, que tornou-se aquém do que se deseja no cenário atual.

Entretanto, alguns ataques possíveis no IPv4 foram portados para o IPv6 sem diferenças significativas em sua construção e implementação, criando a falsa ilusão de que o IPv6 é seguro apenas por possuir suporte nativo a algumas técnicas de segurança (SILVESTRE, 2016). Assim, justifica-se a produção deste artigo evidenciar os problemas de segurança em redes IPv6.

O objetivo geral deste trabalho consiste em apresentar as falhas dos protocolos e evidenciar as similaridades dos ataques disponíveis no IPv4 e IPv6. Para alcançar o objetivo, os ataques foram realizados em uma rede IPv4 e uma IPv6 em laboratório virtual controlado, utilizando sistemas operacionais GNU/Linux e ferramentas de exploração e monitoramento para ambas as redes.

O artigo está assim dividido: no item 2 a definição do protocolo IP e suas versões abordadas; no item 3 os conceitos de segurança em uma rede de computadores e a caracterização dos tipos de ataques disponíveis; no item 4 a descrição dos materiais e métodos utilizados; no item 5 a apresentação dos resultados; no item 6 as possíveis soluções; e no item 7 as discussões finais deste trabalho.

2 IP (*Internet Protocol*)

Padronizado no final da década de 1970, sua função principal é a conexão dispositivo-a-dispositivo, através do endereçamento de suas respectivas interfaces, a fim de conectar sistemas de comunicação de computadores (RFC 791).

Este protocolo, posteriormente chamado de IPv4, não foi desenvolvido com funções de segurança, pois seu uso era restrito às universidades que faziam parte da então ARPANET (*Advanced Research Projects Agency Network*) (HAGEN, 2014).

Conforme Tanenbaum e Wetherall (2011), previu-se que o esgotamento dos endereços IPv4 disponíveis ocorreria em alguns anos e os potenciais riscos de segurança e a privacidade, que os usuários estavam expostos, começaram a ser discutidos (RFC 1636).

As soluções de NAT (*Network Address Translation*), CIDR (*Classless Inter Domain Routing*) e o desenvolvimento de técnicas e mecanismos de segurança e privacidade extenderam sua vida útil (COMER, 2006). Contudo, com o crescimento constante de usuários, o advento da mobilidade, dos *smartphones*, *wearables* e conceitos de IoT (*Internet of Things*), o IPv4 tornou-se aquém do que se deseja, surgindo a necessidade da implantação de uma nova versão, o IPv6.

Segundo Davies (2012) afirma, a versão 6 foi projetada para lidar com as exigências de segurança, mobilidade e serviços que o cenário atual exige, descartando as limitações do antecessor, estendendo seu endereçamento, sua escalabilidade e flexibilidade. Além de adicionar funções, através da inclusão de cabeçalhos de extensão, tornando-o adaptativo a diferentes cenários.

2.1 IPv4

No IPv4, os endereços possuem 32 bits, divididos em octetos, escritos em decimal separados por "." (TANENBAUM; WETHERALL, 2011). O endereço 201.32.47.200 é um exemplo. Estes endereços são definidos em três tipos distintos: *unicast*, *multicast* e *broadcast*.

Um pacote com o endereço de destino de *broadcast* é enviado a todos os dispositivos do segmento de rede, todos receberão e processarão o pacote (TANENBAUM; WETHERALL, 2011). É utilizado para a identificação da rede

dos dispositivos participantes, sendo que a própria rede é um domínio de *broadcast*.

Os *multicasts* são agrupamentos de diversos dispositivos, e quando um pacote é destinado a um deles, todos os participantes do grupo receberão e processarão o pacote (DAVIES, 2012). Funciona por meio de *broadcast,* todos recebem o pacote, mas nem todos o processam, somente aqueles que participam do grupo.

Os endereços *unicast* são endereços designados a uma única interface e são classificados em *global unicast*, *link-local unicast*, e *loopback. Os* endereços *global unicast* são os endereços distribuídos aos dispositivos finais: computadores, notebooks, smartphones e outros para serem roteados globalmente (RFC 5735).

Os *link-local unicast* são endereços para serem usados em uma comunicação local simples ou ponto-a-ponto, e no IPv4 são utilizados apenas para atribuir endereços IP às interfaces de rede quando não existe um mecanismo de configuração externo como o DHCP (RFC 5735).

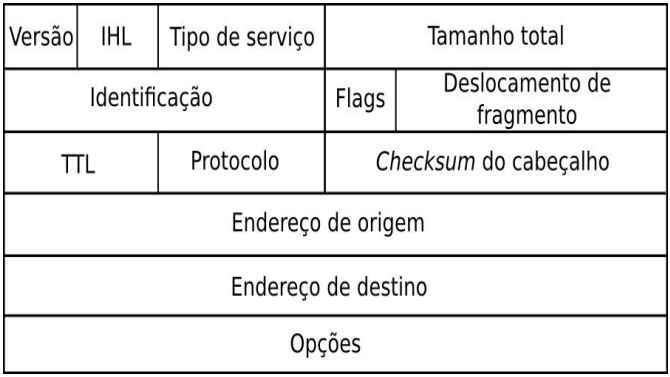
O endereço de *loopback* (i.e. 127.0.0.1) designa o próprio dispositivo (RFC 5735).

2.1.1 Cabeçalho IPv4

O cabeçalho IPv4 possui tamanho variável, seguido da unidade do protocolo superior, que pode ser um segmento TCP (*Transmission Control Protocol*), datagrama UDP (*User Datagram Protocol*) ou mensagem ICMP (*Internet Control Message Protocol*).

A Figura 1 ilustra o cabeçalho IPv4, e seus campos são explicados brevemente adiante.

Figura 1 - Cabeçalho IPv4.



Fonte: Adaptado de Tanenbaum e Wetherall (2011).

a) Versão: Campo que define a versão do protocolo IP utilizado, neste caso leva o valor 0x4 (RFC 791);
b) IHL (*Internet Header Lenght*): Informa o tamanho total em bytes do cabeçalho (RFC 791);
c) Tipo de Serviço: Utilizado para distinguir diferentes tipos de pacotes IPv4 por classes e prioridades (RFC 791);
d) Tamanho Total: Informa o tamanho total do pacote, incluindo cabeçalho e dados (RFC 791);
e) Identificação: Permite que o dispositivo de destino identifique a qual pacote o fragmento pertence (RFC 791);
f) *Flag* DF (*Don't Fragment*): Não permite que os roteadores fragmentem o pacote e são utilizados para descobrir o MTU (*Maximum Trasmission Unit*) do caminho (RFC 791);
g) *Flag* MF (*More Fragments*): Identifica se o fragmento pertence ou não a um pacote fragmentado anteriormente (RFC 791);

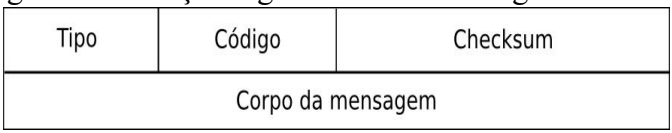
h) Deslocamento de Fragmento: Informa a qual ponto do pacote original o fragmento pertence (RFC 791);
i) TTL (*Time To Live*): Limita o número de saltos, isto é, o total de roteadores que um pacote IPv4 pode atravessar, evitando *loops* infinitos (RFC 791);
j) Protocolo: Identifica o próximo cabeçalho, da camada superior (RFC 791);
k) *Checksum* do Cabeçalho: Código que verifica a integridade do cabeçalho (RFC 791);
l) Endereço de Origem: Identifica o endereço de origem do pacote (RFC 791);
m) Endereço de Destino: Identifica o endereço de destino do pacote (RFC 791);
n) Opções: De tamanho variável, permite dados adicionais no tratamento do pacote (RFC 791).

2.1.2 ICMPv4

O ICMP é um protocolo utilizado para testes de rede simples e para tratamento de erros em redes IPv4 (TANENBAUM; WETHERALL, 2011). É o responsável por avisar um dispositivo quando a rede solicitada é inexistente por exemplo. Devido suas diferenças com o ICMPv6, ele é chamado de ICMPv4.

A Figura 2 ilustra a estrutura da mensagem ICMPv4, e seus campos são explicados brevemente adiante.

Figura 2 - Cabeçalho geral de uma mensagem ICMPv4.



Fonte: Adaptado da RFC 792.

a) Tipo: Especifica o tipo de mensagem. Identificado por um número (0-255) (RFC 792);

b) Código: Tipo específico da mensagem (RFC 792);
c) *Checksum*: Utilizado para identificar a integridade da mensagem (RFC 792);
d) Corpo da mensagem: Dados específicos ao tipo de mensagem (RFC 792).

Os tipos de mensagem ICMPv4 são divididos em 2 classes: tratamento de erros e informacionais. Existem muitas mensagens ICMPv4, sendo que a sua maioria não são mais utilizadas, por isso serão apresentadas apenas as principais.

As mensagens informacionais são: *Echo Request* (tipo 8) e *Echo Reply* (tipo 0), utilizados pelo *ping* para diagnósticos de rede simples.

As mensagens de erro são: Destino Inalcançável (tipo 3); Tempo Excedido (tipo 11) e Problema de Parâmetro (tipo 12).

2.1.3 ARP (*Address Resolution Protocol*)

O ARP é utilizado na resolução de endereço lógico (IP) para endereço físico (MAC - *Media Access Control*) (TANENBAUM; WETHERALL, 2011).

Um cenário simples, para ilustrar seu funcionamento, é quando um dispositivo quer saber qual o endereço MAC do portador do endereço IPv4 192.168.0.1. O dispositivo envia um pacote *ARP Request* para o *broadcast* da rede fazendo a solicitação. Dessa maneira, todos recebem, mas somente o responsável pelo endereço IP alvo responde com um pacote *ARP Reply*, informando o seu endereço MAC para que o dispositivo solicitante atualize sua tabela de vizinhos para não ter que perguntar novamente (TANENBAUM; WETHERALL, 2011).

2.2 IPv6

No IPv6, os endereços de 128 bits são divididos em grupos de 16 bits separadas por “ : ”, sendo cada grupo formado por 4 dígitos hexadecimais (DAVIES, 2012). O endereço 2001:0db8::9c5a é um exemplo. Estes endereços são definidos em três tipos distintos: *unicast*, *anycast* e *multicast* (HAGEN, 2014).

Os endereços *anycast* são designados para múltiplas interfaces, e foram criados para produzirem redundância e balanceamento de carga em locais onde diversos dispositivos ou roteadores hospedam os mesmos serviços (HAGEN, 2014). Um pacote enviado para um endereço *anycast* é entregue a uma única interface, a mais próxima (DAVIES, 2012).

Os endereços *multicast* são análogos aos disponíveis no IPv4, entretanto, dispositivos IPv6 podem possuir múltiplos endereços *multicast*, e são substitutos do *broadcast,* pois viu-se que não eram mais necessários no IPv6 (HAGEN, 2014).

Os endereços *unicast* no IPv6 são classificados em *global unicast*, *link-local unicast*, *unique-local address*, *loopback*, *unspecified address* e *transition address* (HAGEN, 2014). Os endereços *global unicast* e *loopback* (i.e. ::1) são análogos aos disponíveis no IPv4.

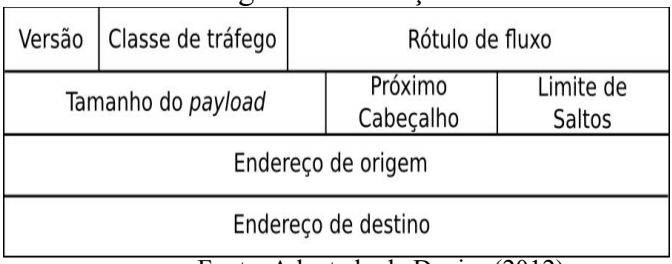
Todos os dispositivos que implementam o IPv6 são obrigados a ter, no mínimo, um *link-local unicast* atribuído para comunicação local (RFC 2460), além de serem utilizados nos mecanismos de autoconfiguração do IPv6 (HAGEN, 2014). O *unique-local address* é um endereço global único, não deve ser roteado e é designado às redes corporativas específicas ou redes especiais confinadas (HAGEN, 2014). Os *transition addresses* são endereços designados às tecnologias de transição IPv4 para IPv (HAGEN, 2014).

2.2.1 Cabeçalho IPv6 base

O cabeçalho IPv6 possui tamanho fixo de 40 bytes, seguido por zero ou mais cabeçalhos de extensão de tamanhos variados, seguido da unidade do protocolo superior, que pode ser um segmento TCP, datagrama UDP ou mensagem ICMPv6.

A Figura 3 ilustra o cabeçalho IPv6, e seus campos são explicados brevemente adiante.

Figura 3 - Cabeçalho IPv6.



Fonte: Adaptado de Davies (2012).

a) Versão: Campo que define a versão do protocolo IP utilizado, neste caso leva o valor 0x6 (RFC 2460);
b) Classe de Tráfego: Análogo ao campo Tipo de Serviço do IPv4 (RFC 2460);
c) Rótulo de Fluxo: Identifica o pacote como pertencente a uma sequência de pacotes entre origem e destino que precisam de tratamento especial por dispositivos intermediários (RFC 2460);
d) Tamanho do *Payload:* Especifica o tamanho dos dados que seguem o cabeçalho, incluindo as extensões (RFC 2460);
e) Próximo Cabeçalho: Análogo ao campo Protocolo do IPv4, mas se houver cabeçalhos de extensão no pacote, este campo identifica a extensão, que por sua vez identifica o protocolo superior (RFC 2460);
f) Limite de Saltos: Análogo ao campo TTL do IPv4 (RFC 2460);

g) Endereço de Origem: Identifica o endereço de origem do pacote (RFC 2460);
h) Endereço de Destino: Identifica o endereço de destino do pacote (RFC 2460).

2.2.2 Cabeçalhos de Extensão

Conforme Hagen (2014) e Davies (2012), as especificações atuais do protocolo IPv6 definem 6 cabeçalhos de extensão, suas características estão fora do escopo deste trabalho:

a) *Hop-by-Hop Options*: Carrega informações que devem ser processadas por cada dispositivo ao longo do caminho, e deve vir imediatamente após o cabeçalho base;
b) *Routing Header*: Utilizado para predeterminar a rota dos pacotes ao longo do caminho;
c) *Fragment Header*: Quando um pacote IPv6 é fragmentado, cada fragmento leva um *fragment header* que indica a qual pacote original ele pertence, para ser remontado no destino com sucesso;
d) *Destination Options*: Carrega informações opcionais que só serão processadas unicamente pelo destino;
e) *Autentication Header*: Faz parte do IPSec (*IP Security*) e é responsável pela autenticidade e integridade do pacote IPv6, mas não sua confidencialidade;
f) *Encapsulating Security Payload*: Também faz parte do IPSec, sendo responsável pela confidencialidade dos dados, além das mesmas funções do campo anterior.

2.2.3 ICMPv6

O ICMPv6 é uma versão melhorada do ICMP utilizado no IPv4, e deve ser implementado obrigatoriamente junto com o IPv6 (RFC 4443).

Além das funções de diagnóstico de rede, presentes no ICMPv4, ele incorpora as funções do IGMP (*Internet Group Management Protocol*) e do ARP, ambos presentes no IPv4 (HAGEN, 2014). Ou seja, as funções de gerenciamento de grupos *multicast* e descobrimentos de vizinhos no IPv6, utilizam mensagens específicas do ICMPv6.

A estrutura da mensagem ICMPv6 é idêntica a versão anterior, as diferenças são os tipos de mensagem que também são divididas em 2 classes: tratamento de erros e informacionais.

As mensagens informacionais são: *Echo Request* (tipo 128) e *Echo Reply* (tipo 129), utilizados pelo *ping* para diagnósticos de rede simples.

As mensagens de erro são: Destino Inalcançável (tipo 1); Pacote Muito Grande (tipo 2); Tempo Excedido (tipo 3) e Problema de Parâmetro (tipo 4).

2.2.4 NDP (*Neighbor Discovery Protocol*)

O NDP combina as funcionalidades presentes no ARP com as mensagens ICMP *Router Discovery* e *Redirect* disponíveis no IPv4 (HAGEN, 2014). Suas funcionalidades são utilizadas por dispositivos finais e roteadores. Para Davies (2012), o NDP é utilizado para os seguintes propósitos:

a) Encontrar roteadores vizinhos, mesmo procedimento do ICMP *Router Discovery* no IPv4;
b) Descobrir características e parâmetros de rede;
c) Autoconfiguração, *stateless* e *statefull;*

d) Resolução de endereço físico, mesmo procedimento do ARP no IPv4;
e) Determinar se um dispositivo ainda é alcançável (NUD – *Neighbor Unrechability Detection*);
f) Detecção de endereço duplicado (DAD – *Duplicate Address Detection*);
g) Função *Redirect*, mesmo processo do ICMP *Redirect* no IPv4.

O NDP utiliza 5 mensagens ICMPv6, do tipo informacionais, para suas funções. De acordo com a RFC 4861, são elas: *Router Solicitation* (RS, tipo 133); *Router Advertisement* (RA, tipo 134); *Neighbor Solicitation* (NS, tipo 135); *Neighbor Advertisement* (NA, tipo 136) e *Redirect* (tipo 137).

No processo de descobrimento de roteadores, mensagens RS são utilizadas por dispositivos para determinar características de rede e rotas de comunicação. Mensagens RA são utilizadas por roteadores para responder às mensagens RS.

No processo de resolução de endereços físicos são utilizadas mensagens NS. Mensagens NA são respostas às NS. Mensagens NS/NA também são utilizadas no processo NUD e DAD.

Mensagens *Redirect* são utilizadas, por roteadores, para avisar um dispositivo que existe um caminho melhor para um determinado pacote IPv6 enviado anteriormente.

3 Segurança em Redes

A segurança de uma rede de computadores não era prioridade no início de sua concepção, pois era controlada por uma pequena quantidade de indivíduos. Quando tornou-se pública, ela foi adaptando-se às atividades cotidianas das pessoas, organizações e governos.

Dessa maneira, surgiram várias formas de ataque às redes, na tentativa de roubo de informações confidenciais, através de operações simples a complexas, pequeno e grande porte, capazes de derrubar redes inteiras (SANTOS, 2007).

Assim, técnicas e mecanismos foram desenvolvidos para prover autenticidade, confidencialidade, não-repúdio e integridade às informações, utilizando meios digitais como *firewalls*, IDS (*Intrusion Detection System*), IPS (*Intrusion Protection System*), criptografia e meios físicos, como câmeras de segurança, portas, sensores de presença e outros (KIZZA, 2015).

Santos (2007) afirma que as Políticas de Segurança atuais devem abranger desde os softwares utilizados até a infraestrutura física das instalações.

3.1 Tipos de Ataques

Os ataques são classificados conforme sua extensão, construção e danos causados.

3.1.1 *Scanning*

Consiste em descobrir quantos dispositivos estão ativos na rede, quais seus endereços de IP, MAC, quais portas TCP estão abertas e outras informações (McCLURE; SCAMBRAY; KURTZ, 2009). Após a listagem dos dispositivos, o atacante inicia a execução de aplicativos específicos que buscam brechas de segurança que possam ser exploradas pelos ataques (McCLURE; SCAMBRAY; KURTZ, 2009).

3.1.2 DoS - *Denial of Service*

São ataques em que um dispositivo atacante interrompe determinado serviço ou comunicação do alvo, partindo de um ou mais dispositivos distribuídos, sendo, este último, mais perigoso

(McCLURE; SCAMBRAY; KURTZ, 2009). É mais conhecido por ocorrer através da sobrecarga de rede ou processamento do alvo, deixando de responder, e técnicas mais simples como mudar a rota padrão do alvo para um lugar que não existe, perdendo a comunicação com a rede (SANTOS, 2007).

3.1.3 MiTM - *Man in The Middle*

Na sua forma básica, é quando um atacante fica entre a comunicação de dois ou mais dispositivos comuns capturando seus dados de forma transparente (McCLURE; SCAMBRAY; KURTZ, 2009). Este ataque ocorre de duas formas, quando o atacante se passa por um usuário legítimo ou quando o atacante se passa por um roteador legítimo.

Nos dois casos, o atacante deve encaminhar a comunicação aos seus respectivos destinos, senão o ataque torna-se DoS e sua descoberta mais fácil.

4 Materiais e Métodos

A metodologia empregada é classificada como pesquisa aplicada que, conforme Marconi e Lakatos (2012), é uma investigação original concebida pelo interesse em novos conhecimentos, primordialmente dirigida em função de um objetivo específico.

Corroborando com as autoras supra citadas, Gil (2010) descreve a pesquisa aplicada como sendo voltada à aquisição de conhecimentos com o objetivo de aplicá-la em situações específicas.

Os materiais e os procedimentos da pesquisa propostos foram dois computadores que simulam usuários comuns e um computador que simula o atacante. Eles são conectados às redes IPv4 e IPv6, internas, controladas para a realização dos testes, além de possuírem conexão com a Internet externa para a instalação das ferramentas.

Este laboratório foi implementado em uma máquina física de propriedade de um dos autores, com sistema operacional Debian GNU/Linux na versão 8.2 com ambiente gráfico Gnome na versão 3.14.1. Todas as máquinas virtuais possuem sistema operacional Debian GNU/Linux versão 9.1.0 com ambiente gráfico XFCE versão 4.12.

Estas máquinas possuem *hostnames* próprios para identificação. São eles: os usuários, *host-AAA-usuario* e *host-123-usuario;* e o atacante, *host-FFF-atacante.*

Os endereços IPv4 e IPv6 foram configurados estaticamente conforme a RFC 3849 e a RFC 5737, específicos para serem utilizados em documentações. Os endereços de MAC foram configurados seguindo um padrão próprio dos autores para melhor visualização dos resultados.

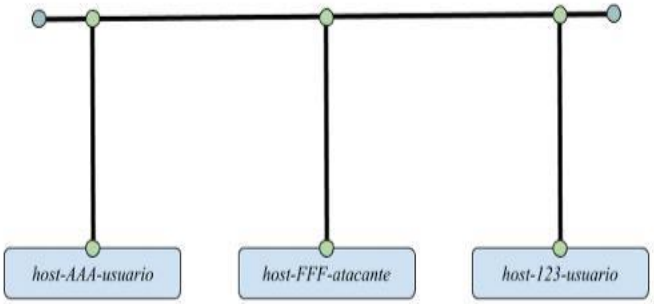
Segue abaixo as configurações utilizadas nas máquinas virtuais:

- *host-AAA-usuario*:
  - Interface *enp0s3*: configurada em NAT para conexão com a internet externa para instalação das ferramentas;
  - Interface *enp0s8*: endereço IPv4 de número 192.0.2.100/24, e endereço MAC de número 08:00:27:44:AA:BC;
  - Interface *enp0s9*: endereço IPv6 de número 2001:DB8::AABC/32, e endereço MAC de número 08:00:27:66:AA:BC;
  - *host-123-usuario*:
  - Interface *enp0s3*: NAT;
  - Interface *enp0s8*: IPv4 192.0.2.200/24, e MAC 08:00:27:44:AA:BC;
  - Interface *enp0s9*: IPv6 2001:DB8::1123/32, e MAC 08:00:27:66:11:23;

- *host-FFF-atacante*:
  - Interface *enp0s3*: NAT;
  - Interface *enp0s8*: IPv4 192.0.2.254/24, e MAC 08:00:27:44:FF:FF;
  - Interface *enp0s9*: IPv6 2001:DB8::FFFF/32, e MAC 08:00:27:66:FF:FF;

A Figura 4 ilustra a topologia lógica do laboratório.

Figura 4 - Topologia lógica do laboratório de testes.



Fonte: Elaborado pelos autores, 2017.

As ferramentas utilizadas pelo *host-FFF-atacante* para os ataques realizados ao IPv4 foram três: o Nmap versão 7.40, para o ataque de *scanning*; o arpoison versão 0.7, para o ataque de DoS; e o arpspoof versão 2.4, para o ataque de MiTM. Estas ferramentas funcionam por meio de comandos no terminal e estão disponíveis para outros sistemas operacionais além do GNU/Linux.

Por outro lado, no IPv6 foi utilizado uma ferramenta única, o Thc-ipv6 versão 3.2. Consiste de uma coleção de aplicativos que produzem diversos tipos de ataques e escaneamentos em redes IPv6. Disponível somente para sistemas operacionais GNU/Linux, com funcionamento por comandos no terminal.

Para a captura dos pacotes nas duas redes, IPv4 e IPv6, foi instalado no *host-FFF-atacante* o Wireshark versão 2.2.6. Além de possuir interface gráfica amigável, é capaz de capturar dados de diversos tipos de protocolos além do IP e organizar sua estrutura graficamente, facilitando sua análise.

4.1 Métodos de exploração

Os ataques serão divididos em três partes: na primeira parte foi executado o ataque de *scanning;* na segunda o de DoS; e na terceira o de MiTM. Em cada parte, eles foram executados primeiro na rede IPv4 e, em seguida, na rede IPv6. Os métodos descritos a seguir ocorrem em ambas as redes.

4.1.1 Método para o ataque de *scanning*

O *host-FFF-atacante* executará o aplicativo que irá escanear a rede e retornar como resultado os dispositivos que estão conectados na rede.

4.1.2 Método para o ataque de DoS

O *host-AAA-usuario* receberá uma mensagem do *host-123-usuario* pedindo para ele atualizar sua tabela de vizinhos com seu novo endereço MAC. Porém, este MAC é falso, forjado pelo *host-FFF-atacante*, e não representa nenhum endereço, fazendo com que o *host-AAA-usuario* perca comunicação com o *host-123-usuario*.

4.1.3 Método para o ataque de MiTM

O *host-FFF-atacante* executará o aplicativo que irá sobrescrever a tabela de vizinhança no *host-AAA-usuario* com uma nova entrada que aponta para o *host-FFF-atacante*, dizendo que foi o *host- 123-usuario* quem enviou a mensagem. O mesmo método também será feito no *host-123-usuario*. Porém, para os dispositivos, eles estarão enviando para seus vizinhos legítimos

quando, na verdade, estarão passando pelo *host-FFF-atacante* que ficará entre sua comunicação.

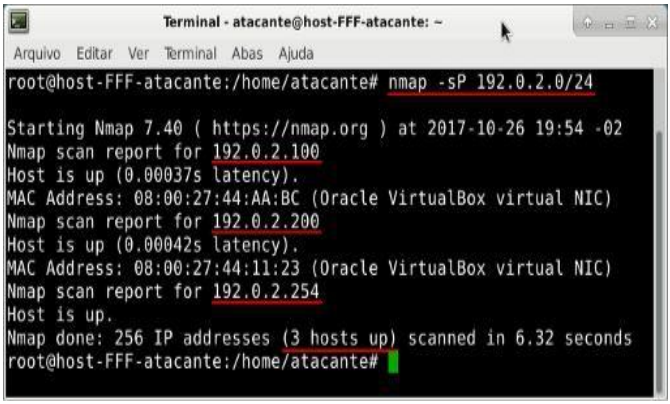
5 Resultados

Todos os comandos utilizados para obtenção dos resultados foram executados em modo terminal como usuário *root*.

5.1 *Scanning* em rede IPv4

A Figura 5 mostra o *host-FFF-atacante* que executou o aplicativo Nmap com o seguinte comando: # nmap -sP 192.0.2.0/24.

Figura 5 - Ataque *scanning* utilizando o aplicativo Nmap.



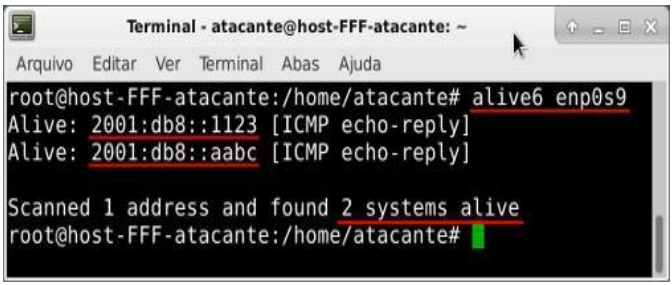
Fonte: Elaborado pelos autores, 2017.

Observa-se que foi possível detectar os endereços IPv4 de todas as três máquinas virtuais do laboratório, incluindo o próprio *host-FFF-atacante* que executou o comando. Na última linha da saída do aplicativo é possível perceber que ele escaneou todos os 256 endereços IPv4 disponíveis dentro da rede e retornou os três que estão ativos.

5.2 *Scanning* em rede IPv6

A Figura 6 mostra o *host-FFF-atacante* que executou o aplicativo alive6 da coleção Thc-ipv6 com o seguinte comando: # alive6 enp0s9.

Figura 6 - Ataque *scanning* utilizando o aplicativo alive6.



Fonte: Elaborado pelos autores, 2017.

Observa-se que foi possível detectar os dois endereços IPv6 das máquinas virtuais do laboratório pertencentes ao *host-123-usuario* e *host-AAA-usuario,* respectivamente, excluindo o *host-FFF-atacante* que executou o comando. O tempo de execução do aplicativo foi de 5 a 6 segundos.

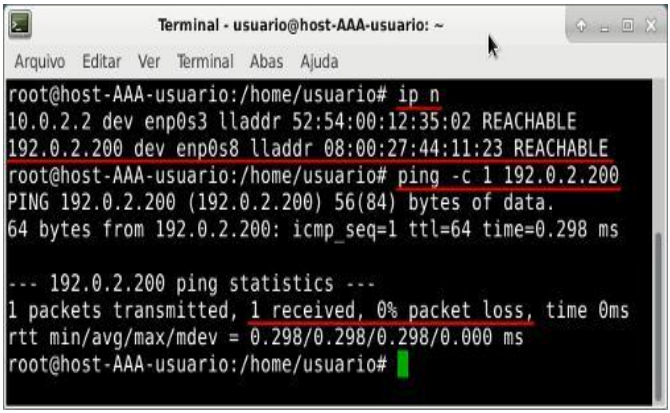
Com este resultado, demonstrou-se que é possível escanear uma rede IPv6 rapidamente, o que diverge das afirmações de Davies (2012), Durdağı e Buldu (2010), e Hagen (2014), que indicam que o escaneamento de uma rede IPv6 é improvável devido aos seus endereços possuírem tamanho extenso, o que implicaria em um longo período de tempo para ser executado com sucesso.

5.3 DoS em rede IPv4

A Figura 7 mostra que o *host- AAA-usuario* possui o MAC verdadeiro do *host-123-usuario*, através da análise da sua tabela ARP com o comando: # ip n. Na Figura 7, abaixo do comando anterior, é possível ver que há comunicação entre eles, através

do comando *ping* que testa se há comunicação entre dois dispositivos.

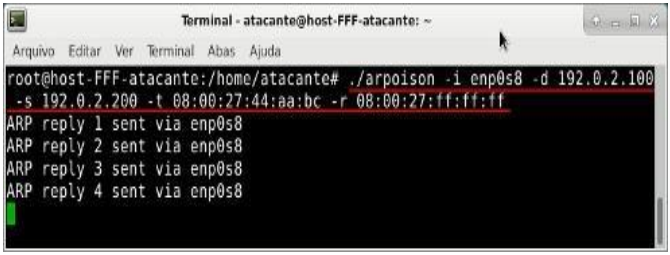
Figura 7 - Tabela ARP de *host-AAA-usuario* sem entradas falsas.



Fonte: Elaborado pelos autores, 2017.

A Figura 8 mostra o *host- FFF-atacante* que executou o aplicativo arpoison com o seguinte comando: # ./arpoison -i enp0s8 -d 192.0.2.100 -s 192.0.2.200 -t 08:00:27:44:aa:bc -r 08:00:27:ff:ff:ff; enviando como MAC falso a sequência 08:00:27:FF:FF:FF.

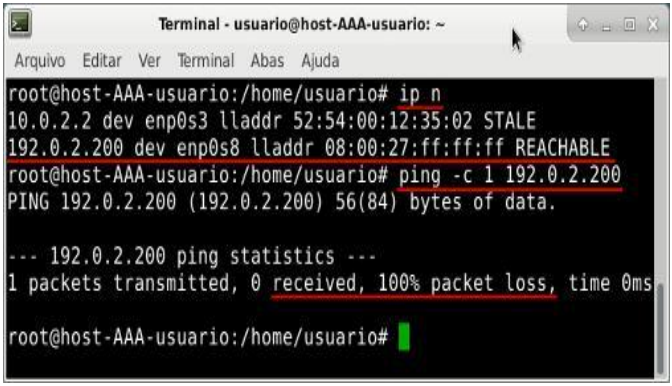
Figura 8 - ARP *poisoning* utilizando o aplicativo arpoison.



Fonte: Elaborado pelos autores, 2017.

A Figura 9 mostra a tabela ARP do *host-AAA-usuario*, através do comando: # ip n; com o novo MAC falso apontando para o IP de *host- 123-usuario*. Ainda na Figura 9, abaixo do comando anterior, é possível perceber que a comunicação com o *host-123-usuario* foi perdida.

Figura 9 - Tabela ARP de *host-AAA-usuario* com a entrada falsa.



Fonte: Elaborado pelos autores, 2017.

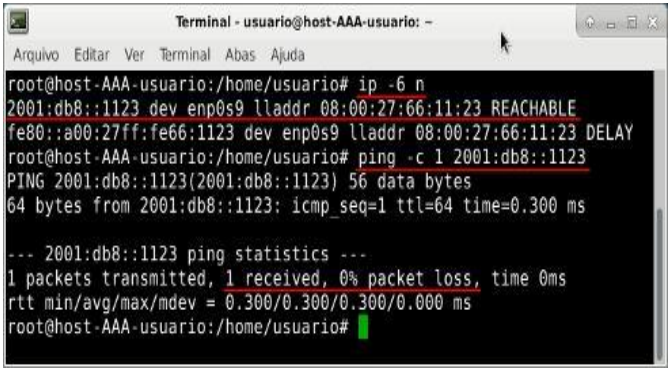
5.4 DoS em rede IPv6

A Figura 10 mostra que o *host-AAA-usuario* possui o MAC verdadeiro do *host- 123-usuario*, através da análise da sua tabela de vizinhos IPv6 com o comando: # ip -6 n. Ainda na Figura 10,

abaixo do comando anterior, é possível ver que há comunicação entre o *host-AAA-usuario* e *host*

-123-usuario através do *ping*.

Figura 10 - Tabela de vizinhos IPv6 sem entradas falsas.

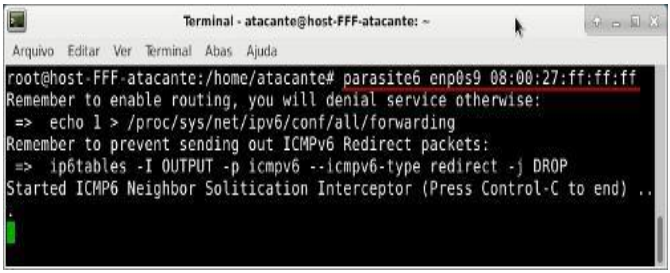


Fonte: Elaborado pelos autores, 2017.

A Figura 11 mostra o *host-FFF-atacante* que executou o aplicativo parasite6 com o seguinte comando: # parasite6 enp0s9 08:00:27:ff:ff:ff; enviando como MAC falso a sequência 08:00:27:FF:FF:FF.

Este comando funciona apenas quando a tabela de vizinhos IPv6 está vazia, devido a forma como o NDP trabalha. Então, neste caso, foram reiniciados o *host-AAA-usuario* e o *host-123-usuario* para que o ataque possa ser realizado.

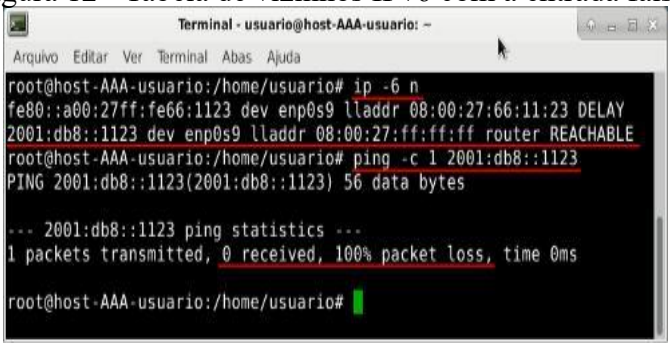
Figura 11 - NDP *poisoning* utilizando o aplicativo parasite6.



Fonte: Elaborado pelos autores, 2017.

A Figura 12 mostra a tabela de vizinhos IPv6 do *host-AAA-usuario*, através do comando: # ip -6 n; com o novo MAC falso apontando para o IP de *host-123-usuario*. Ainda na Figura 12, abaixo do comando anterior, é possível perceber que a comunicação com o *host-123-usuario* foi perdida.

Figura 12 - Tabela de vizinhos IPv6 com a entrada falsa.



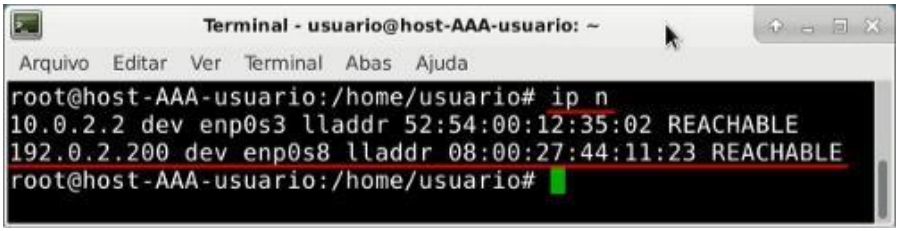
Fonte: Elaborado pelos autores, 2017.

Com este resultado, demonstrou-se que os conceitos de alterar a tabela de vizinhos de um dispositivo que implementa IPv6 mantém-se a mesma para o IPv4, sendo a diferença entre as abordagens, a utilização de ferramentas específicas para cada protocolo.

5.5 MiTM em rede IPv4

A Figura 13 mostra que o *host-AAA-usuario* possui o MAC verdadeiro do *host-123-usuario*, olhando sua tabela ARP com o comando: # ip n.

Figura 13 - Tabela ARP de *host-AAA-usuario* sem entradas falsas.



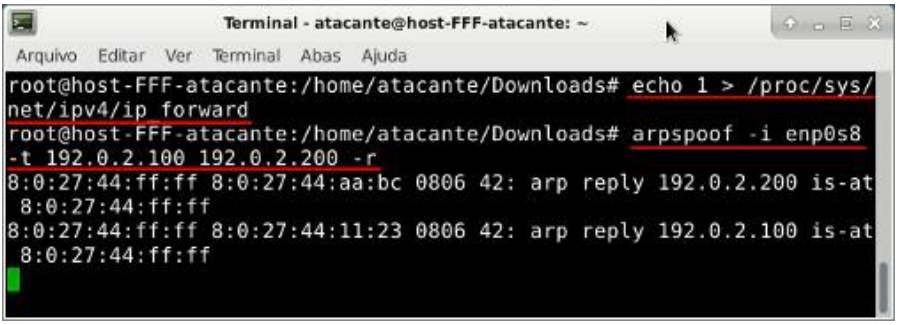
Fonte: Elaborado pelos autores, 2017.

Antes de iniciar o ataque, é necessário que o roteamento de pacotes IPv4 esteja ativado no *host-FFF-atacante* para que a comunicação entre os alvos ocorra, senão o ataque torna-se DoS, devido os pacotes não chegarem ao seu destino original.

A Figura 14 mostra o *host-FFF-atacante* que executou o comando: # echo 1 > /proc/sys/net/ipv4/ip_forward; que ativa o roteamento IPv4. Em seguida, ele executou o aplicativo arpspoof com o seguinte comando: # arpspoof -i enp0s8 -t

192.0.2.100 192.0.2.200 -r.

Figura 14 - ARP *poisoning* utilizando o aplicativo arpspoof.



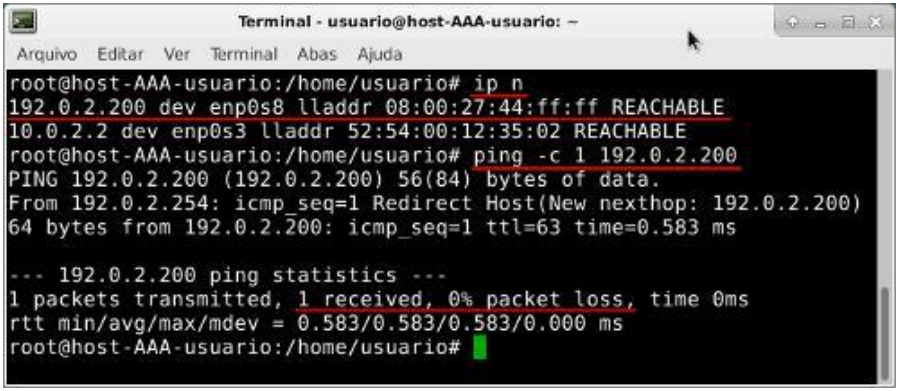
Fonte: Elaborado pelos autores, 2017.

As Figuras 15 e 16 mostram a tabela ARP do *host-AAA-usuario* e *host-123-usuario*, respectivamente, através do comando: # ip n; com o novo MAC falso apontando para o *host-FFF-atacante*.

Nas Figuras 15 e 16, ainda é possível perceber que a comunicação entre *host-AAA-usuario* e *host-123-usuario* permanece ativa.

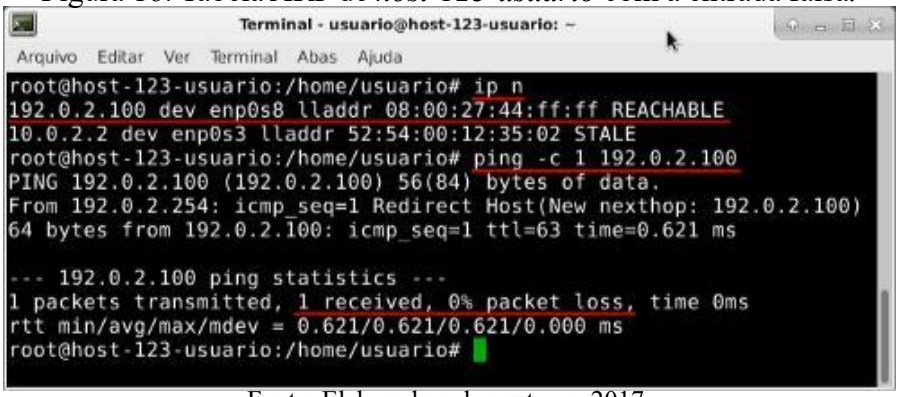
Entretanto, em ambos os casos, a comunicação passa primeiro pelo *host-FFF-atacante,* para então ser entregue ao destino original, sendo passível de captura e modificações dos dados.

Figura 15 - Tabela ARP de *host-AAA-usuario* com a entrada falsa.



Fonte: Elaborado pelos autores, 2017.

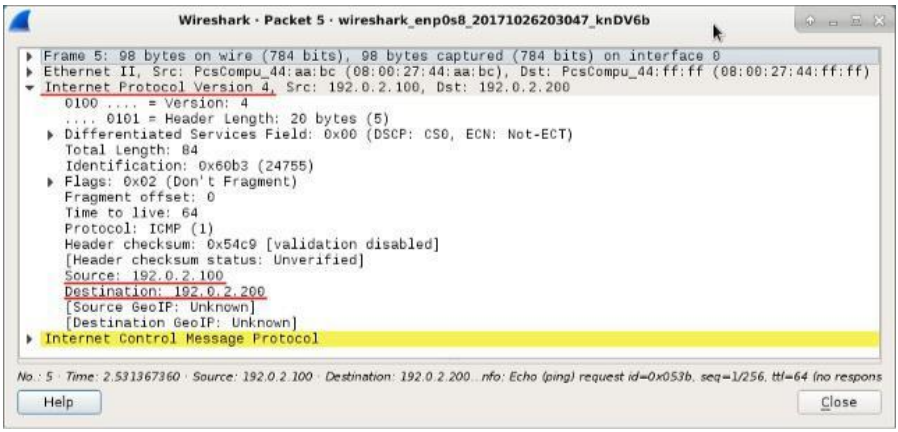
Figura 16: Tabela ARP de *host-123-usuario* com a entrada falsa.



Fonte: Elaborado pelos autores, 2017.

A Figura 17 mostra a interceptação da comunicação entre o *host-AAA-usuario* e o *host-123-usuario* através da captura de um pacote IPv4 pelo *host-FFF-atacante* utilizando o Wireshark, evidenciando os campos de origem e destino que apontam para os dispositivos dos usuários.

Figura 17 - Captura de pacote IPv4 pelo *host-FFF-atacante.*

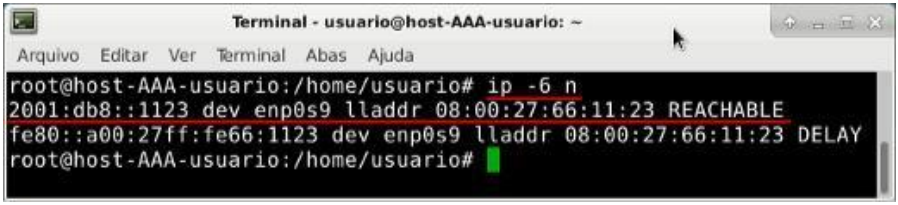


Fonte: Elaborado pelos autores, 2017.

5.6 MiTM em rede IPv6

A Figura 18 mostra que o *host-AAA-usuario* possui o MAC verdadeiro do *host-123-usuario*, olhando sua tabela de vizinhos IPv6 com o comando: # ip -6 n.

Figura 18 - Tabela de vizinhos IPv6 sem entradas falsas.



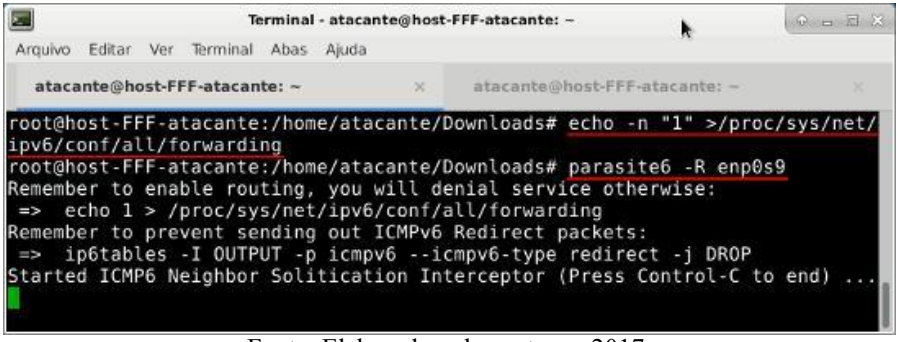
Fonte: Elaborado pelos autores, 2017.

Como no IPv4, antes de iniciar o ataque é necessário que o roteamento de pacotes IPv6 esteja ativado no *host- FFF-atacante*. A Figura 18 mostra o *host-FFF-atacante* que executou o comando:

echo -n “1” > /proc/sys/net/ipv6/conf/all/forwarding; que ativa o roteamento IPv6.

Em seguida, executou o aplicativo parasite6 com o seguinte comando: # parasite6 -R enp0s9. Devido este comando só funcionar quando a tabela de vizinhos IPv6 está vazia, foram reiniciados o *host-AAA-usuario* e o *host123-usuario* para continuar o ataque.

Figura 19 - NDP *poisoning* utilizando o aplicativo parasite6.



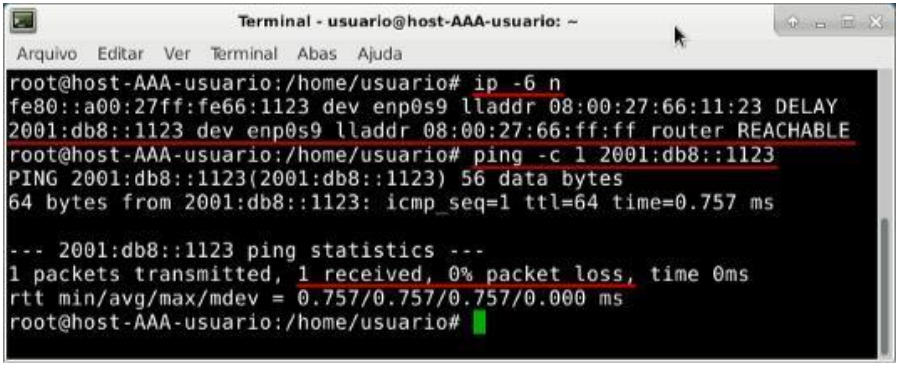
Fonte: Elaborado pelos autores, 2017.

As Figuras 20 e 21 mostram a tabela de vizinhos IPv6 do *host-AAA-usuario* e *host-123-usuario*, respectivamente, através do comando: # ip -6 n; com o novo MAC falso apontando para o *host-FFF-atacante*.

Nas Figuras 20 e 21, ainda é possível perceber que a comunicação entre *host-AAA-usuario* e *host-123-usuario* permanece ativa.

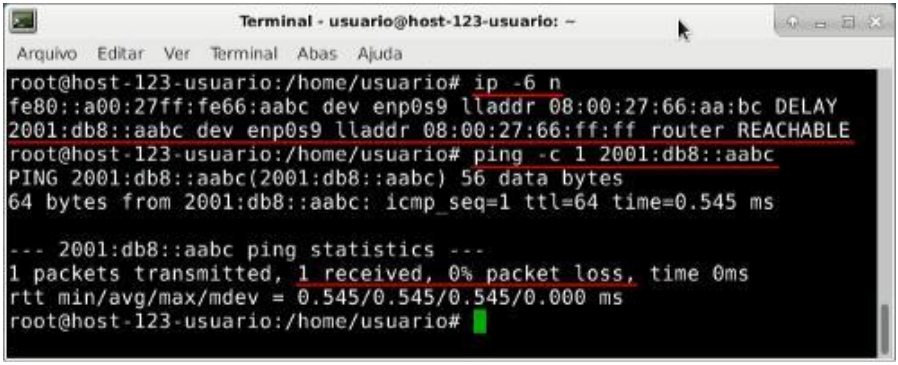
Entretanto, ocorre o mesmo que no IPv4, a comunicação passa primeiro pelo *host-FFF-atacante,* para então ser entregue ao destino original.

Figura 20 - Tabela de vizinhos IPv6 com a entrada falsa.



Fonte: Elaborado pelos autores, 2017.

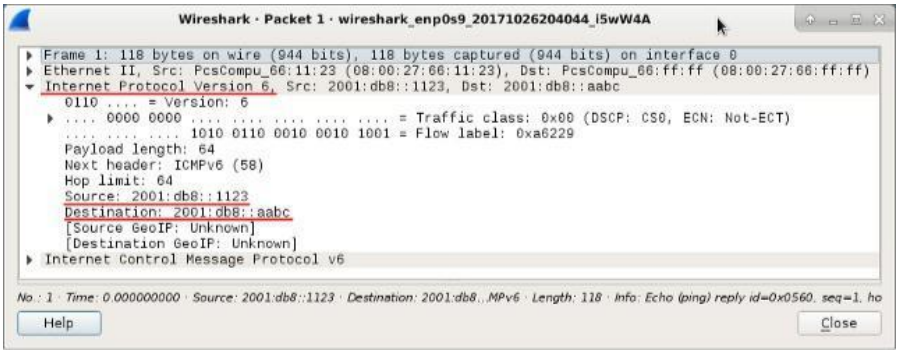
Figura 21 - Tabela de vizinhos IPv6 com a entrada falsa.



Fonte: Elaborado pelos autores, 2017.

A Figura 22 mostra a interceptação da comunicação entre o *host-AAA-usuario* e o *host-123-usuario,* através da captura de um pacote IPv6 pelo *host-FFF-atacante* utilizando o Wireshark, evidenciando os campos de origem e destino que apontam para os dispositivos dos usuários.

Figura 22 - Captura de pacote IPv6 pelo *host-FFF-atacante.*



Fonte: Elaborado pelos autores, 2017.

Com este resultado, demonstrou-se que o *host- FFF-atacante* interceptou a comunicação entre os dois dispositivos tanto na rede IPv4 como na rede IPv6, ficando passível de alterações que podem prejudicar a comunicação da rede como um todo.

A Tabela 1 apresenta a relação entre os dois protocolos, suas similaridades e diferenças na utilização ou configuração das ferramentas.

Tabela 1. Relação dos ataques realizados.

Ataques	IPv4	IPv6	Observações
Scanning	Utilizado o aplicativo Nmap que retornou os dispositivos conectados	Utilizados aplicativos Alive6 que retornou os aplicativos conectados	A única diferença entre os dois é o aplicativo utilizado
DoS	Utilizado o aplicativo Arpoison e	Utilizado o aplicativo parasite6 e	Para o ataque ser efetivo no IPv6 é

	passado como parâmetro um endereço MAC falso	passado como parâmetro um endereço MAC falso	necessário reiniciar os dispositivos devido a forma como o NDP trabalha
MiTM	Utilizado o aplicativo arpspoof e passado como parâmetro o endereço MAC do atacante.	Utilizado o aplicativo parasite6 e passado como parâmetro o endereço MAC do atacante	Para o ataque ser efetivo é necessário que o roteamento de pacotes IPv4 e/ou IPv6 esteja habilitado no atacante senão este ataque torna-se DoS.

6 Possíveis soluções

Apesar de não estar dentro do escopo deste trabalho, é necessário evidenciar que existem técnicas e ferramentas disponíveis atualmente para solucionar e evitar estas vulnerabilidades.

Nos trabalhos de Durdağı e Buldu (2010), Hagen (2014), e Davies (2012), são apresentadas algumas técnicas e ferramentas atuais, suas características e status de utilização contra as vulnerabilidades aqui apresentadas.

7 Considerações Finais

O IPv6 foi projetado para lidar com as exigências de segurança, mobilidade e serviços que o cenário atual exige, substituindo o IPv4, que tornou-se aquém do que se deseja. Entretanto, apesar de possuir suporte nativo à algumas técnicas e serviços de segurança, o IPv6 não está totalmente livre de problemas relacionados ao funcionamento do protocolo.

De acordo com os resultados obtidos neste trabalho, foi demonstrado que, utilizando as mesmas técnicas de exploração no IPv4, é possível explorar brechas de segurança existentes no IPv6, contradizendo o fato do protocolo ser seguro apenas por possuir suporte nativo a alguns mecanismos de segurança.

Dessa forma, considera-se relevante destacar que o IPv6 necessita de atenção à segurança do mesmo modo que o IPv4. É preciso estar atento às novas vulnerabilidades e métodos de exploração que irão surgir, a medida que o IPv6 for sendo adotado.

Uma das maiores dificuldades encontradas nesse trabalho foi sobre a correta utilização das ferramentas empregadas nos testes, sendo necessários estudos complementares dos aplicativos para compreensão do funcionamento dos parâmetros.

Sendo assim, contramedidas para as vulnerabilidades apresentadas podem ser implementadas, porém, estavam fora do escopo deste trabalho, ficando como sugestão de trabalhos futuros o tratamento e a prevenção destes problemas

Referências

COMER, Douglas E. **Interligação de Redes com TCP/IP.** 5. ed. v.1. Rio de Janeiro: Elsevier, 2006.

DAVIES, Joseph. **Understanding IPv6.** 3. ed. Sebastopol: O'Reilly, 2012.

DURDAĞI, Emre; BULDU, Ali. IPv4/IPv6 Security and Threat Comparisons. **Procedia – Social and Behavioral Sciences.** v. 2. n. 2. p. 5285-5291. 2010.

GIL, Antonio C. **Como elaborar projetos de pesquisa.** 5. ed. São Paulo: Atlas, 2010.

HAGEN, Silvia. **IPv6 Essentials.** 3. ed. Sebastopol: O'Reilly, 2014.

KIZZA, Joseph M. ***Guide to Computer Network Security*.** 3. ed. Londres: Springer, 2015.

MARCONI, Marina de A. LAKATOS, Eva M. **Técnicas de Pesquisa:** Planejamento e execução de pesquisas, amostragnes e técnicas de pesquisa, elaboração, análise e interpretação de dados. 7. ed. São Paulo: Atlas, 2012.

McCLURE, Stuart; SCAMBRAY, Joel; KURTZ, George. ***Hacking Exposed 6.*** Nova York: McGraw-Hill, 2009.

RFC 791. **Internet Protocol.** 1981. Disponível em: <https://tools.ietf.org/html/rfc791>. Acesso em: 17 Ago. 2017.

RFC 792. **Internet Control Message Protocol.** 1981. Disponível em: <https://tools.ietf.org/html/rfc792>. Acesso em: 29 Mai. 2017.

RFC 1636. **Report of IAB Workshop on.** 1994. Disponível em: https://tools.ietf.org/html/rfc1636>. Acesso em: 17 Ago. 2017. RFC 2460. **Internet Protocol, Version 6 (IPv6) Specification.** 1998. Disponível em:

<https://tools.ietf.org/html/rfc2460>. Acesso em: 17 Ago. 2017.

RFC 3849. **IPv6 Adsress Prefix Reserved for Documentation.** 2004. Disponível em: <https://tools.ietf.org/html/rfc3849>. Acesso em: 27 Ago. 2017.

RFC 4443. **Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.** 2006. Disponível em: <https://tools.ietf.org/html/rfc3631>. Acesso em: 10 Ago. 2017. RFC 4861. **Neighbor Discovery for IP Version 6 (IPv6).** 2007. Disponível em: <https://tools.ietf.org/html/rfc3631>. Acesso em: 17 Ago. 2016.

RFC 5735. **Special Use IPv4 Addresses.** 2010. Disponível em: <https://tools.ietf.org/html/rfc5735>. Acesso em: 21 Ago. 2017.

RFC 5737. **IPv4 Address Blocks Reserved for Documentation.** Disponível em: <https://tools.ietf.org/html/rfc5737>. Acesso em: 22 Ago. 2017.

SANTOS, Omar. ***End-toEnd Network Security:*** *Defense-in-Depth*. Indianapolis: Cisco Press, 2007.

SILVESTRE, Victor H. **Um Estudo Comparativo entre os Protocolos IPv4 e IPv6 com ênfase em Criptografia.** 2016. 49 f. Monografia (Trabalho de conclusão de curso em Segurança da Informação) - FATEC - Faculdade de Tecnologia, Ourinhos, 2016.

TANENBAUM, Andrew S.; WETHERALL, David. **Redes de Computadores.** 5. ed. São Paulo: Pearson Prentice Hall, 2011.

APÊNDICE A – Descrição dos comandos utilizados

nmap -sP 192.0.2.0/24

(a) -sP: Este parâmetro faz o Nmap fazer um *ping scan*, isto é, apenas determina se há endereços alcançáveis em uma rede específica;
(b) 192.0.2.0/24: Este parâmetro é a rede alvo, onde o Nmap irá executar.

alive6 enp0s9

(a) enp0s9: Este parâmetro determina em qual interface o aplicativo alive6 será executado, neste caso a interface enp0s9.

ip n

(a) n: Este parâmetro faz com que apenas as informações da tabela ARP do dispositivo sejam mostradas.

ip -6 n

(a) -6 n: Este parâmetro faz com que apenas as informações da tabela de vizinhos IPv6 do dispositivo sejam mostradas.

ping -c 1 192.0.2.200

(a) -c 1: Este parâmetro delimita o *ping* a lançar apenas um pacote ICMPv4;
(b) 192.0.2.200: Este parâmetro é o endereço alvo a ser testado, neste caso o endereço 192.0.2.200.

./arpoison -i enp0s8 -d 192.0.2.100 -s 192.0.2.200 -t 08:00:27:44:aa:bc -r 08:00:27:ff:ff:ff

(a) -i: Este parâmetro determina em qual interface o aplicativo será executado, neste caso a interface enp0s8;
(b) -d: Identifica qual o IP será alvo do envenenamento, neste caso o endereço 192.0.2.100;
(c) -s: Identifica qual o IP que falsamente originou a mensagem ICMPv4 envenenada, neste caso o endereço 192.0.2.200;
(d) -t: Identifica o endereço MAC do IP alvo, neste caso a sequência 08:00:27:44:aa:bc;
(e) -r: Identifica o endereço MAC falso a ser injetado no alvo, neste caso a sequência 08:00:27:ff:ff:ff.

parasite6 enp0s9 08:00:27:ff:ff:ff

(a) enp0s9: Este parâmetro determina em qual interface o aplicativo alive6 será executado, neste caso a interface enp0s9;
(b) 08:00:27:ff:ff:ff: Identifica o endereço MAC falso a ser injetado em qualquer alvo, neste caso a sequência 08:00:27:ff:ff:ff.

arpspoof -i enp0s8 -t 192.0.2.100 192.0.2.200 -r

(a) -i: Este parâmetro determina em qual interface o aplicativo será executado, neste caso a interface enp0s8;
(b) 192.0.2.100: Identifica qual o IP será alvo do envenamento, neste caso o endereço 192.0.2.100;
(c) 192.0.2.200: Identifica qual o IP que falsamente originou a mensagem ICMPv4 envenenada, neste caso o endereço 192.0.2.200;
(d) -r: Este parâmetro faz com que o aplicativo execute o comando recursivamente, isto é, faça o mesmo para o endereço 192.0.2.200, sem precisar executar dois comandos seguidos.

parasite6 -R enp0s9

(a) -R: Este parâmetro faz com que o aplicativo execute o comando recursivamente, isto é, execute o envenenamento tanto no destino quanto na origem.

(b) enp0s9: Este parâmetro determina em qual interface o aplicativo será executado, neste caso a interface enp0s9.

Ferramentas de Geração e Armazenamento de *Logs* com Registros de Atividades dos Usuários no Acesso à Rede de Computadores

Adriano Raupp de Borba[1], Tiago da Silva Leal[1], Marcos Henrique de Morais Golinelli[2]

[1]Acadêmicos do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88960000 – Sombrio – SC – Brasil

[2]Docente do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88.960-000 – Sombrio – SC – Brasil

{adrianoraupborba,tiagodasilvaleal}@gmail.com,marcos.golinelli@ifc.edu.br

Abstract. *In order to help solve information security problems, it is necessary the use of tools that help to identify those responsible ones for traffic in computer networks. This article aims to apply a proposal for generation, storage and manipulation of logs, recording the connections established by users, devices and systems. The methodology used was a bibliographic and applied research, where, through the Rsyslog and LogAnalyzer applications, it was possible to remotely receive the events generated by the RouterOS, Windows and Linux systems for a centralizer. As results obtained, it is possible to identify the users that used the resources of the network, as well as the time and date of the connection, source and destination IP address, protocol and MAC address of the equipment connected to the network. Despite the decrease in performance of the tools used, as long as the amount of data generated increased, the*

implementation employed proved to be efficient in recording and analyzing the data.

Resumo. *Para ajudar a resolver problemas da segurança da informação, é necessário o uso de ferramentas que auxiliem na identificação dos responsáveis pelo tráfego nas redes de computadores. Este artigo tem como objetivo aplicar uma proposta para geração, armazenamento e manipulação de logs, registrando as conexões estabelecidas pelos usuários, dispositivos e sistemas. A metodologia utilizada foi uma pesquisa na litratura e aplicada, onde, por meio das aplicações Rsyslog e LogAnalyzer, foi possível receber remotamente os eventos gerados pelos sistemas RouterOS, Windows e Linux para um centralizador. Como resultados obtidos, pode-se identificar os usuários que utilizavam os recursos da rede, bem como o horário e data da conexão, endereço IP de origem e de destino, protocolo e endereço MAC dos equipamentos conectados à rede. Apesar da diminuição do desempenho das ferramentas utilizadas, à medida em que o volume de dados gerados ia aumentando, a implementação empregada mostrou-se eficiente quanto ao registro e análise dos dados.*

1 Introdução

As organizações no mundo necessitam de sistemas computacionais para dar agilidade e produtividade em suas atividades. Com isso, o número de dispositivos conectados às redes cresce constantemente. Entretanto, para manter o controle do uso desses recursos e serviços pelos usuários, são necessários mecanismos que possam registrar as atividades de forma individualizada.

A NBR ISO/IEC 27002 (ABNT, 2013), que trata das questões sobre técnicas de segurança, informa a importância do Registro (*log*) de eventos num sistema computacional, que tem a finalidade de armazenar o histórico de eventos dos usuários. A norma enfatiza, também, que essas informações são importantes para auxiliar uma auditoria e que esses dados sejam produzidos e armazenados por um período de tempo acordado, com objetivo de esclarecer futuras investigações.

De acordo com a lei Nº 12.965, conhecida como "Marco Civil da Internet" (BRASIL, 2014), o provedor de conexão à Internet não será responsabilizado civilmente por danos decorrentes de conteúdo de acesso à Internet gerado por terceiros, desde que forneça as informações necessárias para identificar a origem do infringente. Deste modo, aumenta a responsabilidade das organizações públicas e privadas quanto à identificação dos acessos de suas redes locais.

A importância do armazenamento dos *logs* dos equipamentos e a busca de ferramentas para fazer o armazenamento centralizado dos eventos, motivaram a elaboração deste trabalho. Além disso, a possibilidade de busca e análise destes registros para uso em caso de monitoramento e auditoria, bem como apresentar uma solução para geração e armazenamento de registros de conexões dos usuários em uma rede utilizando NAT (*Network Address Translation*), de modo que se possa atender à legislação vigente sobre a responsabilidade das organizações quanto ao fornecimento da Internet aos usuários.

Este trabalho tem como objetivo geral implementar uma solução utilizando ferramentas que possam gerar, armazenar e manipular *log* de conexões dos usuários com a Internet numa rede local, com uso de NAT. Outrossim, de forma que por meio das informações registradas, seja possível solucionar problemas de segurança da informação.

Como objetivos específicos visa apresentar protocolos e ferramentas utilizadas para sincronia de relógios, gerar e gravar eventos em sistemas e conexões. A utilização destes recursos torna possível obter informações necessárias para solucionar incidentes referentes ao uso indevido dos recursos da rede.

As seções abordadas neste trabalho são as seguintes: introdução, revisão da literatura, materiais e métodos, implementação, resultados e considerações finais.

2 Revisão da literatura

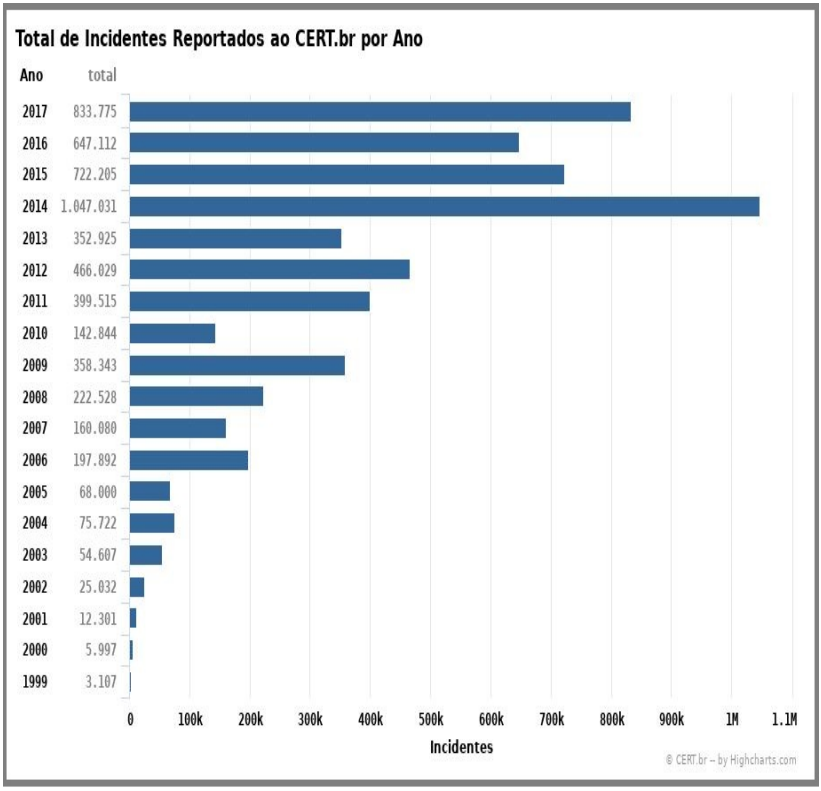
Nesta seção serão abordados assuntos sobre segurança da informação, registro de eventos em um sistema computacional, redes com NAT como mecanismo para compartilhamento de conexão com a Internet, protocolo *syslog* e ferramentas de gerência de *logs*.

2.1 Segurança da informação

O CERT.br (Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil), responsável por tratar de casos relativos a dispositivos que envolvam redes conectadas à Internet, recebeu, em 2017, o total de 833.775 de incidentes reportados. Este número é 29% maior que os casos relatados em 2016 (CERT.BR, 2017).

A Figura 01 mostra o total de incidentes reportados, por ano, no período entre 1999 e 2017. Entre os fatos citados estão: ataques de negação de serviço, tentativas de fraude, varreduras e propagação de códigos maliciosos e ataques à servidores Web. O pico desses valores foi alcançado no ano de 2014, tendo uma queda nos dois anos seguintes e voltando a aumentar em 2017.

Figura 01 - Incidentes Reportados ao CERT.br de 1999 a 2017.



Fonte: Incidentes reportados ao cert.br (2017).

A Lei Nº 12.965 de 23 de abril de 2014, conhecida como "Marco Civil da Internet", regulamenta a guarda de registros de conexão pelos provedores de Internet em seu artigo 13, e estabelece como dever do administrador de sistema autônomo, responsável por fornecer a conexão aos usuários, de manter os respectivos registros de acesso pelo prazo de 1 (um) ano. A guarda dos dados deve ser sob sigilo, em ambiente controlado e com segurança. A Lei ainda esclarece que autoridade policial ou

administrativa ou o Ministério Público poderão requerer cautelarmente as informações armazenadas (BRASIL, 2014).

2.2 Registro de eventos num sistema computacional

A ABNT NBR ISO/IEC 27002 (2013), que fornece diretrizes para prática dos controles de segurança da informação, define a importância do *log* nas atividades dos usuários, com o objetivo de registrar eventos e gerar evidências. Além destes registros, devem ser produzidos e armazenados *logs* referentes à falhas e eventos de segurança.

Ainda segundo a norma, os *logs* devem conter uma série de informações, pode-se citar como exemplo a identificação dos usuários (ID), datas, horários e detalhes de eventos, como horário de entrada (log-on) e saída (log-off) no sistema, registros das tentativas de acesso ao sistema, acessos aceitos e rejeitados, endereços e protocolos de rede, além do registro dos alarmes provocados pelo sistema de controle de acesso.

Outro ponto considerado importante é a sincronização dos relógios de todos os sistemas dentro da organização ou do domínio de segurança com uma fonte de tempo precisa. A respeito da importância da sincronização do horário dos dispositivos numa rede, a RFC 5905 (MILLS; et al., 2010) define o protocolo NTP (Network Time Protocol) amplamente utilizado para sincronizar os relógios dos sistemas entre um conjunto de servidores de tempo e clientes distribuídos.

O NTP.br (2018) ratifica a possibilidade de uso do protocolo em estações de trabalho, roteadores e outros equipamentos a partir de referências de tempo confiáveis. O projeto NTP.br também disponibiliza uma lista de endereços para ajustar dispositivos com a hora oficial do Brasil.

Para o envio e recebimento de *logs*, a RFC 5424 (GERHARDS, 2009) apresenta o syslog, que é um protocolo

que foi originalmente desenvolvido para enviar mensagens de notificação de eventos em redes IP. Este formato padrão para mensagens de *logs* permite descrever os elementos de dados de forma organizada e pode ser usado para transmitir facilmente informações estruturadas e analisáveis.

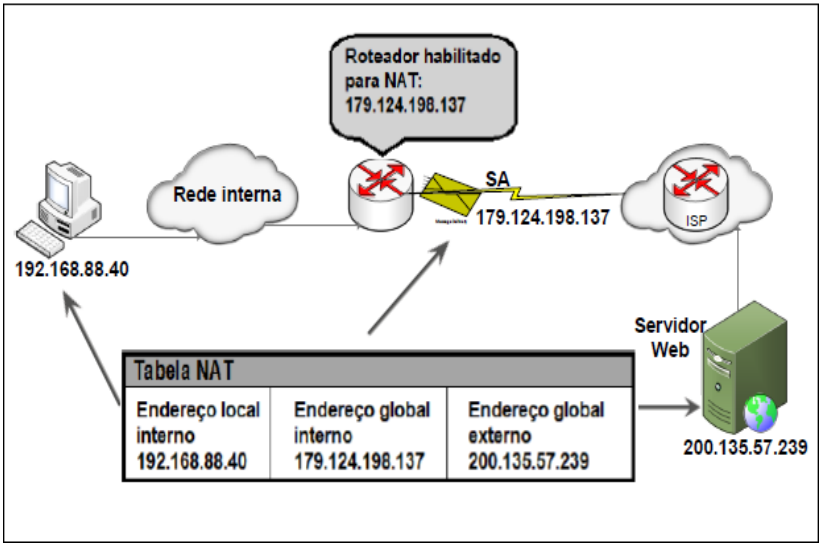
2.3 Redes com NAT

NAT é definida na RFC 3022 (SRISURESH, 2001), como o método no qual um único endereço IP público, ou um pequeno número deles, possa permitir que vários *hosts* acessem à Internet recebendo uma faixa de endereço IP privado.

Conforme Comer (2007), numa rede com NAT, cada computador recebe um endereço IP privado único. No entanto, quando um pacote precisa ser roteado para fora da rede, uma tradução de endereço é realizada, convertendo endereços IP privados em endereços IP públicos globalmente únicos, como mostra a Figura 02.

Esta técnica, de acordo IPv6.BR (2012), mostrou-se eficiente no sentido da economia de endereços IP, facilidade na numeração, oculta a topologia das redes internas e permite somente a entrada de pacotes de resposta às solicitações da rede interna. Porém, apresenta inconvenientes como a quebra do modelo fim-a-fim da Internet, impossibilitando o rastreamento dos pacotes, o que ocasiona problemas de segurança. E mesmo com o iminente esgotamento de endereços IPv4 no mundo, o Brasil alcançou apenas 20,17% de tráfego de dados em IPv6 no ano de 2017 (IPv6.BR, 2017).

Figura 02 – Como funciona o NAT.



Fonte: Os autores, 2018.

Em redes locais, onde o uso do IP privado é muito utilizado, o gerenciamento de acesso ao usuário é um fator determinante para prevenir que pessoas não autorizadas acessem aos sistemas e serviços. Um processo formal de autenticação deve ser implementado, permitindo ou negando a utilização de um determinado serviço na rede (NBR ISO/IEC 27001, 2013).

2.4 Ferramentas de geração e armazenamento de *logs*

Conforme Morimoto (2008), o *firewall* é o dispositivo colocado entre a Internet e a rede local, com a função de permitir apenas os serviços que se deseja disponibilizar, reduzindo os pontos vulneráveis da rede. O *firewall* trabalha realizando a filtragem dos pacotes, decidindo por meio de regras definidas aquilo que pode ser aceito e o que deve ser retido.

O *RouterOS* é um sistema operacional independente criado pela empresa *MikroTik*, baseado no *kernel* do *Linux*

versão 3.3.5. Este sistema disponibiliza vários recursos para geração de logs com suporte a *firewall*, roteamento, DHCP, rede sem fio, QoS, *Proxy*, NTP cliente/servidor e *Hotspot*. Possui interface de configuração avançada baseada na Web, sendo possível o acesso por meio de diversos sistemas operacionais. Com o *RouterOS* é possível criar um ponto de acesso aos usuários fazendo autenticação de clientes das redes locais (MIKROTIK, 2017).

Os equipamentos da *Mikrotik* com o *RouterOS* são capazes de registrar diversos eventos do sistema e informações de *status*. Os registros podem ser salvos no próprio equipamento ou até mesmo serem enviados para um servidor remoto. Cada entrada contém a hora e a data em que o evento de *log* ocorreu (MIKROTIK, 2017).

2.4.1 Rsyslog e Loganalyzer

O *Rsyslog* é um software de código aberto, resultante do aprimoramento do *syslogd*. Esse aperfeiçoamento proporcionou mais recursos e confiabilidade à ferramenta. Criado inicialmente por Rainer Gerhards, o software implementa o protocolo *syslog* com filtragem de conteúdo e oferece suporte para envio de mensagens no formato *syslog,* por meio do protocolo TCP (*Transmission Control Protocol*) e gravação em banco de dados MySQL (RSYSLOG, 2017).

O *LogAnalyzer* é uma ferramenta de código aberto para sistemas *Unix*. É uma interface Web para fazer a leitura de arquivos com registro de eventos de rede no formato *syslog*. Com o *Loganalyzer* é possível visualizar os registros com atualizações na tela a cada 5 segundos, ou com um período maior, permitindo acompanhar os *logs* gerados pelos dispositivos nas redes de computadores. A aplicação oferece relatórios e gráficos, exibindo de maneira dinâmica a leitura dos dados gerados. (ADISCON, 2018).

3 Materiais e métodos

Nesta seção serão abordados os métodos utilizados na realização deste trabalho, assim como os materiais utilizados na implementação e realização dos testes.

3.1 Métodos

Gil (2008) aponta quatro aspectos indispensáveis no qual uma pesquisa precisa ser classificada que são: (a) natureza; (b) abordagem; (c) objetivos; e (d) procedimentos técnicos, assim, a classificação deste estudo é ilustrada no quadro 01.

Quadro 01 – Classificação da metodologia.

Classificação	Metodologia utilizada
Quanto à natureza	Pesquisa aplicada
Quanto à abordagem	Qualitativa
Quanto aos objetivos	Exploratória
Quanto aos procedimentos	Experimental

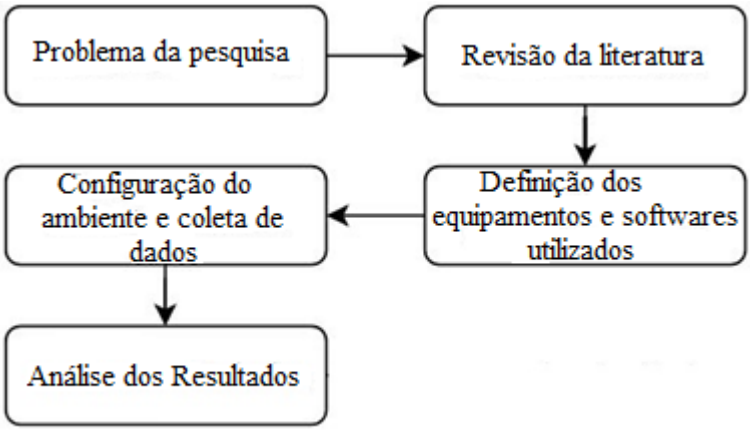
Fonte: os autores, 2018.

A pesquisa aplicada objetiva gera conhecimentos para aplicação prática, dirigidos à solução de problemas específicos. Quanto à abordagem, classifica-se como qualitativa, por meio de uma compreensão mais profunda de determinado assunto, procura explicar o porquê das coisas. Quanto aos objetivos, a pesquisa exploratória torna o problema de pesquisa mais explícito, ou seja, familiarizar o pesquisador com o objeto que está sendo pesquisado. Quanto aos procedimentos, a pesquisa experimental, para Gil (2008), consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de

influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto.

A Figura 03 representa as etapas da realização deste trabalho. Após a revisão da literatura sobre as ferramentas de geração e armazenamento de *logs*, aconteceu a escolha dos dispositivos envolvidos e os sistemas a serem utilizados nos testes. Na terceira etapa ocorreu o período da coleta de dados gerado pelas ferramentas empregadas. O passo seguinte foi analisar os dados gerados e finalizando com as conclusões do experimento.

Figura 03 – Etapas do trabalho.



Fonte: Os autores, 2018.

3.2 Materiais

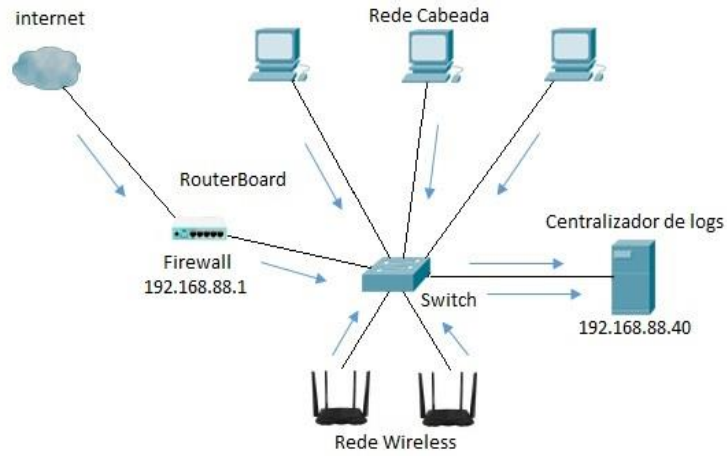
Para realização dos testes foi utilizado o ambiente da Secretaria Municipal de Saúde do município de São João do Sul – SC, conforme autorização expedida pelo gestor. Foram utilizados na implementação uma Routerboard RB 750Gr3, que possui 5 interfaces de rede FastEthernet, com o sistema operacional RouterOS versão 3.5.5. Um computador com processador core

i5, 6 GB de memória RAM e disco rígido de 500 GB, configurado como servidor remoto para armazenamento dos *logs*, com o sistema operacional *Linux* Debian 9.5. Também foram utilizados os seguintes dispositivos: dois switches encore 24 portas e cinco Access Points das marcas TP-Link e D-Link.

3.3 Ambiente de Pesquisa

A Figura 04 representa a topologia física da rede da Secretaria de Saúde e a maneira como foram distribuídos os equipamentos para registrar os eventos ocasionados pelos dispositivos da rede no ambiente de teste. Todos os aparelhos utilizados na pesquisa pertencem a secretaria, com excessão da *Routerboard* da Mikrotik, que foi adquirida pelos autores.

Figura 04 – Topologia Física da Rede.



Fonte: Os autores, 2018.

O ambiente de rede existente antes da implementação do trabalho não contava com nenhum tipo de registro das atividades

dos usuários, e todos compartilham a mesma senha da rede *Wireless*, não havendo nenhum registro das conexões. O compartilhamento de Internet é realizado por meio da técnica de NAT, em que o 'roteador' recebe um endereço IP válido na interface WAN (*Wide Área Network*) e compartilha a conexão com os computadores da rede local conectados na interface LAN (*Local Area Network*), traduzindo a rede de endereços IP privados dos computadores para o IP válido do roteador. O número de dispositivos conectados à rede, incluindo computadores, impressoras, aparelhos móveis dos funcionários e outros é de, aproximadamente, sessenta.

4 Implementação

De acordo com os objetivos propostos para este trabalho, a ferramenta utilizada para armazenar e centralizar os *logs* gerados pelos equipamentos na rede foi o *Rsyslog.* A escolha da aplicação foi por ser um software livre, compatível para receber mensagens do *firewall* da rede e de computadores com sistemas operacionais *Linux* e *Windows*. Os passos seguintes demonstram os procedimentos para a instalação e configuração da ferramenta, além de outros serviços necessários para o funcionamento adequado do sistema de gerenciamento de *logs*.

Foi utilizada a última versão estável do Sistema Operacional *Linux*, distribuição Debian 9.5. Inicialmente foi feita uma instalação padrão, sem interface gráfica, juntamente ao serviço de acesso remoto SSH (*Secure Shell*). Após as configurações iniciais do Debian, quando foi configurado um endereço IP fixo na mesma faixa da rede local, algumas configurações foram realizadas e serviços instalados antes da instalação do *Rsyslog*.

4.1 Sincronização do relógio

Para que o servidor de *log* trabalhe em sincronia de horário com os dispositivos clientes, de acordo com o fuso horário brasileiro, foram instalados alguns pacotes e realizados ajustes em algumas configurações, conforme segue no Quadro 01.

Quadro 01 – Instalação e configuração do serviço NTP.

Comando	Descrição
apt-get install ntp ntpdate	Instala o serviço e permite atualizar o relógio diretamente com um servidor NTP de referência.
service openntpd stop	Parar o serviço.
nano /etc/ntp.conf	Editar o arquivo ntp.conf.
dpkg-reconfigure tzdata	Reconfigura a região e o fuso horário do Sistema Operacional.
ntpdate -u pool.ntp.br	Comando para forçar o ajuste do horário independente do atraso.

Fonte: Os autores, 2018.

Durante a configuração do serviço NTP no centralizador de *logs,* o arquivo *"ntp.conf"* foi modificado com a inserção da linha *"servers pool.ntp.br"*. Além do servidor Debian, todos os clientes *syslog* tiveram que ajustar os relógios ao mesmo fuso horário e, de preferência, com o mesmo servidor de horário na Internet. Os dispositivos que enviam *logs* para o centralizador com SO *RouterOS*, *Linux* e *Windows* tiveram seus horários sincronizados com o mesmo servidor NTP.

4.2 Instalação do Rsyslog

O Debian 9.5 já vem com o utilitário *Rsyslog* nativo na sua instalação, sendo possível apenas configurá-lo para receber *logs* externos em arquivos de texto. Entretanto, como os dados serão armazenados numa base de dados, foi instalado o MySQL por ser um sistema de gerenciamento de banco de dados gratuito e mais utilizado atualmente (DB-Engines, 2018).

Como usuário root, foi necessário utilizar o comando *"apt-get install mysql-server"* para instalação do MySQL. O passo seguinte foi utilizar o comando *"apt-get install rsyslog-mysql"*. Este é o pacote que possibilita a conexão do serviço ao banco de dados.

Na sequência, conforme Quadro 02, foi editado o arquivo *syslog.service* com o comando *"nano /etc/systemd/system/syslog.service"*, incluindo a linha 4 para que o serviço inicie depois do MySQL.

Quadro 02 – Arquivo Syslog.service.

```
 1  [Unit]
 2  Description=System Logging Service
 3  Requires=syslog.socket
 4  After=mysql.service
 5  Documentation=man:rsyslogd(8)
 6  Documentation=http://www.rsyslog.com/doc/
 7  [Service]
 8  Type=notify
 9  ExecStart=/usr/sbin/rsyslogd -n
10  StandardOutput=null
11  Restart=on-failure
12  [Install]
13  WantedBy=multi-user.target
14  Alias=syslog.service
```

Fonte: Os autores, 2018.

4.2.1 Configuração do Rsyslog

O arquivo principal de configuração da aplicação é o *"rsyslog.conf"*, localizado no diretório *"/etc/"*. O Quadro 03 mostra as principais linhas que foram modificadas para que o servidor aceitasse as conexões dos clientes *syslog* e armazenasse os *logs* enviados remotamente. Foram retirados os comentários (#) das linhas 12,13,15 e 16 do arquivo, permitindo que a aplicação receba as conexões por meio dos protocolos UDP e TCP na porta 514. As linhas 17,18 e 19 foram inseridas, estabelecendo ao servidor somente aceitar dispositivos autorizados por meio de seus endereços IP.

Quadro 03 – Configuração Rsyslog.conf.

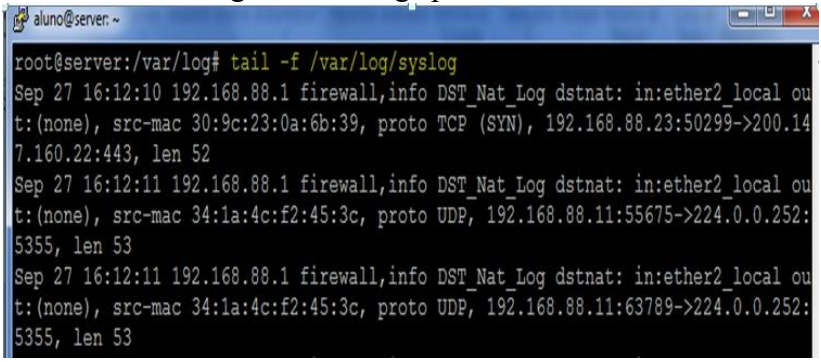
```
 1  #  /etc/rsyslog.conf    Configuration file for rsyslog.
 2  #
 3  #                       For more information see
 4  #                       /usr/share/doc/rsyslog-doc/html/rsyslog_conf.html
 5  ##################
 6  #### MODULES ####
 7  ##################
 8  module(load="imuxsock") # provides support for local system logging
 9  module(load="imklog")   # provides kernel logging support
10  #module(load="immark")  # provides --MARK-- message capability
11  # provides UDP syslog reception
12  module(load="imudp")
13  input(type="imudp" port="514")
14  # provides TCP syslog reception
15  module(load="imtcp")
16  input(type="imtcp" port="514")
17  # CLIENTES_SYSLOG  ACEITOS#
18   $AllowedSender UDP, 127.0.0.1, 192.168.88.0/24, 177.72.25.130,
179.124.198.136
19   $AllowedSender TCP, 127.0.0.1, 192.168.88.0/24, 177.72.25.130,
179.124.198.136
```

Fonte: Os autores, 2018.

Terminada a configuração, com permissões de root, aplicou-se o comando *"/etc/init.d/rsyslog restart"* para que o serviço pudesse reiniciar com as novas alterações. Para testar se a aplicação estava escutando na porta 514, foi executado o comando *"netstat -tnap | grep 514"*. Com a linha contendo *"tcp 0 0.0.0.0:514 OUÇA 672/rsyslogd"*, apresentada no terminal, significa que o serviço foi configurado corretamente.

A partir deste ponto, o servidor Debian está apto a receber *logs* enviados remotamente pelos dispositivos clientes configurados e autorizados, etapa que será apresentada na seção 4.4 (clientes *syslog)*. Além do banco de dados, o centralizador recebe os registros de eventos no diretório *"/var/log/"*. Para visualizar os *logs* conforme são recebidos com atualização instantânea, pode-se utilizar o comando *"tail -f /var/log/syslog"*, como ilustrado na Figura 05.

Figura 05 – Logs pelo Terminal.



Fonte: Os autores, 2018.

4.3 Instalação do LogAnalyzer

O *LogAnalyzer* é uma solução com interface Web para fazer a leitura dos logs na base de dados do *Rsyslog* com MySQL. Para instalação da ferramenta, é necessário que um conjunto de serviços seja instalado e algumas configurações realizadas para

o seu funcionamento. O Quadro 04 mostra os passos efetuados durante a instalação.

Quadro 04 – Instalação do LogAnalyzer

Comando	Descrição
apt-get install apache2 libapache2-mod-php php-mysql php-gd php php-cli php-mcrypt	Com permissão de root instala os pacotes necessários para a utilização do LogAnalyzer.
nano /etc/php/7.0/cli/php.ini	Edita o arquivo "php.ini", modificando a região e cidade, na linha "date.timezone = America/Sao_Paulo".
nano /etc/apache2/apache2.conf	Modifica o "apache2.conf" incluindo no final do arquivo "ServerName localhost"
/etc/init.d/apache2 restart	Reinicia o apache
wget http://download.adiscon.com/loganalyzer/loganalyzer-4.1.6.tar.gz	Baixar a última versão estável do LogAnalyzer no site do desenvolvedor.
tar -zxvf loganalyzer-4.1.6.tar.gz mkdir /var/www/html/loganalyzer mv loganalyzer-4.1.6/src/* /var/www/html/loganalyzer/ mv loganalyzer-4.1.6/contrib/* /var/www/html/loganalyzer/ chmod +x /var/www/html/loganalyzer/configure.sh /var/www/html/loganalyzer/secure.sh	A sequência de comandos irá descompactar o arquivo baixado, cria uma pasta para o LogAnalyzer no do diretório padrão do apache, move os arquivos da instalação para dentro da pasta criada. Na continuação, vai conceder permissão para os arquivos e depois executa os scripts. Seguindo, permite acesso do usuário para a pasta do LogAnalyzer do

cd /var/www/html/loganalyzer/ ./configure.sh && ./secure.sh chown -R www-data:www-data /var/www/html/loganalyzer/ usermod -G adm www-data	servidor Web e atribui grupo para o usuário.

Fonte: Os autores, 2018.

Para concluir o processo de configuração do *LogAnalyzer*, utilizou-se o navegador de um computador da rede local, digitando-se o endereço do servidor *"http://192.168.88.40/loganalyzer /install.php"*. O procedimento seguinte foi confirmar com botão *"Next"* nas duas primeiras telas. Na terceira etapa foi habilitada a opção *"Enable User Database"*, editando os campos *"Database Name"*, *"Database User"* e *"Database Password"* de acordo com as informações contidas no arquivo *"/etc/rsyslog.d/mysql.conf "*. Pressionando o botão *"Next",* nas próximas telas, fará com que o sistema crie as tabelas necessárias para o banco de dados.

Nos passos seguintes foi apresentada a tela para criação da conta do primeiro usuário. Na sequência é preciso preencher os campos informando os elementos do banco MySQL do *rsyslog* de onde o *LogAnalyzer* vai obter os dados. Por fim, ao concluir a instalação, o sistema redirecionará para a tela de *login.*

4.4 Clientes Syslog

Para individualizar o acesso à rede da Secretaria de Saúde, de maneira que se possa ter a identificação do usuário logado, um sistema de autenticação foi implementado. Para isso, foi utilizada a *Routerboard* da *Mikrotik*, habilitando o recurso *Hotspot*[12] do *RouterOS*. Desta forma, todos os usuários quando

[12]Hotspot é um sistema que solicita nome do usuário e senha.

precisavam utilizar a Internet, tanto por meio da rede cabeada, bem como os aparelhos que tinham conexão com a rede wireless, tiveram que usar suas credenciais para se autenticarem no *Hotspot.*

Com todos os usuários da Secretaria já possuindo *login* e senha, foi a vez de configurar a *Routerboard* para gerar os *logs* das conexões dos dispositivos. Foi no *firewall* do roteador, por meio de regras configuradas, que se definiu que tipo de evento deveria ser registrado. O equipamento utilizado da *Mikrotik* possui uma pequena capacidade de armazenamento, fazendo com que os logs gerados gravem apenas algumas linhas na memória RAM e sendo sobrescritos logo em seguida. Com isso, a etapa seguinte foi configurar a *Routerboard* de forma a enviar esses registros para serem gravados no centralizador de logs.

O roteador da *Mikrotik* foi configurado para receber um IP público na interface que faz conexão com o link de Internet do provedor e NAT para rede local, distribuindo a faixa de IP privado 192.168.88.0/24. De acordo com os serviços necessários para a Secretaria, algumas regras na tabela NAT do *firewall* foram incluídas, abrindo algumas portas e redirecionando para determinados dispositivos. O *Hotspot,* além de atribuir *logins* para os usuários, também permitiu controle de banda por perfil.

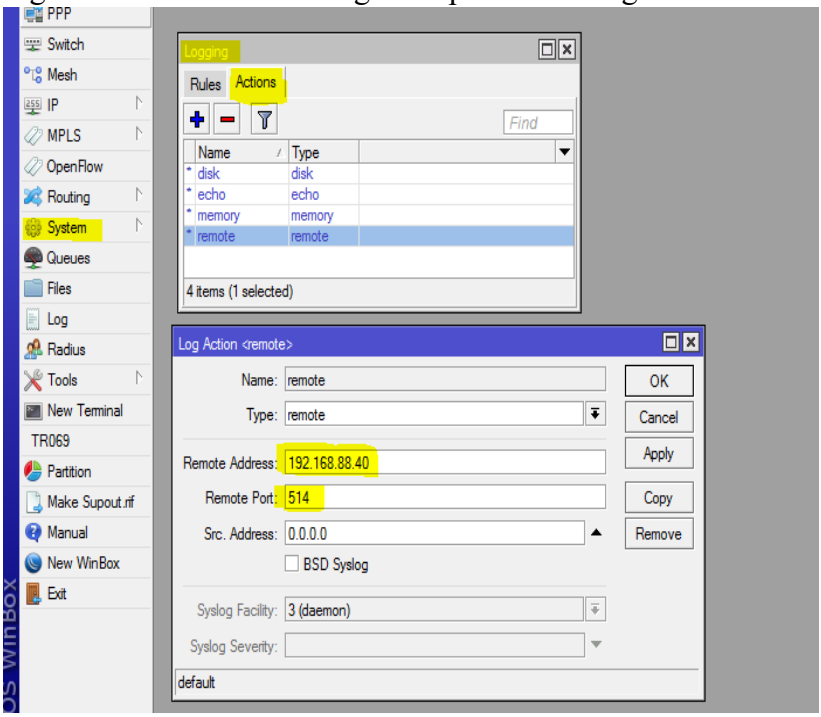
As regras de logs aplicadas no *firewall* do *RouterOS,* que permitiram registrar informações de eventos da rede, foram as seguintes: a primeira, na tabela "NAT" na cadeia "dstnat", para registrar todas conexões originadas da rede local. A segunda, na tabela "NAT" na cadeia "srcnat", para gravar conexões estabelecidas da rede externa (Internet) com a rede interna. A terceira, na tabela "NAT" na cadeia "dstnat", com destino às portas mais utilizadas na interface WAN da *Routerboard*. A última na tabela "*Filter*" na cadeia "*input*" com destino à interface de entrada da Internet, bloqueando e gerando logs de

acessos por meio do protocolo ICMP *(Internet Control Message Protocol).*

Após todos os equipamentos estarem operantes na rede local, com serviços habilitados e usuários cadastrados, o passo seguinte foi configurar a *Routerbord* para enviar os logs remotamente para o servidor. A interface gráfica do *RouterOS*, exibido na Figura 06, mostra a janela do menu *"System"*, submenu *"Logging"*, aba *"Actions"*, opção *"remote"*, sendo adicionado o endereço "192.168.88.40" e porta "514" do servidor. Também no submenu *"Logging"*, agora na opção *"Rules"*, foi escolhido os tópicos para geração de logs e a ação *"remote"*, já configurada anteriormente.

Para testar a compatibilidade do servidor *Rsyslog* com outros sistemas operacionais, além do *RouterOs*, uma outra máquina cliente com SO *Linux Debian* e mais três computadores com sistema *Windows* foram configurados para enviar logs para o centralizador. A máquina *Linux* teve apenas a inclusão de uma linha no seu arquivo *"rsyslog.conf"*, inserindo o endereço do servidor. Nas máquinas *Windows* foi instalada a aplicação *"el2sl"*, um utilitário para configurar os registros de eventos para serem encaminhados para o servidor externo.

Figura 06 – RouterOS configurado para enviar log remotamente.

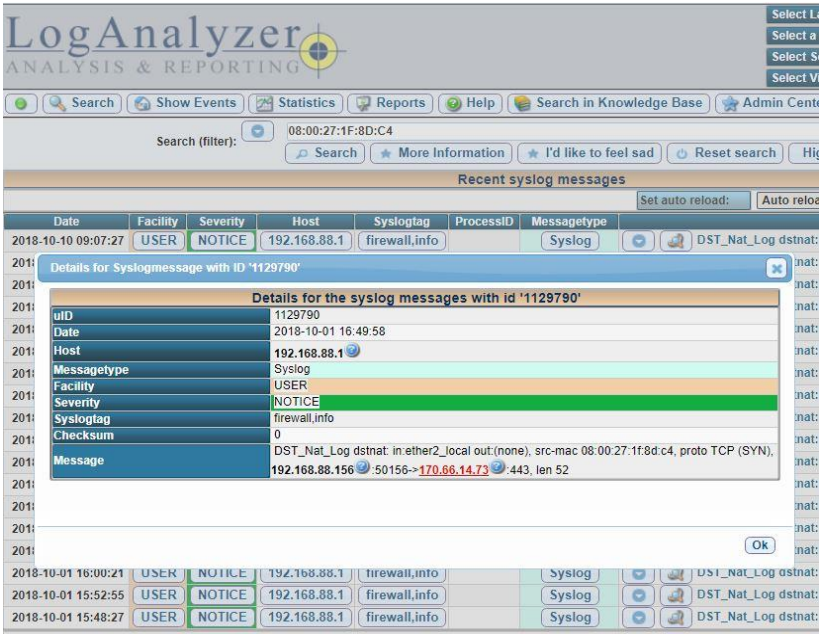


Fonte: Os autores, 2018.

5 Resultados

A tarefa de armazenar e analisar os registros de eventos produzidos pelos dispositivos na rede da Secretaria de Saúde tornou-se possível com o uso do *Rsyslog* em conjunto com o *LogAnalyzer*. A interface Web, oferecida pela aplicação, exibida na Figura 07, ilustra uma busca por meio de filtros de forma dinâmica no banco de dados. No exemplo da Figura, é possível identificar os elementos mais importantes no *log*, sendo mostrados data, horário, o endereço IP de origem da conexão e a porta utilizada, bem como o endereço IP e porta de destino no campo mensagem.

Figura 07 – Interface Web LogAnalyzer.
Fonte: Os autores, 2018.



Dentro do ambiente real de funcionamento dos recursos da rede da Secretaria, algumas simulações foram feitas para poder demonstrar os dados gerados pelos equipamentos. Entre as informações coletadas, foi possível ver o exato momento que o serviço dhcp do *RouterOS* atribui um endereço IP *"192.168.88.156"* ao dispositivo com endereço MAC *"08:00:27:1F:8D:C4"*. No passo seguinte, o usuário "adriano" ao conectar-se ao *Hotspot*, vincula suas credenciais ao IP *"192.168.88.156"*. A partir deste momento, todas as conexões estabelecidas por meio deste IP com a rede externa começaram a ser armazenadas.

A interface gráfica do *LogAnalyzer* permite várias combinações de filtros de pesquisa. Entre elas, está a possibilidade de concatenar informações por meio do campo

"*Search* (*filter*)" para aprimorar a pesquisa. Como exemplo, utilizou-se a busca de informações com acesso do usuário *"adriano"*, inserindo os seguintes dados: *"source: =192.168.88.1 adriano logged in"*. O resultado da pesquisa permitiu visualizar todas as vezes em que o usuário *"*adriano*"* esteve logado no *Hotspot*, quantidade de dispositivos utilizados, data e horário das conexões, endereços IP e endereços MAC vinculados aos aparelhos. A Figura 08 foi elaborada com base nas linhas de logs do *LogAnalyzer* para ilustrar alguns desses acessos do usuário *"*adriano*"*.

Figura 08 – Logins do usuário.

Linha	Host	Date	Message
01	192.168.88.1	2018-11-09 15:06:46	adriano (A4:81:EE:93:D9:04): replace cookie: user logged in
02	192.168.88.1	2018-11-09 15:06:46	adriano (192.168.88.62): logged in
03	192.168.88.1	2018-11-09 14:17:04	adriano (90:2B:34:F9:50:76): replace cookie: user logged in
04	192.168.88.1	2018-11-09 14:17:04	adriano (192.168.88.10): logged in
05	192.168.88.1	2018-10-30 13:56:34	adriano (08:00:27:1F:8D:C4): replace cookie: user logged in
06	192.168.88.1	2018-10-30 13:56:34	adriano (192.168.88.92): logged in
07	192.168.88.1	2018-10-24 10:45:59	adriano (08:00:27:A4:E1:D7): add cookie: user logged in
08	192.168.88.1	2018-10-24 10:45:59	adriano (192.168.88.184): logged in
09	192.168.88.1	2018-10-24 08:35:55	adriano (5C:C9:D3:9C:95:D7): add cookie: user logged in
10	192.168.88.1	2018-10-24 08:35:55	adriano (192.168.88.182): logged in
11	192.168.88.1	2018-10-10 09:03:20	adriano (08:00:27:1F:8D:C4): add cookie: user logged in
12	192.168.88.1	2018-10-10 09:03:20	adriano (192.168.88.156): logged in

Fonte: Os autores, 2018

Neste momento, todas as conexões estabelecidas da rede interna com a Internet já estavam sendo geradas e armazenadas. No entanto, tão importante quanto aquilo que é originado da rede local com destino à Internet são as conexões oriundas das redes externas com a rede interna. As regras aplicadas no *firewall* da *Routerboard* também permitiram observar as conexões externas com as portas abertas para os serviços internos na rede. Por

exemplo: os *logs* exibiam os endereços IP públicos que estabeleciam conexão com os servidores da Secretaria.

Durante a análise dos *logs*, pode-se identificar comportamento fora dos padrões habitual da rede. Por meio do horário de acesso e o endereço IP do host, foi possível acompanhar usuários com equipamentos que tinham tráfego na rede. Os eventos da própria *Routerboard* também eram registrados. Tentativas de acesso ao sistema *RouterOS*, tanto as que vinham da interface local ou pela interface WAN, identificavam a origem dos endereços IP.

O servidor de *logs* registra num curto espaço de tempo uma grande quantidade de linhas de eventos produzidos pelos dispositivos monitorados. Para fazer a análise das informações geradas é necessário usar filtros e ainda percorrer as várias páginas apresentadas pela aplicação Web. Para poder exemplificar alguns logs gravados no banco de dados, foram capturadas linhas dos registros que serão apresentadas na Figura 09.

Figura 09 – Linhas de Logs.

Linha	Host	Date	Message
01	192.168.88.1	2018-11-19 20:38:23	SRC_Log_NAT srcnat: in:(none) out:ether2_local, proto TCP (SYN), 200.135.57.239:40770->192.168.88.40:22, NAT 200.135.57.239:40770->(179.124.198.137:9522->192.168.88.40:22), len 52
02	server	2018-11-19 20:38:27	Invalid user from 200.135.57.239 port 40770
03	server	2018-11-19 20:39:10	Connection closed by 200.135.57.239 port 40770 [preauth]
04	192.168.88.1	Today 16:42:50	SRC_Log_NAT srcnat: in:(none) out:ether2_local, proto TCP (SYN), 192.69.253.148:59370->192.168.88.40:22, NAT 192.69.253.148:59370->(179.124.198.137:9522->192.168.88.40:22), len 60
05	server	Today 16:43:08	Accepted password for aluno from 192.69.253.148 port 59370 ssh2
06	192.168.88.1	2018-11-20 01:18:19	ICMP_Log input: in:pppoe-out1 out:(none), src-mac 6c:3b:6b:29:53:e0, proto ICMP (type 8, code 0),69.25.7.69 {performance-measurement-174-1.mia003.pnap.net} ->179.124.198.137, len 44
07	192.168.88.1	2018-10-02 01:40:32	login failure for user admin from 82.100.63.189 via web
08	192.168.88.1	2018-11-19 21:12:49	login failure for user admin from 200.135.57.239 via winbox

Fonte: Os autores, 2018.

A Figura 09 apresenta um compilado da tela do *LogAnalyzer* com alguns campos exibidos de acordo com a tela original da ferramenta, exibindo algumas atividades dos dispositivos na rede. As colunas "*Host*", "*Date*" e "*Message*" e os dados apresentados pelas linhas da figura estão dispostos conforme a interface Web do *LogAnalyzer*.

Observando a linha 01 da Figura 09, é possível visualizar que o *firewall* da rede (*Routerboard*), representado pelo *host* "192.168.88.1", registrou uma conexão estabelecida entre o IP público "200.135.57.239" e o servidor de logs por meio da porta "22" e endereço IP "192.168.88.40" da rede local da secretaria. O log mostra o prefixo "*SRC_Log_NAT*", regra criada para identificar todo log gerado por endereços externos com destino à rede local. A linha mostra, ainda, que a conexão passou pelo IP público "179.124.198.137" na porta "9522" do roteador e foi redirecionada para o IP do servidor.

O desfecho da tentativa de ter acesso ao servidor de log por meio de uma conexão *SSH*, conforme mostrado no parágrafo anterior, é exibida na linha 02 da Figura 09. O *host "server"*, que representa o servidor de logs armazenando seus próprios registros, mostra na linha 02 que, 3 segundos após o log gerado na linha 01, o acesso foi negado por digitação de usuário inválido. E por fim, alguns segundos depois, a linha 03, da Figura 08, exibe a mensagem de conexão fechada.

A linha 04, da Figura 09, mostra outra tentativa de acesso via protocolo *SSH* ao servidor de logs. Dessa vez, a conexão tem o endereço IP de origem "192.69.253.148" e porta "59370". Como esse foi um acesso planejado para demonstrar no trabalho uma entrada bem sucedida no servidor, a linha 05 mostra que as credenciais foram aceitas, exibindo a mensagem de senha aceita para o usuário "aluno".

Para impedir possíveis mapeamentos e até mesmo ataques por meio do protocolo ICMP, uma regra no *firewall* da rede foi criada com prefixo *"ICMP_Log"* para bloquear os pacotes e gerar logs. A linha 06, da Figura 09, mostra que no dia 20/11/2018, "01:18:19", houve um bloqueio do protocolo ICMP[13] no endereço IP configurado na porta WAN do roteador da secretaria. Com auxílio do site *"https://www.iplocation.net/"* foi possível identificar a localização geográfica do endereço IP *"69.25.7.69"*, com coordenadas dos Estados Unidos.

As linhas 07 e 08, da Figura 09, mostram tentativas malsucedidas de acesso à interface Web de gerenciamento da *Routerboard*. A linha 07 exibe a mensagem de falha de *login* para o usuário "*admin"* que foi realizada por meio de um navegador de Internet. Com relação à linha 08, a mesma mensagem de falha de acesso é exibida, porém, agora o log

[13]Geralmente enviado pelo utilitário ping.

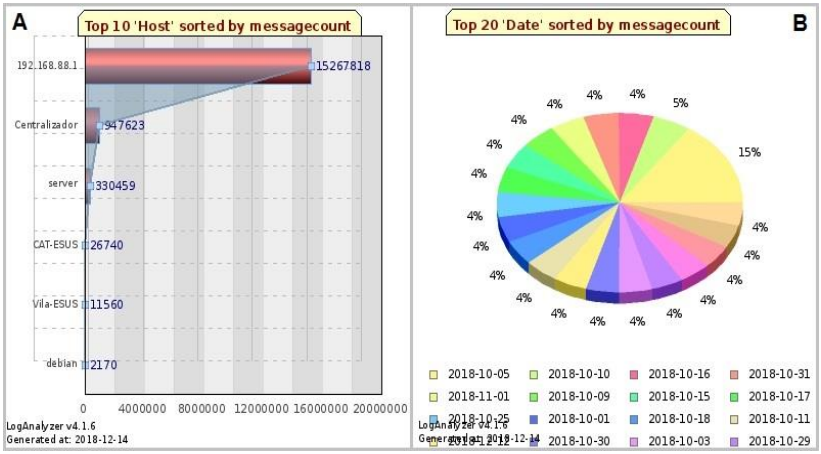
informa que o acesso foi efetuado por meio do aplicativo *winbox* (utilitário de configuração gráfica para o sistema *RouterOS*).

Com relação ao desempenho das ferramentas, pode-se perceber que, ao fazer uma busca no banco de dados com um período de tempo superior a 24 horas, a resposta começou a ficar um pouco mais lenta. Entretanto, quando a pesquisa é definida para um dia, a resposta foi quase instantânea. Com isso, a possibilidade do usuário definir na aplicação o período do filtro, foi um ponto positivo quanto à performance do sistema.

As configurações de regras no *firewall* do equipamento da *Mikrotik* tiveram influência no desempenho da aplicação e no espaço utilizado em disco. O objetivo foi criar regras que registrassem as informações mais importantes no centralizador, sem que isso ocasionasse um número excessivo de dados. As configurações aplicadas no *firewall* foram sendo acompanhadas na medida em que cada regra era alterada, sempre observando o tráfego gerado na interface de rede e o crescimento do tamanho do banco de dados no servidor de logs.

O *LogAnalyzer* possui gráficos que podem ser editados e também permite a criação de novos com base em seus registros. A Figura 10 mostra dois exemplos de gráficos que já vêm configurados com a instalação padrão do sistema. Na parte A da figura, é possível ver a imagem com uma configuração dos 10 clientes que mais enviavam logs para servidor. Como foram apenas 7 *hosts* configurados para envio de eventos, todos aparecem com a quantidade de linhas de logs enviados para o centralizador. Chama a atenção, nesta tela, o grande número de mensagens produzidas pela *Routerboard*, sendo esse o equipamento que era responsável pela geração de logs dos usuários com a Internet.

Figura 10 – Gráficos do LogAnalyzer.



Fonte: Os autores, 2018.

Na parte B da Figura 10, é apresentado um gráfico em formato de *pizza*. São exibidos os 20 dias que mais produziram logs no servidor. O primeiro ponto a ser observado na imagem, foi o grande volume de dados criados no dia 05/10/2018, devido alguns testes com regras no *firewall* do *Mikrotik* que proporcionaram um volume excessivo de registros de logs. A figura mostra também as datas em que esses percentuais de ocupação em disco ocorreram. Pode-se perceber que todos os dias que fazem parte do gráfico são dias úteis de trabalho da Secretaria de Saúde.

As regras de logs aplicadas no *firewall* do *RouterOS*, juntamente aos eventos enviados pelos servidores da secretaria e os dados gerados pelo próprio servidor de logs, foram os responsáveis pelo volume de informações coletadas. Durante cinquenta e cinco dias consecutivos, o centralizador de logs gerou 12.269.812 registros e ocupou um espaço em disco de 3,5 *Gigabyte* no banco de dados.

6 Considerações Finais

Após a pesquisa na literatura e análise da legislação vigente sobre a necessidade do registro dos *logs* numa rede de computadores, foi possível apresentar proposta de solução para geração e armazenamento dos eventos. O *rsyslog,* junto ao *LogAnalyzer,* mostrou-se eficiente quanto ao recebimento dos *logs* remotos que foram originados pelo *firewall* do *RouterOS* e também por máquinas com sistemas *Linux* e *Windows*. A implementação utilizada demonstrou que é possível registrar e monitorar as conexões dos dispositivos mesmo numa rede com uso de NAT.

O IPv6 retirará a necessidade do *log* das conexões dos clientes em NAT. No entanto, os resultados deste trabalho continuam sendo úteis, visto que o gerenciamento de eventos diversos da rede é importante tanto na versão 4 ou 6 do protocolo IP. Além disso, a implementação realizada envolveu conteúdo de várias disciplinas da grade curricular[14] do curso de Redes, proporcionando a prática do conhecimento adquirido durante toda a formação.

Dentre as dificuldades encontradas, destaca-se a baixa quantidade de material bibliográfico localizado sobre a aplicação das regras no *firewall* na *Routerboard* da *Mikrotik* para geração de logs na relação da quantidade de espaço ocupado em disco. Muitas regras foram testadas até que se encontrasse uma boa situação, onde informações importantes fossem registradas e não produzissem um grande volume de dados no armazenamento.

Por fim, considerando que a quantidade de informações armazenadas no disco rígido acarreta a diminuição de

[14]Sistemas operacionais, protocolo de comunicação, dispositivos de redes, banco de dados, serviços de redes, segurança de redes e gerência de redes.

desempenho da aplicação, fica como proposta de trabalhos futuros a utilização em conjunto de uma ferramenta de *backup* para exportação dos dados. Isso possibilitará transferir os arquivos gerados pelo centralizador de logs para um servidor externo, o que permitirá uma limpeza da base de dados numa periodicidade definida e proporcionará ao *Rsyslog* e ao *LogAnalyzer* um melhor desempenho quanto às buscas de registros.

Referências

ADISCON LOGANALYZER. **View System Messages Via Web.** 2018. Disponível em: < https://loganalyzer.adiscon.com/> Acesso em: 17 set 2018.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 27001. **Tecnologia da Informação – Técnicas de Segurança – Sistemas de gestãoda segurança da informação - Requisitos.** Rio de Janeiro: ABNT, 2013.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 27002. **Tecnologia da informação – Técnicas de segurança – Código de prática para a gestão da segurança da informação.** Rio de Janeiro: ABNT, 2013a.

BRASIL. Lei nº 12.965, de 23 de abril de 2014. **Marco Civil da Internet.** Brasília, DF, Disponível em:<http://www.planalto.gov.br/ccivil_03/_ato20112014/2014/lei/l12965.htm>. Acesso em: 09 ago. 2018

CERT.BR. **Incidentes Reportados ao Cert.br.** 2017. Disponível em: <https://www.cert.br/stats/incidentes/2017-jan-dec/analise.html> Acesso em: 09 ago 2018.

COMER, Douglas E. **Redes de computadores e Internet:** abrange transmissão de dados, ligações inter-redes, web e aplicações. 4.ed. Porto Alegre: Bookman, 2007.

DB-ENGINES. **DB-Engines Ranking.** 2018. Disponível em:<https://db-engines.com/en/ranking>. Acesso em: 20 nov 2018.

GERHARDS. Rainer. **The Syslog Protocol.** IETF. RFC 5424**.** 2009. Disponível em: <https://www.ietf.org/rfc/rfc5424.txt>. Acesso em: 11 ago 2018.

GIL, Antônio Carlos. **Métodos e técnicas de pesquisa social.** 6. ed. São Paulo: Atlas, 2008.

IPv6.BR. **Brasil cresce e chega 20% de adoção de IPv6 segundo o Google.** 2017. Disponível em: < http://ipv6.br/post/medicoes-google/> Acesso em: 14 ago 2018.

IPv6.BR. **Soluções Paliativas.** 2012. Disponível em: <http://ipv6.br/post/introducao/> Acesso em: 10 ago 2018.

MIKROTIK. **Manual: Recursos do RouterOS.** 2017. Disponível em:<https://wiki.mikrotik.com/wiki/Manual:RouterOS_features#RouterOS_features> Acesso em: 11 ago 2018.

MIKROTIK. **Manual: Sistema / Log.** 2018. Disponível em: <https://wiki.mikrotik.com/wiki/Manual:System/Log#Topics> Acesso em: 11 ago 2018.

MILLS. David; et. al. **Network Time Protocol Version 4.** IETF. RFC 5905**.** 2010. Disponível em: <http://www.ietf.org/rfc/rfc5905.txt>. Acesso em: 10 ago 2018.

MORIMOTO, Carlos Eduardo. **Servidores Linux:** guia prático. Porto Alegre: Sul Editores, 2015.

NTP.BR. **Projeto NTP.br.** 2018. Disponível em: < https://ntp.br/estrutura.php > Acesso em: 10 ago 2018.

RSYSLOG. **The Rocket-Fast System For Log Processing.** 2017. Disponível em:<https://www.rsyslog.com/doc/master/history.html > Acesso em: 17 SET 2018.

SRISURESH, Pyda. **Traditional IP Network Address Translator (Traditional NAT).** IETF. RFC 3022**.** 2001. Disponível em: <https://www.ietf.org/rfc/rfc3022.txt>. Acesso em: 10 ago 2018.

Implementação de Melhorias na Infraestrutura de Redes da Empresa Tonetto

João Antônio do Prado Azevedo[1], Rafael do Nascimento[1], Jéferson Mendonça de Limas[2], Marcos Henrique Morais Golinelli[2]

[1]Acadêmicos do Instituto Federal Catarinense – Campus Avançado Sombrio – 88960000 – Sombrio – SC – Brasil

[2]Docente do Instituto Federal Catarinense – Campus Avançado Sombrio – 88960000 – Sombrio – SC – Brasil

joao.a.azevedo@hotmail.com,rafanacimento@gmail.com
{jeferson.limas, marcos.golinelli}@ifc.edu.br

Abstract. *This paper presents the development and implementation of a structured cabling in the company Tonetto, where a feasibility study of the implantation was first elaborated, with which it was observed the importance of the elaboration of a standardized design for the company. It is a small agricultural equipment industry that already has a functional structure, which is the object of study of this article. It is intended to standardize and adapt the existing materials to an updated, organized, and mainly safe and reliable model. For this, we analyzed the current security systems with remote access, surveillance cameras, intrusion alarm and also the telephony part, in addition to the implementation of a server virtualization system; a reconstruction of the logical and physical cabling as a standardized design of*

Ethernet cabling. With this, it was observed that it is an acceptable project and it is important to any company, leaving the standard structure ready for expansion.

***Resumo**: O presente artigo, apresenta o desenvolvimento e a implantação de um cabeamento estruturado na empresa Tonetto, na qual, em um primeiro momento, foi elaborado um estudo de viabilidade da implantação. Com isso, foi observada a importância da elaboração de um projeto padronizado para a empresa. A empresa em questão é uma indústria de equipamentos agrícolas de pequeno porte, que já conta com uma estrutura funcional, a qual está sendo o objeto de estudo deste artigo. É pretendido padronizar e adequar os materiais já existentes a um modelo atualizado, organizado, e principalmente seguro e confiável. Para isso, foram analisados os sistemas atuais de segurança com acesso remoto, câmeras de vigilância, alarme de invasão e também a parte de telefonia, além da implantação de um sistema de virtualização do servidor. Fez-se também, a reconstrução do cabeamento lógico e físico, sendo este referente a um projeto padronizado de cabeamento Ethernet. Com isso, observou-se que é um projeto aceitável e de necessidade a qualquer empresa, deixando a estrutura padronizada e pronta para a expansão.*

1 Introdução

No Brasil, a maioria das empresas de pequeno e médio porte possuem cabeamento não estruturado, normalmente devido ao

crescimento desordenado ou à falta de políticas de desenvolvimento tecnológico. (PINHEIRO, 2003, p.3).

O objetivo fundamental do sistema de cabeamento estruturado é organizar e reunir os cabos existentes e preparar a estrutura para expansões futuras, criando assim um sistema organizado e podendo servir como referência para novos projetos de redes (PINHEIRO, 2003).

Conforme pesquisas internacionais, citadas por Júlio Ross (2007, p.6), cerca de 25% dos funcionários sofrem mudanças em postos de trabalho em menos de 1 ano. Esta realidade faz com que as empresas devam estar preparadas para tais alterações, contando com um layout de cabeamento estruturado, facilitando, assim, quaisquer mudanças. Ross concorda que um Sistema de Cabeamento Estruturado (SCE) dentro das empresas, surge para atender às necessidades de mudanças com o passar dos anos.

Dentro das perspectivas apontadas, encontramos a empresa Tonetto, localizada na cidade de Turvo - SC e local de trabalho de um dos autores. No decorrer do curso, os autores reconheceram problemas que poderiam ser resolvidos, tais como desconhecimento da rede lógica e física e falta de controle ao acesso de informações. A empresa, a partir de tais constatações, concordou em participar do projeto.

Os autores, no início do projeto, realizaram uma entrevista com a direção da empresa Tonetto, na qual foi identificada a realidade do ambiente institucional, as tecnologias que já estavam difundidas na empresa e as melhorias que os diretores sentiam necessidade. Nessa entrevista, também foram apresentadas algumas tecnologias e soluções de mercado. Assuntos como acesso remoto, CFTV (Circuito Fechado de TV),

alarme de furto e incêndio, telefonia, nobreaks e *backups*, foram discutidos e acordados entre as partes, como itens que deveriam ser trabalhados durante o desenvolvimento do projeto.

Com a realização deste projeto, objetivou-se redesenhar o cabeamento estruturado da rede, melhorando o ambiente empresarial para os diretores e colaboradores da empresa. Para alcançar tais finalidades, foi necessário melhorar a infraestrutura física e lógica da rede de dados. Para isto, foram utilizadas técnicas e conceitos de mercado, já difundidos em outros projetos.

Entre as possíveis alterações do projeto, estão a virtualização do servidor, a reconstrução do cabeamento estruturado, a instalação do sistema de alarme e monitoramento por câmeras e a disponibilidade de acesso remoto aos diretores da empresa.

Assim, este trabalho está organizado em seis seções: a seção 1, apresenta a introdução; a seção 2, mostra o referencial teórico, servindo como embasamento para o desenvolvimento do tema específico; a seção 3, se dedica a normas de cabeamento; seção 4, trata da virtualização; a seção 5, expõe os aspectos metodológicos, detalhando a forma como o estudo foi executado; a seção 6, exibe os resultados obtidos e explana uma discussão articulada dos elementos que compõem o trabalho; e por fim, na seção 7, apresenta-se as considerações finais.

2 Referencial Teórico

Neste capítulo, serão explanados alguns conceitos básicos para a compreensão do projeto, tecnologias, e suas formas construtivas.

2.1 Projeto Lógico

Cabeamento lógico é uma rede lógica e conceituada como uma abstração da infraestrutura da rede física, com o objetivo de tornar mais simples a organização de redes para *hosts*, máquinas virtuais e serviços em redes que podem estar ou não conectados entre si, apesar de pertencerem a uma mesma rede física (RODRIGUES, 2017).

A topologia de rede é o padrão no qual o meio de rede está conectado aos computadores e a outros componentes de rede. Podendo então ela ser uma estrutura topologicamente física ou lógica (PAULINO, 2013).

A topologia lógica refere-se à maneira como os sinais agem sobre os meios de rede ou a maneira como os dados são transmitidos através da rede, a partir de um dispositivo para outro, sem ter em conta uma interligação física. Topologias lógicas são capazes de serem reconfiguradas dinamicamente por tipos especiais de equipamentos, como roteadores e *switches* L2 (PAULINO, 2013).

No caso de um projeto de rede física para *campus* e corporações, há a seleção de tecnologias de LANs e WANs. Durante essa fase do projeto, são realizadas escolhas referentes ao cabeamento, protocolos da camada física e da camada de enlace de dados, além de dispositivos de interligação de redes (como *hubs*, *switches* e roteadores). Um projeto lógico influencia como será fundamentado o projeto físico (OPPENHEIMER, 1999, p. 259 – 304).

Sendo assim, para atender às metas de um cliente, o projetista de rede desenvolve a arquitetura de uma topologia lógica antes de selecionar produtos ou tecnologias físicas,

identificando nesta fase: as redes e pontos de interconexão, os tipos de dispositivos para interligar as redes e o tamanho do escopo para estas redes. Isso tudo é pensado, documentado e planejado, antes ainda de serem feitas aplicações reais (OPPENHEIMER, 1999, p.113).

2.2 Projeto Físico

A topologia física representa como as redes estão conectadas (*layout* físico) e o meio de conexão dos dispositivos de redes (nós ou nodos). A forma com que os cabos são conectados, e que genericamente são chamamos de topologia da rede (física) influência nas questões de flexibilidade, velocidade e segurança (PAULINO, 2013).

A topologia física é a verdadeira aparência ou layout da rede, tendo como redes mais populares o uso do cabo par trançado sem blindagem (UTP). Nesse modelo, há a necessidade de um dispositivo concentrador, tipicamente um s*witch*, ou *hub* para fazer a conexão entre os computadores. Quando se tem um sistema de cabeamento estruturado, é utilizado um concentrador de cabos chamado *patch panel* (painel de conexões). Ao invés de cabos que vem das tomadas conectarem-se ao *hub*, eles são conectados ao *patch panel* (MCM TECNOLOGIA).

ROSS (2007, p.12) considera que com o avanço das tecnologias, a tendência é para a necessidade de redes de altíssima velocidade, com alcances cada vez maiores e um cabeamento de cobre de alto desempenho. Existem no mercado uma variedade de categorias e classes de cabos, divididas por capacidade de desempenho.

Vale relembrar, que a topologia física é a maneira como os cabos se conectam fisicamente aos computadores. A

topologia lógica, por sua vez, é a maneira como os sinais fazem o tráfego através dos cabos e placas de rede.

Para Paulino (2013), há várias formas nas quais se pode organizar a interligação entre cada um dos nós (computadores) da rede. A topologia física é a verdadeira aparência ou layout da rede, enquanto a lógica é a descrição do fluxo dos dados através da rede. Como já dito, o projeto físico não existe sem o projeto lógico.

A qualidade da parte física do projeto, com a escolha dos itens corretos e dentro das normas, tem relação com o suporte, a segurança e durabilidade dos equipamentos.

Para Carvalho (2017) as vantagens do cabeamento físico referem-se a:

- Preço: Mesmo com a obrigação da utilização de outros equipamentos na rede, a relação custo benefício se torna algo atrativo;
- Flexibilidade: Como ele é bastante flexível, ele pode ser facilmente embutido em paredes;
- Facilidade: A facilidade com que se pode adquirir os cabos é evidente, pois em diversas lojas de informática existe esse cabo à venda;
- Velocidade: Atualmente, esse cabo trabalha com uma taxa de transferência de 100 Mbps em CAT 5 e até 10Gbps em cabos CAT6.

Carvalho (2017), também considera que para a organização dos cabos de uma empresa ou estabelecimento que utilize redes de comunicação, aderir a um sistema de cabeamento estruturado oferece mais estas vantagens:

- Facilidade para instalação de novas conexões;
- Melhoria no ambiente visual da empresa;
- Melhoria na forma de identificação dos cabos;
- Facilidade na identificação de erros na rede;
- Manutenção mais rápida;
- Melhoria no desempenho da rede;
- Melhoria na segurança da rede.

A topologia lógica em estrela, mais comum atualmente e utilizada na aplicação deste projeto, utiliza cabos de par trançado e um concentrador, como ponto central da rede. O concentrador se encarrega de retransmitir os dados para todas as estações, com a vantagem de tornar mais fácil a localização dos problemas, pois se um dos cabos, uma das portas do concentrador ou uma das placas de rede estiverem com problemas, apenas o nó ligado ao componente defeituoso ficará fora da rede.

Paulino (2013) corrobora descrevendo como vantagens do cabeamento lógico estrela:

- A codificação e adição de novos computadores é simples;
- Gerenciamento centralizado;
- Falha de um computador não afeta o restante da rede.

2.3 Cabeamento Estruturado

Cabeamento estruturado é a disposição organizada de conectores e meios de transmissão de redes de dado e voz,

internet, vídeo, alarmes, sensores, redes internas e telefonia (ZNET TECNOLOGIA, 2017).

Ainda, conforme o Znet Tecnologia, este sistema de cabeamento permite a transmissão de qualquer tipo de sinal elétrico que envolva dados, telefonia, áudio, vídeo, segurança e controles ambientais.

Vale destacar ainda as seguintes afirmações da Znet Tecnologia de que o principal papel do cabeamento refere-se a descomplicar procedimentos administrativos, de manutenção e resolução de problemas internos e externos de rede.

Todos concordam que quando as instalações de rede são feitas de forma correta e seguindo as especificações adequadas, tem vida útil de pelo menos 10 anos, suportando, além dos processos comuns, também os servidores da rede local, a quantidade de *switches*, de roteadores e suas extensões. Além disso, a confiabilidade da rede melhora muito com a implantação do cabeamento estruturado.

Para completar a importância do cabeamento estruturado, podemos destacar a sua ideia de integração dos serviços (dados e *telecom*) que foi bem definida pela Alctel Telecom (2017). Ainda, este demonstra que a capacidade do cabeamento estruturado de se redirecionar por diferentes caminhos, dentro de uma mesma estrutura, para que pontos diversos se comuniquem é uma das suas principais qualidades. (ALCTEL TELECOM, 2017).

O professor Carvalho (2015) lembra que há pouco tempo as redes eram implantadas sem muito planejamento, que somente se instalava *switches* ou *hubs* nos locais de mais necessidade, sem pensar no amanhã. Segundo ele, hoje há uma exigência maior por redes estáveis

que sejam bem estruturadas e planejadas, a fim de se compartilhar dados, voz e internet com mais rapidez e eficiência. Carvalho completa, dizendo que para possuir uma rede estável é fundamental a instalação de uma infraestrutura de cabeamento estruturado.

Como o professor Carvalho (2015) mesmo comentou, o modelo de rede mais organizado é algo novo, e as empresas brasileiras precisam se preparar. Pinheiro, já mencionava em 2003, que no Brasil prevalece o uso do cabeamento não estruturado nas empresas e indústrias, ou seja, não segue normas ou padrões, não está construído de forma organizada e muito menos preparado para receber novos equipamentos ou expansão da rede.

Atualmente, 77% das empresas dependem da Tecnologia da Informação (TI) para seguir crescendo, sendo que devemos considerar que 70% dos problemas com a rede são decorrentes do cabeamento, conforme dados da *Real Decisions Institute; Furukawa* (ZNET, 2018).

Enquanto que para o cabeamento não estruturado os sistemas de telefonia, rede e vídeo são independentes e utilizam estruturas separadas, no caso do cabeamento estruturado, todos esses recursos convergem para um único sistema de cabeamento. Nesta forma de organização, há também um planejamento detalhado dos elementos a serem utilizados, desde os locais que eles serão alocados, a divisão dos servidores, a distribuição das estações de trabalho, bem como a definição de determinados padrões e normas regulamentadas (CARVALHO, 2015).

A expansão de investimentos na área de Tecnologia da Informação (TI), incluindo também os equipamentos de rede,

está em amplo crescimento, conforme estudo da especialista Gartner. Os investimentos globais em Tecnologia da Informação (TI) tiveram, em 2018, um aumento de 4,3% em relação à 2017, sendo destaque a área de *software* e serviços corporativos de TI, além dos *devices* – equipamentos de infraestrutura, conexão/operação e *mobile*. Conforme a consultora, a linha de *devices* tende a crescer 5% este ano. Entrando nessa lista, temos os equipamentos de rede e internet, computadores de mesa, dispositivos móveis, aparelhos da linha *mobile* e toda a gama de *gadgets* que melhorem o controle das tarefas corporativas (EXAME, 2018).

3 Normas de Cabeamento estruturado

Conforme já descrito anteriormente, as empresas estão aquém de uma estruturação adequada. Apesar disso, de acordo com informações da Znet Tecnologia (2018) e Nicolau (2010), esse modelo de organização de conectores e transmissão de dados e voz (cabeamento estruturado) já havia surgido na década de 1980, com um imenso progresso nos anos 90. Nesse momento, foi preciso a criação de normas a fim de padronizar os cabos, conectores e procedimentos. Inicialmente, o conceito de cabeamento surgiu para atender os serviços de tecnologia de voz.

Carvalho (2015), esboça o objetivo de haverem normas, as quais facilitem a reorganização, a expansão e o conhecimento sobre o caminho da transmissão em qualquer parte da rede. Ele diz, que por meio das normas, é possível saber a disposição em que estão os cabos das infraestruturas, assim como a identificação e administração do cabeamento.

Para os projetos de cabeamento atuais, devem-se seguir as normas e padrões da ANSI/EIA/TIA, criadas por institutos e associações renomadas, sendo elas a IEEE[15] e EIA[16]. Atualmente, uma das normas mais conhecidas no mundo para o cabeamento estruturado é a norma TIA/EIA-568[17].

Conforme descreve o site da MCM Tecnologia (2018), esta norma americana EIA/TIA-568 é a mais utilizada internacionalmente, além de termos ISO/OSI. Na Europa, grande parte dos fabricantes utiliza o sistema IBCS[18]. Segundo a MCM, as variações que existem entre uma e outra devem-se mais às categorizações e conceitos, entretanto, tecnicamente se assemelham e se encaminham no mesmo sentido de uma arquitetura aberta, independente de protocolo. O site exemplifica dizendo que de todo modo, as novas tendências já se desenvolvem pensando no cabeamento, como é o caso do 100 BaseT, do ATM e outros.

No Brasil, para especificar um Sistema de Cabeamento Estruturado, em um único edifício ou um conjunto de edifícios comerciais é utilizada a norma **ABNT NBR 14.565**, norma brasileira que especifica os procedimentos para a elaboração de projetos de cabeamento estruturado em redes de telecomunicações (PINHEIRO, 2003, p.103).

Todos esses padrões definem os tipos de cabos utilizados, bem como os limites e requisitos (distância, segmentos, frequência)

[15]Institute of Electrical and Eletronics Engineers,

[16]Eletronics Industry Association

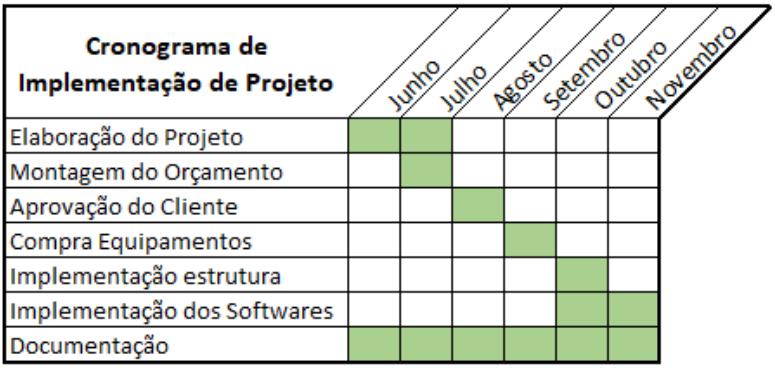
[17]Telecomunications Industry Association

[18]Integrated Building Cabling System

que a estrutura deve atender para garantir o funcionamento adequado. (ALCTEL TELECOM, 2017)

Além disso, conforme artigo de Richard Landim (2016), o não conhecimento e, portanto, o descumprimento de uma norma, podem gerar multas a qualquer empresa, a qual pode ser notificada pela Anatel[19], fiscalizadora das exigências do segmento de cabeamento: "O cumprimento das normas publicadas pela ABNT NBR-14.565 é essencial para garantir a eficácia e a segurança no sistema de cabeamento, sem colocar em risco o bem-estar público e o desenvolvimento da empresa" (LANDIM, 2016). Na sequência, podemos analisar o cronograma que guiou o projeto. Por ele observamos a sequência de implementação de cada uma das etapas.

Figura 1 - Cronograma.



Fonte. Os autores, 2018.

[19] Agência Nacional de Telecomunicações

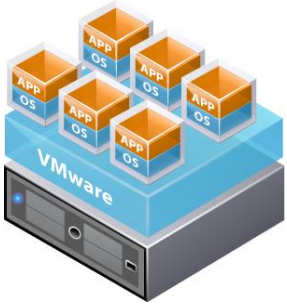
4 Virtualização

Virtualização, apesar de ser um conceito antigo, tem se mostrado cada vez mais moderno e vantajoso. O processo de virtualização consiste em separar Hardware e Software, tornando assim, sistemas operacionais e aplicações, independentes do hardware a qual estão alocados. Isso traz diversas vantagens, como a facilidade de gerenciamento, economia de energia e de hardware.

Veras e Carissimi (2015) complementam as vantagens, dizendo que esta abstração protege o hardware do acesso direto do software (aplicações e Sistema Operacional). Tal forma de separação é o que dá origem às máquinas virtuais de processo e aos monitores de máquinas virtuais (hipervisores).

Conforme a empresa desenvolvedora do sistema operacional Vsphere, a Virtualização é a emulação de um ou mais sistemas operacionais dentro de uma mesma máquina, conseguindo assim aproveitar o máximo do hardware. A virtualização é muito utilizada em máquinas de servidores dentro de empresas (VMWARE, 2018).

Figura 02 - Detalhe de exemplo de virtualização.



Fonte. VMware, 2018.

De modo conciso, um sistema de virtualização é composto por três (03) componentes: Hardware, Hipervisor e os sistemas operacionais com suas aplicações. O hardware pode ser basicamente o mesmo utilizado para os sistemas operacionais. O Hipervisor, na Figura 01, representado pela camada VMware, funciona como um intermediário entre os sistemas operacionais virtualizados e o hardware, sendo este, o responsável pela alocação adequada dos recursos disponíveis. O hipervisor consegue direcionar para cada máquina virtualizada o recurso necessário para seu pleno funcionamento, balanceando memória RAM, processamento, placa de rede, apenas quando são necessários. (VMWARE, 2018)

Veras e Carissimi (2015) também colaboram, trazendo as funções do hipervisor. Ele faz o escalonamento das tarefas, gerenciamento de memória e a manutenção preventiva da máquina virtual. Para eles, a qualidade de um hipervisor define-se pela sua escalabilidade e desempenho.

Um único hipervisor pode suportar diversas máquinas virtuais simultaneamente, desde que tenha capacidade de hardware suficiente para tanto. Os SOs (Sistemas Operacionais) virtualizados, conhecidos como VM (*Virtual Machine*) se comportam como um host qualquer da rede, sem que a virtualização seja percebida por qualquer outro usuário da rede (VMWARE, 2018).

A empresa de consultoria e pesquisas Gartner (TECNOLAN, 2018), com reconhecimento internacional em infraestrutura de TI (Tecnologia da Informação), recomenda que os líderes de Infraestrutura e Operações de TI (I&O) sigam cinco etapas: inventário, processos de gerenciamento, racionamento e renovação da infraestrutura, virtualização e operações nas

organizações. Ainda, segundo a consultoria, “as empresas geralmente iniciam a modernização de suas infraestruturas de TI gastando com novas tecnologias e talentos, quando, na realidade, deveriam antes avaliar, racionalizar e simplificar seus ativos e sistemas já existentes.”

A Etapa 4 (virtualização), orientada por Gartner, tem a finalidade de reduzir recursos. Normalmente, a virtualização é usada para que se executem mais tarefas no mesmo recurso físico, com a redução do custo total de infraestrutura de TI (Tecnologia de Informação), e assim aumentando ainda mais a eficiência da estrutura (TECNOLAN, 2018).

Na empresa Tonetto conseguimos que o hardware que antes era apenas dedicado ao servidor de arquivos, também acomodasse o sistema de gestão da empresa, com um host exclusivo para a operações bancárias, e, futuramente, um firewall com pfSense. Essa operação economizou 2 conjuntos de hardware completos, a energia elétrica para mantê-los ligados e a mão de obra para manutenção deles.

5 Instalação da Infraestrutura e métodos

No ponto inicial do projeto, a Tonetto apresentava diversas deficiências, tanto lógicas quanto físicas. Essas deficiências surgiram em decorrência de anos de crescimento da fábrica e a não existência de um setor de Tecnologia.

Figura 02 - Pátio da Tonetto no 2000.

Fonte: Arquivo Tonetto, 2000.

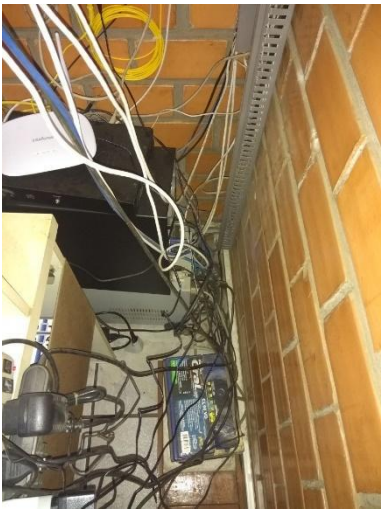
A figura 02 é o primeiro registro fotográfico da fachada da empresa Tonetto, fundada em 1994. Após seis anos, conforme figura 03, evidencia-se o início da fase de crescimento da empresa.

Figura 03. Fachada da Tonetto em 2018.

Fonte: Os autores, 2018.

Destacam-se entre os problemas tecnológicos da Tonetto, o total desconhecimento das estruturas físicas e lógicas locais, fruto de diversas instalações não planejadas e mal executadas. Na figura 04, abaixo, podemos ver um exemplo.

Figura 04 - Cabeamento não estruturado.



Fonte: Os autores, 2018.

Nos próximos subcapítulos, serão abordados temas referentes as metodologias de desenvolvimento do projeto e das especificidades dos equipamentos usados. Foi decidido, pela direção da Tonetto, que todos os equipamentos que pudessem ser reaproveitados durante o processo de implantação do projeto deveriam ser considerados, afim de economizar recursos financeiros. Portanto, a grande maioria dos equipamentos utilizados já era de propriedade da empresa e foram realocados para terem seu potencial aproveitado ao máximo.

5.1 Métodos

A pesquisa foi desenvolvida com um tratamento tecnológico. Freitas (2014) nos traz uma boa interpretação sobre o assunto, explicando que esse modelo angariaria subsídios que classificam princípios da área tecnológica, misturando, também, uma análise bibliográfica, dessa forma, desenvolve-se um quadro

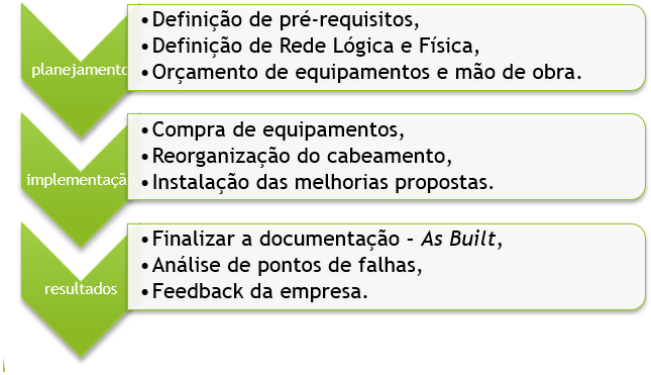
comparativo entre as duas pesquisas.

Segundo GIL (2002, p.41):

> Estas pesquisas têm como objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a constituir hipóteses. Pode-se dizer que estas pesquisas têm como objetivo principal, o aprimoramento de ideias ou a descoberta de intuições. Seu planejamento é, portanto, bastante flexível, de modo que possibilite a consideração dos mais variados aspectos relativos ao fato estudado.

Abaixo, na figura 05, pode-se conhecer o fluxo e a estrutura da execução do projeto. Esta se divide em 3 partes: planejamento, implementação e resultados.

Figura 05 - Fluxograma de etapas do desenvolvimento.



Fonte: Os autores, 2018.

Inicialmente, foi organizada uma entrevista não estruturada junto à diretoria da empresa, a fim de definir as

necessidades e prioridades da empresa Tonetto.

Dentre os mencionados estavam:

- Melhoria do sistema de telefonia;
- Aumento de largura de banda;
- Organização dos cabos atrás das mesas.

Também, nessa entrevista foi sugerida a implantação de algumas novas tecnologias, que antes eram desconhecidas para a direção da empresa, tais como: virtualização, telefonia IP, monitoramento remoto e rotinas de backup.

O projeto foi desenvolvido como um estudo de caso, com o objetivo de desenvolver uma solução adequada às necessidades deste negócio, sempre levando em consideração a melhor proporcionalidade entre custo e benefício para a empresa.

5.2 Instalação da Infraestrutura

A seguir tem-se a descrição dos materiais que foram utilizados no desenvolvimento deste projeto e os detalhes técnicos de cada equipamento utilizado. Ressalta-se que, para a reprodução deste projeto em outras empresas, não é necessária a exata fidelização dos equipamentos, desde que os substitutos tenham funcionalidades iguais ou similares.

O Primeiro passo para a implantação foi definir o layout, onde cada equipamento e cabo deve ficar, para e quais seriam comprados ou reaproveitados. Para tanto, foi elaborado um projeto de execução. Este está disponível em formato pdf pelo link https://bit.ly/2S9arRK.

5.2.1 Servidor

Equipada com um DELL – PowerEdge T130, Servidor Dell do tipo Torre, equipado com processador Intel Xeon E3-1220 v5 de 3.0 Ghz, 8Gb de memória RAM, 2 placas de rede gigalan onboard e armazenamento em raid 01 de 500Gb, a Tonetto já dispunha de um servidor suficientemente robusto para atender as suas necessidades, entretanto mesmo um equipamento robusto se mostrava bastante subestimado, estando apenas funcionando como servidor de arquivos da empresa. Este servidor executava Windows Server 2012 r2.

Com a execução do projeto optou-se por virtualizar este host com o interesse de aumentar o aproveitamento deste hardware.

5.2.2 Virtualização

Para o desenvolvimento da virtualização desse projeto um hipervisor foi selecionado após um comparativo entre os disponíveis no mercado. Comparando alguns pontos como estabilidade, fontes de pesquisa, disponibilidade, custo de implantação e manutenção, entre outros. Os analisados foram:

- ESXi 6.7
  Desenvolvido pela empresa VMware.
- Proxmox
  Distribuição baseada em GNU/Linux Debian 5.0.
- Xenserver
  Desenvolvido pela empresa Citrix.

Após esse comparativo, foi selecionado para este projeto a solução oferecida pela Vmware, o ESXi 6.7, que além de ser o líder de mercado, ainda possibilita o uso de uma versão gratuita.

Essa versão contém algumas limitações, como a restrição de apenas 32Gb de ram por host ou a impossibilidade de controlar múltiplos hardwares com o mesmo software de gestão. Mesmo com estas limitações, a aplicação não interfere no desempenho do seu uso, a empresa se encaixa nos requisitos da versão grátis.

Algumas restrições da licença do ESXi 6.7 são:

- Restringe o hardware a no máximo 32Gb de memória RAM.
- Não tem suporte ao vCenter, que centraliza as operações em diversos ESXi.
- Não tem suporte ao vCLI, que habilita o controle por linhas de comando.

Antes de realizar a implantação completa do sistema virtual na empresa, os autores realizaram testes de instalação do sistema ESXi 6.7, a criação do dispositivo com o sistema operacional do servidor da Tonetto, assim como testes de desempenho e estabilidade no laboratório 37 do Instituto Federal Catarinense – Campus Avançado Sombrio.

Após os testes, foi realizada a implementação oficial do sistema virtual no servidor da empresa, instalando e configurando o novo hipervisor e SO virtualizado no hardware. Foram realizados testes de integridade de dados do banco de dados e servidor de arquivos utilizado pela Tonetto, assim como a verificação de falhas ou quedas de uso, travamentos ou reinício do sistema. Até a publicação deste artigo, o servidor chegou a ficar online por 150 dias, sendo apenas desligado para seu reposicionamento físico.

5.2.3 Sistema de Telefonia, PABX

A função de um PABX (*Private Automatic Branch Exchange)* é interconectar os ramais telefônicos de uma empresa. Os equipamentos mais modernos também são capazes de conectar prédios diferentes como matriz e filiais, através de uma conexão com a internet (PORTAL DA EDUCAÇÃO, 2018).

O sistema PABX funciona do mesmo modo que uma central telefônica, a diferença é que estão conectados à central PABX os aparelhos da empresa administradora. Com isso facilitou a comunicação dentro da empresa. Anteriormente a empresa necessitava de várias linhas telefônicas e alugar os serviços com uma operadora de telefone, o que gerava um gasto elevado. Agora, com o sistema PABX, a empresa necessita apenas de uma linha telefônica ligada à central. Com isso, a responsabilidade pela conexão entre todos os ramais, com a linha central e com a operadora de telefone é da central PABX (LEUCOTRON, 2018).

Figura 06 - PABX Multitoc 208 instalado na Tonetto.

Fonte: Os autores, 2018.

5.2.4 CFTV

Para as questões de segurança, adotou-se uma solução nacional, a Intelbras[20], empresa brasileira que desenvolve tecnologia para diversos setores, inclusive para os setores de CFTV e alarme. Dentre as diversas soluções disponíveis, foram escolhidas as nomeadas de HDCVI 1016 G2 e AMT 2018 e para monitoramento por imagens e alarme, respectivamente.

O HDCVI 1016 V2 é um gerenciador e gravador de imagens que se destaca por ser considerado Tríbrido[21]. Capaz de

[20] http://www.intelbras.com.br/quem-somos

[21] capaz de gravar e gerenciar imagens nas três tecnologias – analógica, HDCVI e IP

lidar com as tecnologias analógicas, HDCVI e IP, o que traz grande flexibilidade aos projetos de CFTV.

"O sistema de CFTV é ferramenta importante para a segurança da empresa e monitoramento das atividades, tanto durante o dia quanto durante a noite, com isso torna-se extremamente importante que seja confiável", segundo Cláudia Tonetto, diretora da empresa.

Figura 07 - Central de monitoramento

Fonte: Os autores, 2018.

Acompanhando a solução adotada pelo sistema de CFTV, utilizou-se um produto Intelbras. Nesse caso foi selecionado o produto AMT 2018 E. Esta é a solução que apresenta maior flexibilidade dentre as opções da fabricante.

A central AMT 2018 E ainda é compatível com o DVR HDCVI 1016 G2 e habilita o modo *Gravar Tudo* na central de câmeras quando o alarme de furto ou arrombamento dispara. Também existe a possibilidade de ser integrada com os sistemas de alarme de incêndio da empresa.

5.2.5 Sistema de Nobreak

O sistema de nobreak foi implementado a fim de garantir a manutenção dos sistemas da empresa em momentos que exista falta de energia, como, também, filtrar as interferências da rede elétrica, sistema de alarme e câmeras.

Na figura 03 observa-se o nobreak fabricado pela SMS. Este modelo de equipamento em particular fornece uma potência de 3000VA e suporte para um banco de baterias estacionárias externas, que aumentam sua autonomia conforme o tamanho deste banco. Este foi o modelo escolhido para atender a empresa Tonetto.

Figura 08. Nobreak Power Vision de 3000VA

Fonte: Os autores, 2018

6 Resultados

Com a implantação do sistema de cabeamento estruturado na Tonetto conseguiu-se otimizar alguns processos, houve melhora no aproveitamento do hardware já disponível na empresa e aumento da segurança física dos equipamentos.

Foi possível através do processo de virtualização dedicar um equipamento exclusivo para a gestão dos arquivos da empresa e sistema de banco de dados. Este foi batizado de ServidorTonetto01, nele dedicou-se também um host específico para as operações bancárias da empresa, assim eliminando os problemas de conflito dos sites dos bancos com os aplicativos instalados no computador. Outro ganho considerável relacionado à virtualização é a facilidade de gerenciamento de múltiplos *hosts* em uma única interface web. Este processo facilitou bastante o acesso à infraestrutura, tanto pela intranet quanto pela internet. Pode-se marcar como relevante o sistema de gerência de *switches* virtuais. Estes possibilitam que se conecte máquinas virtuais em redes logicamente diferentes, assim é possível que em projetos futuros crie-se redes distintas, conforme as necessidades surjam.

O novo sistema de segurança da empresa Tonetto trouxe tranquilidade e conforto para a empresa e seus diretores. O monitoramento remoto possibilitou que uma visão da empresa em tempo real esteja disponível nos computadores e celulares dos responsáveis, não sendo mais necessário se deslocar até a empresa cada vez que o alarme dispara. Os disparos em falso, apesar de menos frequentes, ainda acontecem, seja por conta de aves que voam no galpão ou o vento que derruba algum objeto. Algumas vezes estes eventos ocorrem durante a madrugada.

6.1 Resultado da Organização em RACK

Com a aquisição do rack e a reorganização dos equipamentos já existentes na empresa, agora, esta conta com uma melhor disposição e acesso a todos os dispositivos em um mesmo lugar, melhorando também a segurança, eliminando o uso de uma

tomada para vários equipamentos, diminuindo assim o risco de curto circuitos e incêndios.

Figura 09 - Vista geral do Rack montado.

Fonte: Os autores, 2018.

Olívia Tonetto, diretora financeira, relata que a organização dos equipamentos de TI trouxe um ambiente mais organizado ao escritório. Como pode ser visto a figura 09 e 10.

Figura 10 - Detalhe do Switch e patch panel.

Fonte: Os autores, 2018.

Além de melhorar o visual estético da sala de equipamentos, com a utilização do rack, cessaram os problemas de cabos desligados. Os equipamentos não ficam mais sobrepostos e empilhados um em cima do outro. Agora, quando ocorre uma falha de conexão na rede, ficou mais fácil e rápido detectar qual cabo está com problema, pois ao ser refeito o cabeamento estruturado, foi realizado a marcação dos cabos ponta a ponta.

6.2 Resultado da melhoria na Telefonia e internet

Todo o cabeamento telefônico foi reestruturado, assim como a marcação dos cabos, que além de facilitar na detecção e realização de manutenção nos cabos defeituosos, reduziu o tempo de espera do funcionário sem o equipamento, como também o tempo do técnico para troca do cabo ou equipamento.

Também foi proposto a contratação de um link com IP Fixo e o aumento da largura de banda para 25MB.

Mesmo que para os funcionários internos o aumento da banda da internet não tenha grande significado na questão de velocidade de acesso ao sistema, para os gestores, o aumento foi significativo na utilização do acesso ao sistema externamente, diminuindo a lentidão no acesso, quedas frequentes por tempo limite esgotado para resposta no acesso remoto.

Agora a empresa conta com acesso remoto tanto para o sistema de gerenciamento de dados da mesma, quanto para acesso às informações do servidor, obtendo o status se o mesmo está online ou off-line, assim como acesso a central de monitoramento das câmeras e a central de alarme da empresa.

7 Considerações finais

Após seis meses dedicados entre desenvolvimento, implantação e documentação fica claro que a Tonetto não conseguia empregar bem os equipamentos que já eram de sua propriedade. Também, foi reconhecido que a estrutura da empresa tendia ao colapso e em pouco tempo haveria graves problemas, tecnológicos ou mesmo físicos, já citados anteriormente.

Com a implantação da nova infraestrutura, os equipamentos ganham sobrevida e passam a trabalhar mais próximo do seu máximo ideal, trazendo para toda a empresa mais recursos e seguranças.

Uma empresa que entende e investe em TI e cabeamento estruturado garante não apenas a segurança e prevenção contra problemas técnicos, que podem deixar o sistema inoperante, mas também que sua infraestrutura funcione a longo prazo e esteja pronta para novas tecnologias.

Possuir uma rede de transmissão de dados interna e externa eficiente e rápida pode garantir uma vantagem competitiva no meio corporativo. A restrição orçamentária imposta pela empresa fora um obstáculo considerável a completa implantação do projeto. Estas foram contornadas da melhor forma encontrada.

Fora priorizada a estabilidade da rede e pleno desenvolvimento no dia-a-dia de trabalho. Itens como a Gerência de Redes, certificação dos pontos de rede e instalação de um firewall foram eleitos para implementação em 2020.

O ponto que não foi implantado, e é de suma importância, trata-se de um sistema de backup automatizado, que trará ainda mais segurança à infraestrutura, está previsto para implantação ainda em 2019.

Referências

ALCTEL (Bh). **Entenda o que é cabeamento estruturado e sua importância.** 2018. Disponível em: <https://www.alctel.com.br/blog/entenda-o-que-e-cabeamento-estruturado-e-sua-importancia/>. Acesso em: 21 nov. 2018.

CARVALHO, Paulo. **Cabeamento estruturado, quais são as suas vantagens reais?** 2017. Disponível em: <https://tecnocopa.com.br/2017/09/12/o-que-e-cabeamento-estruturado-e-quais-sao-as-suas-vantagens/>. Acesso em: 21 nov. 2018.

DATACENTERDYNAMICS (Brasil). **Previsão da IDC para o mercado de TIC no Brasil em 2018 aponta crescimento de 2,2%**. 2018. Disponível em:

<http://www.datacenterdynamics.com.br/focus/archive/2018/02/previsão-da-idc-para-o-mercado-de-tic-no-brasil-em-2018-aponta-crescimento-de->. Acesso em: 01 nov. 2018.

DE CARVALHO, Prof. Dr. Marcilio Bergami. **Curso Cabeamento Estruturado**. 2018. Disponível em: <https://www.cpt.com.br/cursos-informatica-redesdecomputadores/cabeamento-estruturado>. Acesso em: 01 nov. 2018.

EXAME. **Com mercado de US$ 3 ,7 trilhões em 2018, tecnologia é o investimento da vez, diz especialista**. 2018. Disponível em: <https://exame.abril.com.br/negocios/dino/com-mercado-de-us-3-7-trilhoes-em-2018-tecnologia-e-o-investimento-da-vez-diz-especialista/>. Acesso em: 21 nov. 2018.

FALQUEIRO, Tiago. **Mercado brasileiro de cabeamento estruturado mantém tendência de crescimento**. 2012. Disponível em: <http://www.datacenterdynamics.com.br/focus/archive/2012/08/mercado-brasileiro-de-cabeamento-estruturado-mant%C3%A9m-tend%C3%AAncia-de-crescimento>. Acesso em: 21 nov. 2018.

FREITAS JUNIOR, Vanderlei. **A pesquisa científica e tecnológica**. 2014. Disponível em: <https://scholar.google.com.br/citations?user=c9LCOxIAAAAJ&hl=pt-BR#d=gs_md_citad&p=&u=%2Fcitations%3Fview_op%3Dview_citation%26hl%3Dpt-BR%26user%3Dc9LCOxIAAAAJ%26citation_for_view%3Dc9LCOxIAAAAJ%3AufrVoPGSRksC%26tzom%3D120>. Acesso em: 21 nov. 2018.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 4 ed. Ed. São Paulo: Atlas, 2002.

ICMP CONSULTORIA EM TI (São Paulo). **O que é cabeamento estruturado e quais são suas vantagens?** 2017. Disponível em: <https://www.icmpconsultoria.com.br/single-post/2017/09/04/O-que-e-cabeamento-estruturado-e-quais-sao-suas-vantagens>. Acesso em: 21 nov. 2018.

IDC BRASIL (Brasil). **IDC Brasil prevê retomada de projetos em 2017 e crescimento de cerca de 2,5% para o mercado de TIC**. 2017. Disponível em: <http://br.idclatin.com/releases/news.aspx?id=2129>. Acesso em: 21 nov. 2018.

IT UNIVERSE TECNOLOGIA (Sp). **Cabeamento Estruturado**. 2018. Disponível em: <http://www.ituniverse.com.br/cabeamento-estruturado/>. Acesso em: 21 nov. 2018.

LANDIM, Richard. **Norma brasileira para cabeamento estruturado auxilia correta execução de projetos.** 2016. Disponível em: <https://www.itforum365.com.br/tecnologia/norma-brasileira-para-cabeamento-estruturado-auxilia-correta-execucao-de-projetos/>. Acesso em: 21 nov. 2018.

LEUCOTRON. **Sistema PABX: entenda os diferentes tipos e seus benefícios.** 2018. Disponível em: <https://blog.leucotron.com.br/sistema-pabx-entenda-os-diferentes-tipos-e-seus-beneficios/>. Acesso em: 01 nov. 2018.

LEUCOTRON. **Você sabe como funciona um sistema PABX?** 2018. Disponível em: <https://blog.leucotron.com.br/voce-sabe-como-funciona-um-sistema-pabx/>. Acesso em: 01 jan. 2018.

MARCONDES, José Sérgio. **Sistemas de Alarme da Segurança Eletrônica: Conceitos, Equipamentos.** 2018. Disponível em: <https://www.gestaodesegurancaprivada.com.br/sistemas-de-alarme-da-seguranca-eletronica/>. Acesso em: 01 nov. 2018.

MCM TECNOLOGIA (Manaus). **Cabeamento Estruturado**. 2017. Disponível em: <http://mcm.com.br/v04/index.php/servicos/cabeamento-estruturado>. Acesso em: 01 nov. 2018.

MICHAEL HAAG (Eua). **Infraestrutura Hiperconvergente para Leigos.** 2. ed. Nova Jersey: Wiley, 2018. 76 p.

MONTEIRO, João. Gartner: **Chegou a hora do mercado de infraestrutura voltar a crescer no Brasil**. 2018. Disponível em: <https://ipnews.com.br/gartner-chegou-hora-do-mercado-de-infraestrutura-voltar-crescer-no-brasil/>. Acesso em: 21 nov. 2018.

NICOLAU, Marcel. **A importância de um Sistema de Cabeamento Estruturado (SCE) nas empresas.** 2010. Disponível em: <https://www.profissionaisti.com.br/2010/12/a-importancia-de-um-sistema-de-cabeamento-estruturado-sce-nas-empresas/>. Acesso em: 21 nov. 2018.

NOVELLO, André. **O que é Virtualização?** 2017. Disponível em: <https://andrenovello.wordpress.com/2017/01/28/o-que-e-virtualizacao/>. Acesso em: 01 nov. 2018.

OPPENHEIMER, Priscilla. **Projeto de redes Top-Down: Um enforque de análise de sistemas para o projeto de redes empresariais**. 2. ed. Rio de Janeiro: Campus Ltda, 1999.

PAULINO, Daniel. **Topologia de Redes**. 2013. Disponível em: <https://www.oficinadanet.com.br/artigo/2254/topologia_de_redes_vantagens_e_desvantagens>. Acesso em: 01 nov. 2018.

PORTAL EDUCAÇÃO (Brasil). **O que É PABX?** 2018. Disponível em: <https://www.portaleducacao.com.br/conteudo/artigos/conteudo/o/27979>. Acesso em: 01 nov. 2018.

RODRIGUES, André. **Redes Lógicas**. 2017. Disponível em: <https://www.portalgsti.com.br/2017/07/redes-logicas.html>. Acesso em: 01 nov. 2018.

ROSS, Julio. **Cabeamento Estruturado**. Rio de Janeiro: Antenna Edições Técnicas Ltda, 2007. 48 p.

TECNOLAN (Sp). **5 recomendações do Gartner em infraestrutura e operações.** 2018. Disponível em: <http://www.tecnolan.com.br/5-recomendacoes-do-gartner-em-infraestrutura-e-operacoes/>. Acesso em: 21 nov. 2018.

TONETTO, Cláudia e Olívia. Entrevista concedida a Joao Azevedo e Rafael Nascimento, 9 de Jun de 2018.

VERAS, Manoel; CARISSIMI, Alexandre. **Virtualização de Servidores.** 2015. Disponível em:

<https://pt.scribd.com/doc/50570155/Virtualizacao-de-Servidores>. Acesso em: 02 dez. 2018.

VMWARE. **O que é virtualização?** 2018. Disponível em: <https://www.vmware.com/br/solutions/virtualization.html>. Acesso em: 01 nov. 2018.

ZNET TECNOLOGIAS (Mg). **O que é Cabeamento Estruturado**. 2018. Disponível em: <https://znet.net.br/blog/cabeamento-estruturado>. Acesso em: 21 nov. 2018.

Implementação de Servidor de Arquivos e Autenticação para alunos no IFC – Campus Avançado Sombrio

João Francisco Dossa [1], Maico Trein Müller [1], Jéferson M. de Limas [2], Victor Martins de Sousa[2]

[1]Acadêmicos do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88960000 – Sombrio – SC – Brasil

[2]Docentes do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88960000 – Sombrio – SC – Brasil

{jfdossa, maycomuller}@hotmail.com,
{jeferson.limas, victor.sousa}@ifc.edu.br

Abstract. *This article describes the installation and configuration of a file server using Samba 4, a free software, that makes it possible to create file and printer sharing for environments that have Windows and Linux machines. It is beeing a tool at no cost, after 10 years of development it is in its stable version. Version 4 of Samba, besides maintaining compatibility with previous versions, has the Directory Service. Based on the Lightweight Directory Access Protocol (LDAP)l, meaning lightweight directory access protocol, it is the main protocol for Active Directory development. This application was tested in virtualized environment (VirtualBox) and physically (single server). The use of this type of Directory Service can be implemented in different environments, because it is a free tool that generates the reduction of costs, with respect to licenses of use of software and Operating*

System. It can influences the economy as a whole with respect to solutions in Information Technology.

Resumo. *Este artigo descreve a instalação e configuração de um servidor de arquivos utilizando Samba 4, um software livre, que possibilita criar o compartilhamento de arquivos e impressoras para ambientes que possuem máquinas Windows e Linux. Trata-se de uma ferramenta sem custo, que após 10 anos de desenvolvimento, encontra-se em sua versão estável. A versão 4 do Samba, além de manter a compatibilidade com as versões anteriores, possui o Serviço de Diretório. Tendo como base o protocolo Lightweight Directory Access Protocol (LDAP), com significado de protocolo leve de acesso a diretórios, sendo o principal protocolo para o desenvolvimento do Active Directory. Esta aplicação foi testada em ambiente virtual (VirtualBox) e físico (servidor único). A utilização desse tipo de Serviço de Diretório pode ser implementado em diferentes ambientes, por tratar-se de uma ferramenta livre, gerando, assim, a redução de custos com relação à licenças de uso de software, influenciando na economia como um todo no que diz respeito à soluções em Tecnologia da Informação.*

1 Introdução

As redes de computadores, instaladas em diversos ambientes, permitem o compartilhamento de informações através de meios digitais. Com isso, foi possível perceber a necessidade de organizar essas informações de forma a manter o controle e o acesso otimizado (MENDES, 2007).

No IFC – Campus Avançado Sombrio, os alunos dispõem de laboratórios de informática com o Sistema Operacional Ubuntu 18.04. Para o uso dos alunos existe um

usuário comum a todos (aluno) e um usuário do pessoal de TI para manutenção (suporte). Neste cenário os alunos compartilham o mesmo repositório de arquivos, o que ocasiona, em diversos momentos, a exclusão equivocada de arquivos, não sendo possível identificar o período em que ocorreu e nem quem foi o responsável pela exclusão.

A implementação do Samba permite que os usuários possam compartilhar seus recursos locais de software e hardware com algum controle e segurança de seus dados. O Samba, na sua versão atual, oferece suporte à estrutura de controle de domínio, que se utiliza de um serviço de diretórios para armazenar os dados, possibilitando ao usuário compartilhar e gerenciar os recursos de rede (SAMBA, 2018).

De acordo com o Samba (2018), tal software fornece serviços de arquivo, sendo um componente importante para integrar servidores e desktops Linux em ambientes do Active Directory (AD). Atualmente, se encontra na versão 4. Nessa versão, o Samba contém suporte a um servidor de diretório Lightweight Directory Access Protocol (LDAP), um protocolo responsável por definir um método para o acesso e a atualização de informações em um diretório (espécie de banco de dados, otimizado para leitura e busca) (LENDECKE, 2004).

Desta forma, este trabalho tem como objetivos implementar um Serviço de Diretório com Autenticação Centralizada e implantar os serviços de um Controlador de Domínios (PDC) e Servidor de Arquivos no Instituto Federal Catarinense Campus Avançado Sombrio, permitindo um controle de acesso aos dados disponibilizados.

Esse trabalho consiste em uma implementação realizada no IFC – Campus Avançado Sombrio, onde foi possível implementar recursos para uma utilização da rede de forma segura e com acesso controlado.

2 Active Directory (AD)

O Active Directory (AD) é uma ferramenta da Microsoft utilizada para o gerenciamento de usuários de rede, denominada Serviço de Diretório. O AD facilita a pesquisa e a autenticação, pois permite o armazenamento das informações de forma organizada, facilitando a recuperação dos dados (STANEK, 2009).

Segundo Metzmacher (2007), vários protocolos são utilizados para o seu funcionamento, sendo o principal deles o Protocolo de Acesso a Diretórios Leves (LDAP).

Através da implementação de serviço LDAP, o Active Directory permite o uso de um único diretório para controle de acesso a todos sistemas e serviços dentro de uma rede corporativa. Isso significa que o colaborador de uma empresa não precisa criar um usuário e senha para cada sistema que tiver acesso, e sim utilizar um único usuário e senha (DOMINGUES, 2011).

3 Server Message Block (SMB)

Segundo o Samba (2018), smb é um protocolo de compartilhamento de arquivos em rede, que permite que os aplicativos de um computador leiam e gravem em arquivos e solicitem serviços dos programas do servidor em uma rede de computadores.

O protocolo SMB pode ser usado sobre os protocolos TCP/IP ou outros protocolos de rede. Conforme especificado na RFC1001 e RFC1002, após estabelecer uma conexão, os usuários podem enviar comandos (SMBs) para o servidor, permitindo que eles acessem compartilhamentos, arquivos ou outros recursos em um servidor remoto. Também, permitindo que os usuários leiam, criem e atualizem arquivos no servidor remoto (SHARPE, 2002).

Para realização deste trabalho, foi utilizado o Samba por ser um software de código aberto, seu código pode ser alterado e aprimorado para atender às necessidades dos usuários. Sua nova versão traz a integração completa com Active Directory e a inclusão da ferramenta de administração de servidor remoto (RSAT). Ele permite uma abordagem flexível da segurança de rede. O servidor Samba para Linux implementa os protocolos mais atualizados, para permitir operações de arquivo e impressão mais seguras no cliente Windows (OLIVEIRA, 2016).

O Samba pode ser gerenciado através da interface gráfica do Windows, mesmo que o servidor seja Linux. Ele traz DNS, *kerberos* e LDAP integrado. Existem várias fontes de suporte às quais pode-se recorrer quando for necessário solucionar dúvidas e dificuldades referentes ao Samba, como fóruns e sites na internet. Ele oferece uma alternativa de excelente desempenho e com todos os recursos necessários para configurar uma rede, quer seja uma rede com poucos ou vários usuários. (OLIVEIRA, 2016).

4 Samba

Samba é um servidor e um conjunto de ferramentas que permite que máquinas Linux e Windows se comuniquem entre si, compartilhando serviços (arquivos, diretório, impressão) através do protocolo Server Message Block (SMB) / Common Internet File System (CIFS) (MORIMOTO, 2013).

Os dois principais programas do Samba são smbd e nmbd. Tendo como funções de trabalho a implementação dos quatro serviços básicos CIFS: serviços de arquivo e impressão, autorização e autenticação, resolução de nomes e anúncio de serviço de navegação (SAMBA, 2001).

Com o Samba, é possível construir domínios completos, fazer controle de acesso em nível de usuário, compartilhamento, configurar um servidor WINS, servidor de domínio e impressão. Normalmente, o controle de acesso e exibição de diretórios no samba é mais minucioso e personalizável que no próprio Windows (MORIMOTO, 2013).

Um compartilhamento no Samba pode ser acessível publicamente (sem senha) ou estar protegido, dificultando o acesso ao seu conteúdo, como senhas, endereço de origem, interfaces, usuários autorizados, permissões de visualização e modificação (MORIMOTO, 2013).

Além de fornecer o compartilhamento de arquivos e impressoras em sistemas Windows e Linux, se for utilizado em um servidor Linux, o Samba também irá se comportar exatamente como se fosse um servidor Windows, sendo possível trabalhar com autenticação dos usuários e também compartilhar impressoras. A utilização do Samba vem aumentando, tendo em vista que se trata de uma ferramenta *open source*, o que favorece empresas que necessitam de um serviço e buscam por uma redução de custos (BRITO, 2017).

5 Metodologia

Para os estudos, análise e implementação desse projeto, utilizou-se de uma pesquisa tecnológica, responsável pelo desenvolvimento de teorias de aplicações que buscam a solução de problemas pontuais e específicos na área da tecnologia. Ao contrário da pesquisa científica, que se limita à teoria, o conhecimento tecnológico trabalha com a tarefa, visando sempre à criação de algo novo (ANDERLE et al, 2017).

Segundo (ANDERLE et al, 2017), “a pesquisa tecnológica deve valer-se cada vez mais de enunciados e

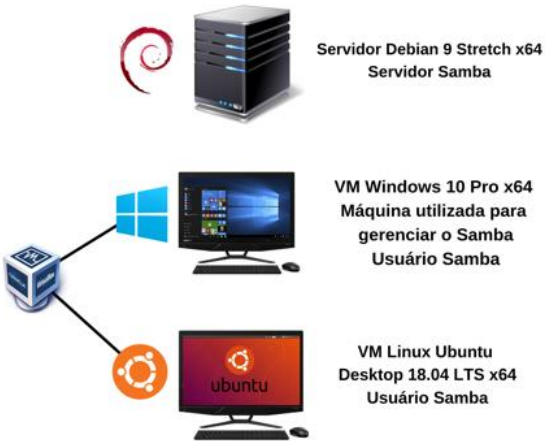
métodos científicos para dar-lhes a segurança necessária para o avanço consciente da inovação e da própria tecnologia.”

O embasamento teórico de pesquisa foi baseado em livros, artigos e sites, possibilitando a coleta de informações importantes utilizadas como fundamentação teórica no desenvolvimento do artigo.

6 Materiais

O ambiente utilizado para fins de estudos e práticas deste artigo é baseado em máquina real e virtual. Utilizou-se o software Oracle VM VirtualBox – versão 5.2.10, que é uma ferramenta gratuita que possibilita a simulação de computadores reais em máquinas virtuais, como demonstrado na Figura 1.

Figura 1 – Sistemas Operacionais



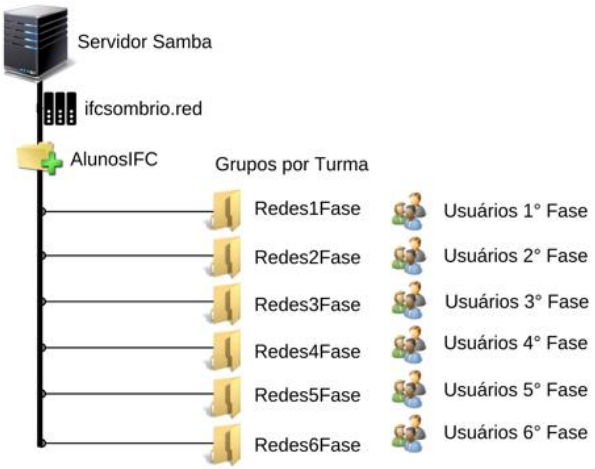
Fonte: Os Autores, 2018

7 Modelo Proposto

A implementação deste artigo será realizada em uma máquina real utilizada como servidor e serão realizados testes de autenticação em máquinas virtuais. Poderá ser utilizada por alunos e professores, que carecem de um Serviço de Diretório, disponível para sua rede local, com a vantagem de ser implementado em software livre.

Na Figura 2, é apresentado o modelo proposto para implementar uma rede local, no IFC Campus Avançado Sombrio, conforme discutido com professores do curso de Redes.

Figura 2 – Implementação Samba 4.0



Fonte: Os Autores, 2018

8 Instalações

Para instalação do servidor, utilizamos uma instalação mínima do sistema operacional Debian 9 Stretch, deixando uma imagem

limpa, evitando problemas com conflito de configurações. A primeira etapa consiste em realizar a atualização de repositórios e pacotes do sistema, e instalação de suas dependências. Em seguida, fazer o download e instalação do pacote Samba em sua versão mais recente, para que posteriormente seja possível configurar como controlador de domínio. Também é necessário instalar outros serviços como, por exemplo, *winbind* e *kerberos*.

Antes de realizar qualquer configuração do domínio por meio do provisionamento do Samba, precisa-se parar alguns serviços, que por padrão são automaticamente executados logo após a instalação dos pacotes, e remover o arquivo original de configurações do Samba. Para a realização do provisionamento do domínio foram utilizadas ferramentas automatizadas do próprio Samba, que serão responsáveis por preparar o servidor como controlador do domínio ifcsombrio.red, conforme pode ser visto no comando do Quadro 1.

Quadro 1 – Comando utilizado para realização do provisionamento do domínio ifcsombrio.red

```
samba-tool domain provision --use-rfc2307 –
realm=ifcsombrio.red --domain=ifcsombrio --dns-
backend=SAMBA_INTERNAL --adminpass=@123teste
--server-role=dc --function-level=2008_R2
```

Fonte: Os Autores, 2018

A implantação dos serviços foi realizada, primeiramente, em máquinas virtuais para realização de testes e validação das configurações que seriam utilizadas posteriormente no servidor físico localizado no IFC.

Para auditar o servidor de arquivos, foi utilizado um recurso do próprio Samba chamado “*full_audit*”, este recurso

serve para registrar ações do usuário em *log*, como por exemplo a criação e exclusão de arquivos.

Para um melhor entendimento das configurações utilizadas, o Quadro 2 demonstra de forma simples e objetiva a configuração de auditoria utilizando o recurso "*full_audit*" do próprio Samba.

Quadro 2 – Configuração principal da Auditoria e Armazenamento de exclusão no Samba.

```
(...)

#Auditoria de Arquivos

full_audit:success = open opendir write rename mkdir rmdir
chmod chown unlink

full_audit:prefix = %U|%I|%S

full_audit:failure = none

full_audit:facility = local5

full_audit:priority = notice

(...)
```

Fonte: Os Autores, 2018

Em relação ao armazenamento de exclusões, é utilizado o recurso "*recycle*", onde o arquivo excluído é encaminhado para uma pasta "lixeira" possibilitando a recuperação do mesmo através do administrador.

As configurações responsáveis pelo armazenamento de exclusão dos arquivos e da pasta “lixeira” onde estes arquivos serão armazenados podem ser vistas no Quadro 3.

Quadro 3 – Configuração do armazenamento de exclusão e pasta responsável de armazenamento dos arquivos.

```
(...)

#Lixeira

recycle:keeptree = yes

recycle:versions = yes

recycle:repository = /data/trash

recycle:exclude = *.~*, ~*.*, *.bak, *.old, *.iso, *.tmp

recycle:exclude_dir = temp, cache, tmp

### Lixeira ###

[lixeira]

path = /data/trash

writeable = yes

read only = no

browseable = no

vfs objects = scannedonly
```

Fonte: Os Autores, 2018

O Quadro 4 demonstra de forma clara o compartilhamento de grupos por fase, no curso de Redes de

Computadores, onde cada aluno pertence ao grupo relacionado com sua fase, estas configurações não são permanentes, podendo ser feitas alterações caso necessário.

Quadro 4 – Compartilhamento de grupos por fases.

```
(...)

### Compartilhamento por Grupos #####

[Redes1Fase]

path = /data/Redes1Fase

read only = no

vfs objects = full_audit, recycle



[Redes2Fase]

path = /data/Redes2Fase

read only = no

vfs objects = full_audit, recycle



[Redes3Fase]

path = /data/Redes3Fase

read only = no
```

```
vfs objects = full_audit, recycle

[Redes4Fase]

path = /data/Redes4Fase

read only = no

vfs objects = full_audit, recycle


[Redes5Fase]

path = /data/Redes5Fase

read only = no

vfs objects = full_audit, recycle


[Redes6Fase]

path = /data/Redes6Fase

read only = no

vfs objects = full_audit, recycle
```

Fonte: Os Autores, 2018

A configuração apresentada no Quadro 5 demonstra o compartilhamento da pasta “dados”, responsável por armazenar arquivos de forma que todos tenham acesso e também do diretório “*home*”, mantendo o propósito de oferecer uma pasta particular para o usuário salvar seus arquivos.

Quadro 5 – Configuração da pasta compartilhada e privada.

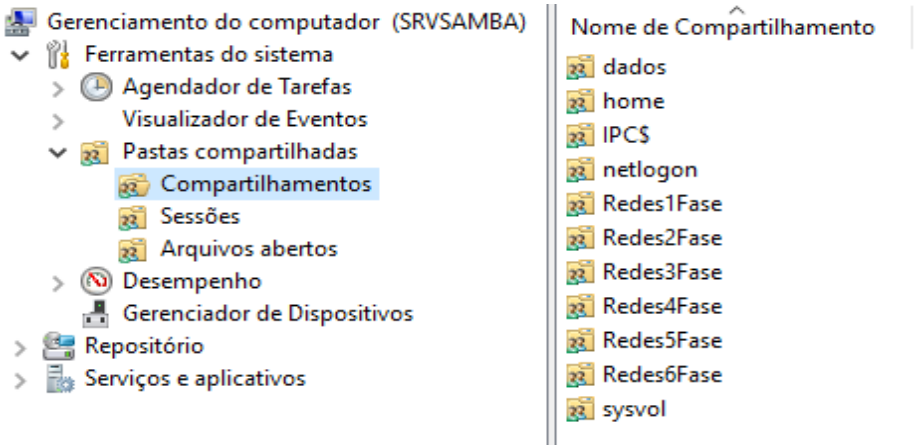
```
(...)

#### Compartilhamento Geral #####

[dados]

path = /dados

read only = No

browseable = yes

vfs objects = full_audit, recycle



### Pasta Pessoal ####

[home]

path = /data/home

read only = no

(...)
```

Fonte: Os autores, 2018

9 Resultados

A criação e compartilhamento do diretório "*home*" no servidor, tem como finalidade ser utilizado como um diretório "pessoal" para cada usuário, no qual somente o proprietário e o administrador terão acesso. Esse recurso é conhecido como "Pasta Base". A Figura 3 ilustra este diretório e os grupos de cada fase do curso de Redes de Computadores.

Figura 3 – Compartilhamento das pastas dos usuários.



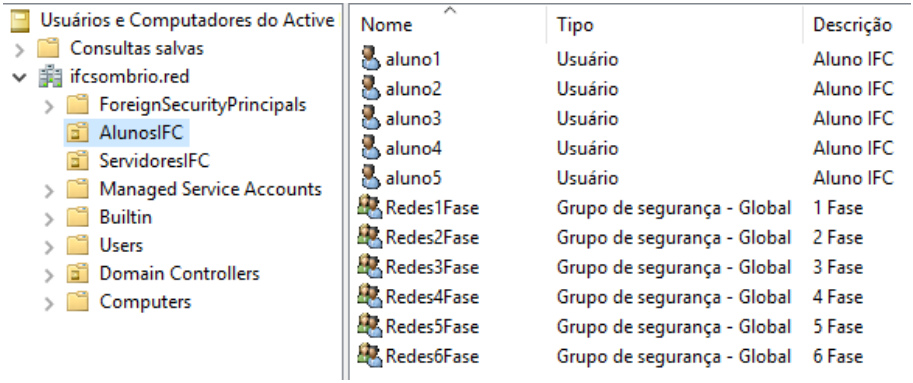
Fonte: Os autores, 2018

Dentro da concepção de um Serviço de diretório, um grupo é um conjunto de usuários e computadores que podem ser gerenciados como uma única unidade. Na figura 4 podemos observar os usuários e os grupos criados para cada fase no curso de Redes de Computadores, onde determinado usuário pertence ao grupo correspondente a sua turma atual, sendo possível compartilhar informações apenas entre os usuários desse mesmo grupo.

O mapeamento é a ação necessária para simplificar o acesso ao compartilhamento, onde é possível acessar o

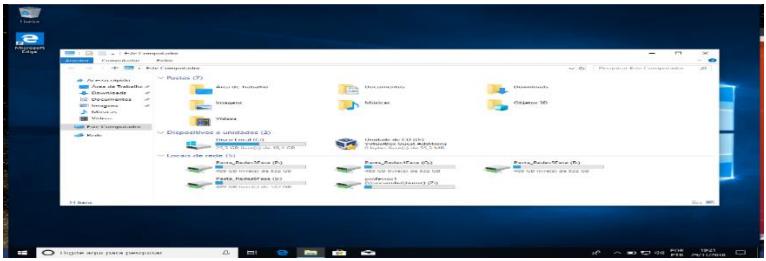
compartilhamento do “Meu Computador” como se fosse um disco rígido local. Conforme a Figura 5, é possível perceber os mapeamentos criados no sistema operacional Windows, onde o usuário “professor1” pertence a vários grupos e mesmo assim possui sua pasta pessoal.

Figura 4 – Usuários e Grupos criados.



Fonte: Os Autores, 2018

Figura 5 – Mapeamento dos grupos e pasta pessoal

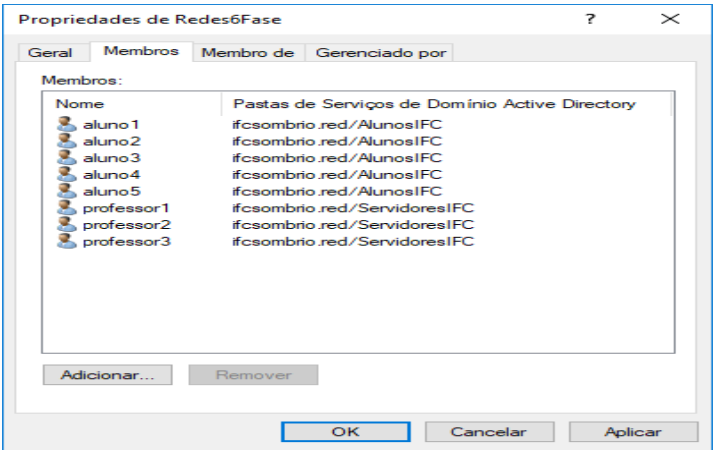


Fonte: Os Autores, 2018

Os grupos em diretórios facilitam o trabalho do administrador de redes, através de atribuição de permissões em um recurso compartilhado, atribuição de direitos de usuário a um

determinado grupo, sendo possível um usuário ser membro de vários grupos. Na figura 6 é possível observar as propriedades do grupo Redes6Fase, onde é possível identificar quais usuários pertencem a esse grupo.

Figura 6 – Propriedades do grupo Redes6Fase



Fonte: Os Autores, 2018

O mapeamento é usado quando se deseja ter o controle do acesso aos arquivos. Isso permite que cada arquivo e diretório tenha acesso à leitura e gravação somente para usuários definidos e autenticados no controlador de domínio.

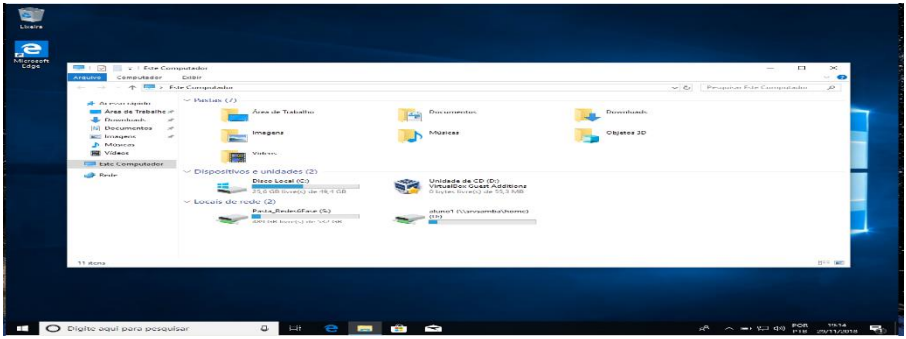
Para a criação de mapeamento automático e inclusão do Ubuntu no domínio ifcsombrio.red, foi necessária a utilização da ferramenta CID. Essa ferramenta trata-se de um conjunto de Scripts elaborados em *Shell Script* que fazem as modificações no sistema GNU/Linux, necessárias à inclusão do PC no domínio AD. Outra ação do CID é permitir de maneira simplificada configurar o *pam-mount* (solução para mapeamento de unidade de rede) de forma centralizada, viabilizando a

montagem automática das unidades de rede durante o login do usuário através do arquivo *shares.xml*, onde deverão ser definidos os mapeamentos para toda a rede no domínio.

No Windows, para que fosse possível mapear a unidade de rede com base nos grupos aos quais o usuário pertence, foram realizadas pesquisas em sites de internet, e a melhor solução encontrada foi a utilização de scripts em Visual Basic. Com base no exemplo de Russo (2007), montou-se o script “mapear.vbs”, que deve ser chamado dentro de logon.bat.

A Figura 7 demonstra o mapeamento no Windows, onde apenas o grupo o qual o usuário “aluno1” pertence e sua pasta pessoal são visíveis em seu computador.

Figura 7 – Mapeamento do grupo o qual o usuário “aluno1” pertence e de sua pasta pessoal.



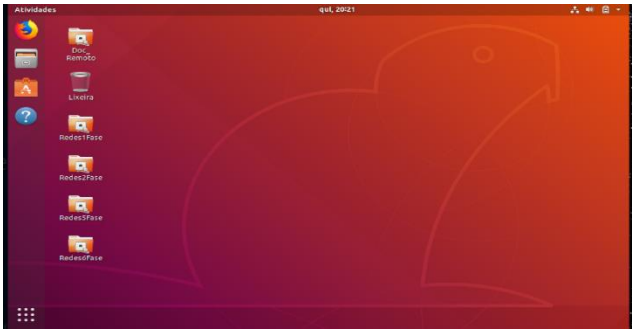
Fonte: Os Autores, 2018

O mapeamento das pastas referentes ao usuário e os grupos não se diferenciam no Ubuntu.

Na Figura 8 é possível perceber que o compartilhamento ficou disponível diretamente na área de trabalho, permitindo ao usuário “professor1” ter acesso fácil e prático a todos grupos,

podendo assim gerenciar da melhor forma o compartilhamento de arquivos.

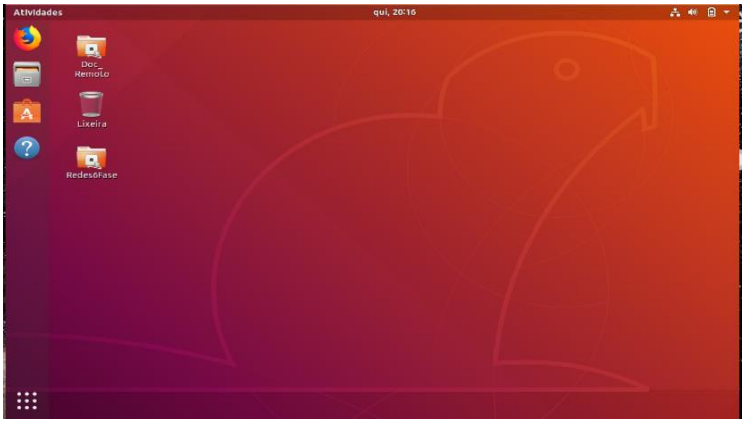
Figura 8 – Mapeamento das pastas do usuário “professor1”



Fonte: Os Autores, 2018

O mapeamento das pastas do usuário “aluno1” no Ubuntu também pode ser encontrado na área de trabalho, como ilustrado na Figura 9.

Figura 9 – Mapeamento das pastas do usuário “aluno1”



Fonte: Os Autores, 2018

10 Considerações Finais

O desenvolvimento do presente estudo possibilitou uma análise de como é necessário disponibilizar um serviço de diretório no IFC-Instituto Campus Avançado Sombrio, permitindo que diferentes tipos de informações sejam manipuladas com diferentes necessidades sobre os arquivos armazenados.

Este trabalho demonstrou a possibilidade de utilização desse serviço, que pode ser configurado e integrado com outras ferramentas, de modo que melhor se adeque aos mais diversos cenários.

A solução apresentada neste trabalho, portanto, está apta para ser implementada de modo que possam ser realizados testes mais aprofundados, sendo implementados em ambiente de estudo para que os alunos e servidores do Instituto avaliem com calma as possibilidades de utilização dos serviços que serão disponibilizados.

Houve alguma dificuldade no desenvolvimento do projeto, em relação ao material para fundamentação teórica sobre a autenticação centralizada utilizando Samba 4 AD. Sendo que o material para fundamentar e implementar foi basicamente encontrado em artigos e no site oficial do projeto. Em relação à configurações no ambiente, a utilização da máquina virtual com o sistema operacional Windows mostrou-se bem simples na questão da configuração, sendo adiciona ao domínio sem maior dificuldade, diferente da VM utilizada pelo Ubuntu, que se mostrou um pouco complexa, sendo necessário buscar outras ferramentas para permitir sua adição no domínio.

Este trabalho apresentou uma solução para autenticação única sem considerar os aspectos de segurança que envolvem um ambiente de rede. Uma proposta para trabalhos futuros é a implementação de uma solução de segurança e backup dos dados armazenados no servidor.

Referências

ANDERLE, D. F.; JUNIOR, V. F.; NAKAYAMA, M. K.; SPERONI, R.; WOSZEZENKI, C. (2017) REVISTAESPACIOS.COM. **Design Science Research Methodology Enquanto Estratégia Metodológica para a Pesquisa Tecnológica,** v.38, n.6, 2017. Disponível em: <http://www.revistaespacios.com/a14v35n09/14350913.html>. Acesso em: Nov, 2018.

BRITO, Samuel H. B. **Serviços de Redes em Servidores Linux.** São Paulo: Novatec, 2017.

DOMINGUES, Carlos Eduardo: **O que é o Active Directory?** <https://www.devmedia.com.br/introducao-ao-active-directory-parte-1/21149> Acesso em: Out, 2018.

LENDECKE, Olker: **Advances in Samba 4.** <https://www.samba.org/samba/news/articles/samba4_vl.pdf> Acesso em: Jun, 2018.

MENDES, Douglas Rocha. **Redes de Computadores:** Teoria e Prática. São Paulo: Novatec, 2007.

METZMACHER, Stefan. **Active Directory Replication.** <http://www.samba.org/~metze/presentations/2007/thesis/StefanMetzmacher_Bachelorthesis_ENG_Draft-9811557.pdf> Acesso em: Nov, 2018.

MORIMOTO, Carlos Eduardo. **Servidores Linux: Guia prático.** 2. ed. Porto Alegre: Sul Editores, 2013.

OLIVEIRA, Paulo (2016) "**Linux Solutions: Porque o servidor Samba para Linux .**" <https://linuxsolutions.com.br/por-que-utilizar-o-samba-como-servidor-de-dominio-da-minha-rede/> Acesso em: Out, 2018.

RUSSO, Bruno T. (2007): <https://brunorusso.com.br/utilizando-vbs-com-o-active-directory/> Acesso em: Out, 2018.

SAMBA (2011) **"Samba: An Introduction".** <http://www.samba.org/samba/docs/SambaIntro.html> Acesso em: Out, 2018.

SAMBA (2018) **"Samba: What is Samba"** <https://www.samba.org/samba/what_is_samba.html> Acesso em: Out, 2018.

SHARPE, Richard **Just what is SMB?** Disponível em: <https://www.samba.org/cifs/docs/what-is-smb.html> Acesso em: Ago. 2018

STANEK, William R. **Windows Server 2008: guia completo.** Porto Alegre: Bookman, 2009.

APÊNDICE A - Conteúdo mapear.vbs

```
Set objNetwork = WScript.CreateObject("Wscript.Network")
Set objShell = CreateObject("Shell.Application")
strComputerName = objNetwork.ComputerName
strDomain = objNetwork.UserDomain
strUser = objNetwork.UserName
strAdsPath = strDomain & "/" & strUser

Function IsMember(sGroup)
Dim oDict, oUser, oGroup
If IsEmpty(oDict) Then
        Set oDict = CreateObject("Scripting.Dictionary")
        oDict.CompareMode = vbTextCompare
          Set oUser = GetObject("WinNT://" & strAdsPath &
        ",user")
  For Each oGroup In oUser.Groups
    oDict.Add oGroup.Name, "-"
      Next
     Set oUser = Nothing
   End If
     IsMember = CBool(oDict.Exists(sGroup))
End Function
'SE FOR MENBRO DESSE GRUPO MAPEI
if isMember("redes1fase") then
   objNetwork.MapNetworkDrive "N:" ,
"\\10.0.254.103\Redes1Fase"

'OCULTA O IP DO SERVER E, VOCÊ PODE DAR O
NOME DE EXIBIÇÃO, INDEPENDENTE DO NOME REAL
DA PASTA

objShell.NameSpace("N:").Self.Name = "Pasta_Redes1Fase"
end if
```

```
if isMember("redes2fase") then
objNetwork.MapNetworkDrive "O:" ,
"\\10.0.254.103\Redes2Fase"

'OCULTA O IP DO SERVER E, VOCÊ PODE DAR O NOME DE EXIBIÇÃO, INDEPENDENTE DO NOME REAL DA PASTA

objShell.NameSpace("O:").Self.Name = "Pasta_Redes2Fase"
end if
if isMember("redes3fase") then
objNetwork.MapNetworkDrive "P:" ,
"\\10.0.254.103\Redes3Fase"

'OCULTA O IP DO SERVER E, VOCÊ PODE DAR O NOME DE EXIBIÇÃO, INDEPENDENTE DO NOME REAL DA PASTA

objShell.NameSpace("P:").Self.Name = "Pasta_Redes3Fase"
end if
if isMember("redes4fase") then
objNetwork.MapNetworkDrive "Q:" ,
"\\10.0.254.103\Redes4Fase"

'OCULTA O IP DO SERVER E, VOCÊ PODE DAR O NOME DE EXIBIÇÃO, INDEPENDENTE DO NOME REAL DA PASTA

objShell.NameSpace("Q:").Self.Name = "Pasta_Redes4Fase"
end if
if isMember("redes5fase") then
objNetwork.MapNetworkDrive "R:" ,
"\\10.0.254.103\Redes5Fase"
```

'OCULTA O IP DO SERVER E, VOCÊ PODE DAR O NOME DE EXIBIÇÃO, INDEPENDENTE DO NOME REAL DA PASTA

objShell.NameSpace("R:").Self.Name = "Pasta_Redes5Fase"
end if
if isMember("redes6fase") then
objNetwork.MapNetworkDrive "S:" ,
"\\10.0.254.103\Redes6Fase"

'OCULTA O IP DO SERVER E, VOCÊ PODE DAR O NOME DE EXIBIÇÃO, INDEPENDENTE DO NOME REAL DA PASTA

objShell.NameSpace("S:").Self.Name = "Pasta_Redes6Fase"
end if

APÊNDICE B - Conteúdo shares.xml

<!--
Description: Configuration file for automatic mapping of file shares.
Copyright (C) 2012-2018 Eduardo Moraes <emoraes25@gmail.com>

This file is part of CID (Closed In Directory).
CID is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
CID is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
You should have received a copy of the GNU General Public License
along with CID. If not, see <http://www.gnu.org/licenses/>.
-->

<!--
To define a mapping, add a "volume" tag with its respective parameters
as attributes between the start and end tags of this configuration file.

The main parameters for mounting a CIFS/SMB share are:
fstype [Network File System]
server [Hostname, FQDN or IP of the File Server]
path [Share Name]
mountpoint [Mount Point]
options [Mounting Options - See: "man mount.cifs"]

The following attributes control whether the volume is going to get mounted
once the user logs in. By default, volumes apply to all users:
user [Username]
uid [UID number or UID range]
gid [GID number or GID range]
pgrp [Primary Groupname]
sgrp [Secundary Groupname]

The following variables are provided by the program and can be useful within
attributes and parameters:
%(USER) [Username of the user logging in]
%(DOMAIN_NAME) [Domain name of the user logging in]
%(USERUID) [UID of the user logging in]

%(USERGID) [GID of the primary group of the user logging in]
%(GROUP) [Groupname for %(USERGID)]

EXAMPLES:
Mapping a Public share (full access to everyone):
<volume
fstype="cifs"
server="fileserver.example.com"
path="PUBLIC"
mountpoint="~/PUBLIC"
options="iocharset=utf8,file_mode=0777,dir_mode=0777,dom
ain=%(DOMAIN_NAME)"
/>

Mapping the User Folder ("homes" share created by CID on Samba):
<volume
fstype="cifs"
server="fileserver.example.com"
path="%(USER)"
mountpoint="~/MyNetFolder"
options="iocharset=utf8,file_mode=0700,dir_mode=0700,dom
ain=%(DOMAIN_NAME)"
/>

Conditional mapping to a specific group:
<volume
sgrp="ITD"
fstype="cifs"
server="fileserver.example.com"
path="ITD$"
mountpoint="~/ITD"
options="gid=ITD,iocharset=utf8,file_mode=0770,dir_mode=0
770,domain=%(DOMAIN_NAME)"
/>

Conditional mapping to a specific group of other "trusted domain":
<volume
sgrp="SAMPLE\ITD"
fstype="cifs"
server="fileserver.example.com"
path="ITD$"
mountpoint="~/ITD"
options="gid="ITD",iocharset=utf8,file_mode=0770,dir_mode=0770,domain=%(DOMAIN_NAME)"
/>

For more information, go to:
http://pam-mount.sourceforge.net/pam_mount.conf.5.html
-->
<pam_mount>
<!-- Application control tags (RECOMMENDED DO NOT MAKE CHANGES) -->
<debug enable="0" />
<mkmountpoint enable="1" remove="true" />
logout wait="0" hup="yes" term="yes" kill="yes" />
<!-- DECLARE HERE YOUR MAPPINGS ("<volume... />" tags)! -->
<volume user="*" fstype="cifs" server="10.0.254.103" path="home/%(USER)"
mountpoint="/home/%(USER)/Doc_Remoto"
options="iocharset=utf8,file_mode=0700,dir_mode=0700" />
<volume icase="no" sgrp="redes6fase" fstype="cifs" server="10.0.254.103" path="Redes6Fase"
mountpoint="/home/%(USER)/Redes6Fase"
options="gid=Redes6Fase,iocharset=utf8,file_mode=0770,dir_mode=0770,domain=IFCSOMBRIO" />
<volume icase="no" sgrp="redes5fase" fstype="cifs" server="10.0.254.103" path="Redes5Fase"
mountpoint="/home/%(USER)/Redes5Fase"

options="gid=Redes5Fase,iocharset=utf8,file_mode=0770,dir_mode=0770,domain=IFCSOMBRIO" />
<volume icase="no" sgrp="redes4fase" fstype="cifs" server="10.0.254.103" path="Redes4Fase" mountpoint="/home/%(USER)/Redes4Fase" options="gid=Redes4Fase,iocharset=utf8,file_mode=0770,dir_mode=0770,domain=IFCSOMBRIO" />
<volume icase="no" sgrp="redes3fase" fstype="cifs" server="10.0.254.103" path="Redes3Fase" mountpoint="/home/%(USER)/Redes3Fase" options="gid=Redes3Fase,iocharset=utf8,file_mode=0770,dir_mode=0770,domain=IFCSOMBRIO" />
<volume icase="no" sgrp="redes2fase" fstype="cifs" server="10.0.254.103" path="Redes2Fase" mountpoint="/home/%(USER)/Redes2Fase" options="gid=Redes2Fase,iocharset=utf8,file_mode=0770,dir_mode=0770,domain=IFCSOMBRIO" />
<volume icase="no" sgrp="redes1fase" fstype="cifs" server="10.0.254.103" path="Redes1Fase" mountpoint="/home/%(USER)/Redes1Fase" options="gid=Redes1Fase,iocharset=utf8,file_mode=0770,dir_mode=0770,domain=IFCSOMBRIO" />
<mntoptions allow="nosuid,nodev,loop,encryption,fsck,nonempty,allow_root,allow_other" />
<mntoption require="nosuid,nodev" />
<path>/sbin:/bin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin</path>
</pam_mount>

APÊNDICE C - Conteúdo smb.conf

Global parameters

```
[global]
workgroup = ifcsombrio
realm = ifcsombrio.red
netbios name = srvsamba
interfaces = lo eno2
bind interfaces only = Yes
server role = active directory domain controller
idmap_ldb:use rfc2307 = yes
dns forwarder = 8.8.8.8
server services = s3fs rpc nbt wrepl ldap cldap kdc drepl
winbind ntp_signd kcc dnsupdate dns
map acl inherit = yes
store dos attributes = yes
vfs objects = acl_xattr, full_audit, recycle, acl_tdb
sync always = yes
strict sync = yes

#Auditoria de Arquivos
full_audit:success = open opendir write rename mkdir rmdir
chmod chown unlink
full_audit:prefix = %U|%I|%S
full_audit:failure = none
full_audit:facility = local5
full_audit:priority = notice

#Lixeira
recycle:keeptree = yes
recycle:versions = yes
recycle:repository = /data/trash
recycle:exclude = *.~*, ~*.*, *.bak, *.old, *.iso, *.tmp
recycle:exclude_dir = temp, cache, tmp
[netlogon]
path = /opt/samba/var/locks/sysvol/ifcsombrio.red/scripts
read only = No

[sysvol]
```

```
path = /opt/samba/var/locks/sysvol
read only = No

#### Compartilhamento Geral #####
[dados]
path = /dados
read only = No
browseable = yes
create mask=0766
vfs objects = full_audit, recycle

### Lixeira ###
[lixeira]
path = /data/trash
writeable = yes
read only = no
browseable = no
vfs objects = scannedonly

### Compatilhamento das Pastas Pessoais ####
[home]
path = /data/home
create mask=0766
read only = no

### Compartilhamento por Grupos #####
[Redes1Fase]
path = /data/Redes1Fase
read only = no
create mask=0766
vfs objects = full_audit, recycle

[Redes2Fase]
path = /data/Redes2Fase
read only = no
create mask=0766
vfs objects = full_audit, recycle
```

```
[Redes3Fase]
path = /data/Redes3Fase
read only = no
create mask=0766
vfs objects = full_audit, recycle

[Redes4Fase]
path = /data/Redes4Fase
read only = no
create mask=0766
vfs objects = full_audit, recycle

[Redes5Fase]
path = /data/Redes5Fase
read only = no
create mask=0766
vfs objects = full_audit, recycle

[Redes6Fase]
path = /data/Redes6Fase
read only = no
create mask=0766
vfs objects = full_audit, recycle
```

Monitoramento de Redes para a Câmara Municipal de Mampituba Utilizando o Software Zabbix

Mariane Bertoti Cordova[1], Victor Martins de Sousa[2]

[1]Acadêmica do Instituto Federal Catarinense – Campus Avançado Sombrio – 88960000 – Sombrio – SC – Brasil

[2]Docente do Instituto Federal Catarinense – Campus Avançado Sombrio – 88960000 – Sombrio – SC – Brasil

marianebertoti@hotmail.com,
victor.sousa@ifc.edu.br

Abstract: *Network management is fundamental to dectect or prevent potential failures in a computer network. As objective, the article presents the implementation of network management for the Municipality of Mampituba, with the use of Zabbix software. With the Zabbix agent, it was possible to monitor the computers with Windows operating system, the Linux server itself and for the other devices the SNMP (Simple Network Management Protocol) was used. Zabbix is an Open Source management tool that monitors and detects possible network device failures. The use of the Internet was monitored as well as the availability of the institutional website. In addition, it is possible to know how is the normal behavior of the network, to identify and correct problems.*

Resumo: *A gerência de redes é fundamental para detectar ou prevenir possíveis falhas em uma rede de computadores. Como objetivo, o artigo apresenta a implementação do gerenciamento de redes para a Câmara Municipal de Mampituba, com o uso do software Zabbix. Com a utilização do agente Zabbix foi possível monitorar os computadores com sistema operacional Windows, o próprio servidor Linux e para os demais dispositivos, utilizou-se o protocolo SNMP (Simple Network Management Protocol). O Zabbix é uma ferramenta de gerenciamento Open Source, que monitora e detecta possíveis falhas nos dispositivos de rede. Monitorou-se a utilização da internet bem como a disponibilidade do site institucional. Além disso, pode-se conhecer como é o comportamento normal da rede, identificar e corrigir problemas.*

1 Introdução

O desempenho adequado das redes de computadores é essencial para todos os tipos de empresas. A partir do momento que uma rede cresce, a possibilidade de surgir problemas também é maior. Com isso, torna-se necessário a utilização de ferramentas que auxiliem na administração das redes, uma vez que somente o esforço humano não é suficiente (STALLINGS, 2005).

Nesse contexto, a gerência de redes vem como uma ótima alternativa para a otimização do tempo e eficiência nos procedimentos, como por exemplo, a detecção de uma falha em um dispositivo, com o envio de alerta a um e-mail previamente configurado, avisando o administrador de redes do ocorrido.

Benício (2015) destaca que ter um controle e gerenciamento de um sistema permite detectar anormalidades, faz com que o administrador de redes possa tomar decisões mais rápidas, ao identificar um problema ou ainda agir preventivamente. Complementa que o administrador de redes deve ter conhecimento suficiente e dispor de ferramentas que o auxiliem a identificar falhas, com objetivo de manter a rede e os serviços nela existentes em funcionamento.

Black (2008) ressalta que por menor e mais simples que seja uma rede, essa precisa ser gerenciada, com o objetivo principal de garantir aos usuários a disponibilidade dos serviços de forma satisfatória. Salienta, que uma ferramenta de gerenciamento não resolverá todos os problemas da rede, isso se deve ao fato de que, na maioria das vezes, o administrador da rede não aproveita todas as funcionalidades que o software tem para oferecer.

O ambiente selecionado para tal implementação é a Câmara Municipal de Mampituba e o fator que justifica este trabalho é a inexistência de gerenciamento de redes nesse local. Com isso, cria-se uma dependência da manutenção corretiva, onde é necessário existir um problema para ser aplicado a correção.

Outro fator que justifica a elaboração deste artigo é demonstrar a importância do gerenciamento de redes, independentemente do tamanho da rede escolhida. A ferramenta escolhida é o Zabbix, tendo em vista que essa tem a possibilidade de ser utilizada em pequenas ou grandes redes, além do mais, possui versão gratuita.

2 Referencial Teórico

Dentre os conceitos revistos nesta pesquisa, destacam-se a Gerência de Redes (seção 2.1), protocolo SNMP (seção 2.2) e Zabbix (seção 2.3).

2.1 Gerência de Redes

Para Kurose e Ross (2010), o gerenciamento de redes inclui monitorar, gerenciar e controlar um sistema com inúmeros elementos. Nesse procedimento, o responsável analisa os dados e, se necessário, corrige as irregularidades.

Dentre os objetivos da gerência de redes, Déo (2012) descreve alguns: garantir a disponibilidade e eficiência; monitorar e manter o funcionamento da rede; coletar informações a respeito do funcionamento dos dispositivos; gerar informações sobre a qualidade dos equipamentos; revisar o projeto e a arquitetura da rede; informar sobre possíveis falhas, justificar o investimento em ativos e em um link adequado a demanda da rede.

A arquitetura de gerência de redes é composta pelo Gerente, Agente, Base de Informações de Gerenciamento e Protocolo, ilustrado na Figura 1. Stallings (2005) descreve cada elemento da arquitetura:

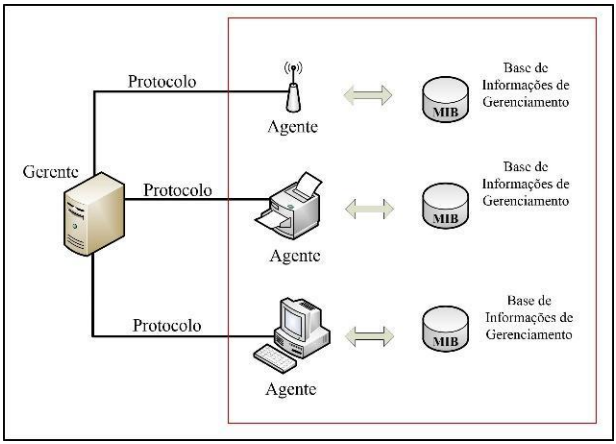
- **Gerente:** pode ser um único dispositivo, sendo utilizado na gerência de redes centralizada ou ainda um sistema compartilhado, onde o ambiente contará com mais de um dispositivo gerenciador. Este último, o autor salienta que é indicado em redes maiores, com alta taxa de tráfego de informações dos Agentes ao Gerente.

- **Agente:** fazem parte deste elemento os computadores, impressoras, roteadores e outros dispositivos que podem ser gerenciados por um Gerente (estação de gerenciamento).

- **Base de Informações de Gerenciamento (MIB):** é uma coleção de objetos que representa as informações de um dispositivo Agente.

- **Protocolo:** a estação de gerenciamento e o agente estabelecem comunicação por meio de um protocolo. O autor destaca que para gerenciamento de redes TCP/IP, o protocolo utilizado é o SNMP, complementa que as versões existentes são SNMPv1, SNMPv2 e SNMPv3.

Figura 1 – Arquitetura de Gerência de Redes



Fonte: Os autores, 2018.

Os padrões de gerenciamento de redes foram iniciados em 1980 pela IETF (*Internet Engineering Task Force*) e ISO (*International Organization for Standardization*), que

desenvolveram os protocolos SNMP e CMIP (*Common Management Information Protocol*) (SOUSA, 2009).

Baseando-se no desenvolvimento da ISO (*International Organization for Standardization*), Kurose e Ross (2010) descrevem um modelo com cinco áreas de gerenciamento de redes, as quais são mencionadas a seguir:

- **Gerenciamento de falhas:** registra, detecta e reage às condições de uma falha na rede;
- **Gerenciamento de configuração:** permite ao administrador ter conhecimento dos dispositivos da rede gerenciada e as suas configurações;
- **Gerenciamento de contabilização:** permite ao administrador registrar e controlar o acesso dos usuários e dos dispositivos que a rede tem para oferecer;
- **Gerenciamento de desempenho:** tem por objetivo quantificar, medir, informar, analisar e controlar o desempenho de diferentes itens da rede;
- **Gerenciamento de segurança:** controla o acesso aos recursos, baseando-se em uma política de segurança.

Scapin (2015) complementa que o nome desse modelo é dado a partir das letras iniciais das áreas e é conhecido como FCAPS, sendo: *Faults* (Gerência de Falhas), *Configuration* (Gerência de Configuração), *Accounting* (Gerência de Contabilidade), *Performance* (Gerência de Desempenho) e *Security* (Gerência de Segurança).

2.2 Protocolo SNMP

Stallings (2005, p. 416) explica que “o SNMP é um protocolo em nível de aplicação que faz parte da família de protocolos TCP/IP. Ele se destina-se a operar sobre o UDP (*User Datagram Protocol*)”.

Case et al. (1990) descrevem na RFC 1157 que a arquitetura SNMP é uma coleção de estações de gerenciamento, que monitoram e controlam elementos de rede. Para isso, os dispositivos gerenciadores (gerentes SNMP) solicitam informações aos dispositivos monitorados (agentes SNMP).

Ainda, Case et al. (1990), expõem alguns objetivos da arquitetura SNMP, que é reduzir o custo de desenvolvimento de software para administração do Agente SNMP, suporte a administração remota e facilidade de utilização por desenvolvedores de ferramentas de gerenciamento de redes.

As versões do protocolo SNMP são SNMPv1, SNMPv2 e SNMPv3. A versão 1 do protocolo é baseada no conceito de “*community strings*”, possui problemas em relação à segurança, uma vez que *strings* e senhas ficam em texto aberto. A segunda versão obteve maior detalhamento de erros, melhorias na eficiência e desempenho, porém, as questões referentes à segurança não foram totalmente corrigidas. O SNMP versão 3 dispõe de autenticação, privacidade e controle de acesso, essa versão buscou melhorar falhas de segurança, presentes nas duas versões anteriores (SCAPIN, 2015).

2.3 Zabbix

O Zabbix é uma ferramenta de monitoramento de redes, capaz de monitorar o desempenho e a disponibilidade dos

equipamentos de rede. A ferramenta realiza a coleta das informações dos dispositivos por meio do protocolo SNMP ou Scripts e armazena no Banco de Dados, informação essa que pode ser consultada e analisada posteriormente (GALIANO FILHO, 2010).

O Zabbix foi desenvolvido na linguagem PHP, por Alexei Vladishev. A primeira versão estável foi lançada no ano de 2001, atualmente, é mantida pela Zabbix SIA. O Zabbix é um software livre, que adota a licença pública geral (GPLv2) (ZABBIX, 2018).

Com base na documentação do Zabbix (2018) e Galiano Filho (2010), estes indicam algumas funcionalidades:

- Monitoramento do desempenho e a disponibilidade de redes, aplicativos e recursos na nuvem;
- Suporte a ambientes pequenos ou grandes;
- Suporte a diferentes arquiteturas;
- Envio de notificações ou execução de comandos à medida que surgem problemas;
- Recursos avançados de visualização, gráficos e mapas de rede personalizáveis;
- Fornecimento de opção de monitoramento distribuído com a utilização do Zabbix Proxy;
- Gerenciamento centralizado;
- Monitoramento em tempo real.

A arquitetura do Zabbix está organizada no modelo *three-tier*, baseada em três camadas: aplicação, banco de dados e interface Web. A camada de aplicação é responsável pela coleta dos dados, na camada de banco de dados é realizado o armazenamento das informações coletadas e na camada de interface Web é possível acessar as informações de monitoramento (LIMA, 2014).

A ferramenta Zabbix é composta por quatro componentes, que são: Zabbix Server, Zabbix Agent, Zabbix Proxy e Interface Web. Dentre esses, a utilização do Zabbix Proxy fica opcional ao administrador da rede (SCAPIN, 2015).

Em conformidade com a documentação do Zabbix (2018), o servidor é o elemento central, que realiza monitoramento, interage com os Agentes e o Proxy. Já o Agente é o componente instalado nos dispositivos monitorados, cujo objetivo é ter conhecimento de seus recursos e aplicações. O Zabbix Proxy realiza a captação dos dados de forma a distribuir o processamento, fazendo com que a sobrecarga no servidor central não seja excessiva.

Para Mota (2017), o Zabbix pode funcionar de duas formas, através de dois tipos de verificações: a verificação passiva e a verificação ativa. A verificação passiva é uma requisição simples de dados, o servidor Zabbix solicita uma informação e o agente Zabbix retorna com o resultado. Na verificação ativa, o agente Zabbix que solicita ao servidor qual informação ele necessita, o servidor envia as informações que deseja solicitar, após isso, o agente armazena essa lista de informações e responde periodicamente ao servidor.

3 Trabalhos Relacionados

A presente seção aborda os trabalhos relacionados, com objetivo de trazer embasamento teórico e prático, a fim de justificar o tema escolhido.

No trabalho de Borges et al. (2015), foi realizado um estudo comparativo entre as ferramentas Nagios e Zabbix, duas das ferramentas de gerência de redes mais utilizadas, segundo a revista Linux Journal em 2014. Utilizando como metodologia a pesquisa bibliográfica e a pesquisa experimental. A implementação foi realizada em um laboratório do Instituto Federal Catarinense. Teve como principais testes o consumo de rede, tempo de resposta e os custos computacionais. Como resultado da pesquisa, os autores apontam a ferramenta Zabbix como mais vantajosa e com mais funcionalidades.

Scapin (2015) descreveu em seu trabalho uma análise de três ferramentas de gerência de redes: MRTG (*Multi Router Traffic Grapher*), Nagios e Zabbix, com ênfase em questões de usabilidade e funcionalidades da interface Web. Comparou e destacou pontos importantes e a forma de gerar gráficos. Foi utilizado uma rede virtualizada, com o software Oracle VM VirtualBox e como resultados, ficou constatado que existem muitas semelhanças entre as três ferramentas. Todavia, Nagios e Zabbix mostraram-se mais completas, sendo que o Zabbix é mais fácil de utilizar em comparação com o Nagios, uma vez que este último precisa de plug-ins para complementar as funcionalidades dos usuários.

Almeida e Rohden (2017) fizeram um estudo do protocolo SNMP e da ferramenta Zabbix, com o objetivo de monitorar equipamentos utilizados em internet via rádio. Foi

utilizada a metodologia de pesquisa aplicada, a qual tem como motivação a necessidade de produzir conhecimento a partir de resultados e contribuir para fins práticos. Utilizou-se um ambiente virtualizado com o software VirtualBox e instalado o Zabbix *Appliance*, este apresenta a configuração previamente instalada.

Silva, Medeiros e Martins (2015) abordaram uma análise de rede, utilizando a ferramenta Zabbix em conjunto com o protocolo SNMP (versões 1 e 2), a fim de monitorar dispositivos e analisar em critérios estatísticos, dados como disponibilidade e quantidade de tráfego dos links, de forma proativa e centralizada. Para a realização dos testes, utilizou-se, em um primeiro momento, um ambiente virtualizado com o auxílio da ferramenta de emulação GNS3 (*Graphical Network Simulator 3*), posteriormente, foi utilizado em ambiente real. Observando o comportamento da rede, verificou-se que a ferramenta resolveu problemas antes mesmo que acontecessem, sem intervenção direta do administrador de redes.

Nesta seção foram apresentados quatro trabalhos que tratam da mesma área de estudo. Pode-se notar que os trabalhos evidenciam a importância da gerência de redes, sendo que nos dois primeiros foi realizado um comparativo de ferramentas e nos dois últimos foi feito um estudo do software Zabbix.

No trabalho de Borges et al. (2015) houve uma vantagem estabelecida pelo Zabbix, no quesito desempenho da ferramenta. Ainda, o Zabbix não depende de plugins/complementos desenvolvidos por terceiros e segundo os autores, dificultou a utilização da ferramenta. Baseando-se nos resultados obtidos e nas considerações dos autores, escolheu-se a ferramenta Zabbix para realizar a implementação.

4 Materiais e Métodos

Quanto a natureza da pesquisa, essa é caracterizada por Marconi e Lakatos (2012) como pesquisa aplicada, tendo em vista que os resultados da implementação são conhecidos e utilizados para solucionar problemas existentes. De acordo com os objetivos, pode-se classificar como pesquisa exploratória. Buscou-se informações sobre o objeto de estudo, gerando conhecimento sobre o tema abordado (SEVERINO, 2007).

Quanto aos procedimentos, a metodologia que foi utilizada neste artigo é a pesquisa bibliográfica, realizada em livros, sites, base de dados de trabalhos científicos, bem como a pesquisa experimental, que segundo Gil (2010), não precisa ser realizada em laboratório, desde que apresente três atributos, sendo estes: manipulação, controle e distribuição aleatória. A manipulação necessita que algo seja feito para manipular alguma característica do que está sendo estudado, o controle diz respeito ao que e como o pesquisador irá fazer para controlar tal experimento e a distribuição aleatória está relacionada com a coleta dos dados de forma contingente, sem que exista intervenção do pesquisador no resultado da pesquisa.

4.1 Câmara Municipal de Mampituba

Para realizar a implantação foi utilizado o ambiente da Câmara Municipal de Mampituba, localizada na Avenida Herculano Lopes, 230, Centro – Mampituba/RS. A descrição e análise dos dispositivos existentes foram possíveis mediante autorização legal portada aos autores[22]. Constatou-se a existência dos

[22] Autorização fornecida pelo Presidente Sérgio Barbosa Martins.

seguintes dispositivos: sete computadores com sistema operacional Windows (sendo dois com Windows 8.1 e cinco com Windows 7); uma impressora IP da marca BROTHER - MFC 8712DW; um *Access Point* da marca INTELBRAS; um HotsPot 300 da marca INTELBRAS; uma routerboard RB 750 com o sistema operacional RouterOS do fabricante Mikrotik e um Switch TP-LINK não-gerenciável.

No cotidiano do ambiente pesquisado, observou-se constantes reclamações referentes à qualidade da internet, contudo, não era possível verificar se o link de internet fornecido pelo provedor estava de acordo com a velocidade contratada, ou ainda se algum dispositivo estava com tráfego excessivo de dados.

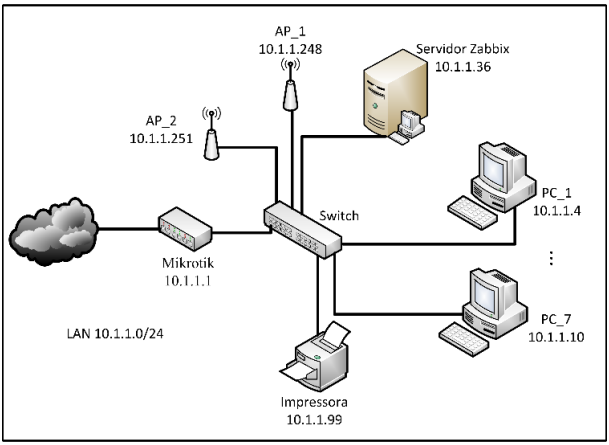
4.2 Ferramentas utilizadas

Para o processo de instalação e implementação do servidor, foi utilizado um ambiente virtualizado, com o software VirtualBox, versão 5.2.20, instalado em um computador da marca Centrium, processador Intel Core i7-7700 de 3.60Ghz, memória RAM (*Random Access Memory*) de 8GB e HD de 1TB, com sistema operacional Windows 8.1 Pro de 64 bits. Salienta-se que não havia computador dedicado para tal implementação, sendo assim, utilizou-se a virtualização do servidor.

Utilizou-se o sistema operacional Ubuntu Desktop 18.04.1 LTS de 64 bits virtualizado, contendo 2GB de memória RAM e 40GB de armazenamento. A versão do Zabbix foi a 3.4.13, uma vez que no início deste estudo era a versão estável em utilização, sendo que o Zabbix 4.0 foi lançado em 1º de outubro de 2018, período esse que a implantação encontrava-se em andamento.

Usou-se o MariaDB, versão 10.1.34, como banco de dados para armazenar as informações de gerenciamento, servidor Web Apache versão 2.4.29 e PHP versão 7.2.10. A topologia lógica está apresentada na Figura 2, inserida abaixo.

Figura 2 - Topologia Lógica.



Fonte: Os autores, 2018.

Conforme ilustrado na imagem, usou-se a representação de PC_1 e PC_7, como sendo o primeiro e o último computador da topologia lógica, com objetivo de reduzir o tamanho da Figura.

5 Implementação

Salienta-se que a instalação do servidor Zabbix está de acordo com a documentação oficial, sendo que não houve escolha entre as versões (informadas na seção 5.2) do banco de dados (MariaDB), servidor Web (Apache) e o PHP, uma vez que o download da ferramenta foi realizado mediante repositório

oficial do Zabbix e integra esses serviços e as respectivas versões.

Primeiramente, é necessário instalar o repositório com Banco de Dados, fazer o download do servidor Zabbix, frontend[23] e do agente. Por padrão, cria-se o banco de dados com o nome de "zabbix" e atribui-se privilégios para que os dados coletados tenham permissão de escrita e, posteriormente, o arquivo "*/etc/zabbix/zabbix_server.conf"* deve ser editado, inserindo a senha do banco de dados. Vale ressaltar que este último é o arquivo de configuração principal do servidor Zabbix, no entanto, as informações inseridas são apenas as citadas anteriormente.

A documentação oficial do Zabbix informa que é necessária a alteração de *timezone* (fuso horário) no arquivo "*/etc/zabbix/apache.conf"* com a seguinte informação "*php_value date-timezone America/Sao_Paulo*". Todavia, somente essa modificação não funcionará. Deve-se alterar o arquivo "*/etc/php/7.2/apache2/php.ini*", o fuso horário, ficando da seguinte forma, "*date.timezone = America/Sao_Paulo*".

O próximo passo é iniciar os processos do servidor e do agente Zabbix, com os comandos: "*systemctl restart zabbix-server zabbix-agent apache2*" para reiniciar os serviços (Zabbix server, Zabbix agente e servidor web) e "*systemctl enable zabbix-server zabbix-agent apache2*" para o Zabbix inicializar junto ao sistema. No navegador, utiliza-se "*http://10.1.1.36/zabbix*" para finalizar as configurações do lado servidor.

---

[23] Interface de interação com o usuário.

Em resumo, as configurações no navegador são apenas para confirmar as que já foram inseridas no servidor Zabbix. Ressalta-se que os requisitos devem ser atendidos, caso contrário não será possível concluir a instalação. Caso exista alguma irregularidade, recomenda-se verificar os passos anteriores, bem como os arquivos de configuração.

5.1 Configuração nos Agentes

Nesta seção será explicado brevemente como foi realizada a configuração dos dispositivos monitorados, cujo objetivo não é explicar um passo a passo, mas sim destacar as configurações mais relevantes. Enfatiza-se que a RB 750 da Mikrotik, a impressora IP e o Hotspot 300 foram gerenciados por meio do protocolo SNMP, em ambos foi necessário apenas habilitar o protocolo SNMP, sendo que na impressora IP e no Hotspot foi utilizado a interface Web do próprio dispositivo e para o Mikrotik, utilizou-se o aplicativo Winbox na versão 3.18 para acessar a interface de configuração.

A ferramenta Zabbix possui suporte nativo ao protocolo SNMP, sendo assim, não foi necessário fazer nenhuma configuração adicional no Gerente, apenas foi adicionado cada host através da interface web do Zabbix.

Para os computadores com sistema operacional Windows, disponibiliza-se o arquivo de configuração no site oficial da ferramenta. O arquivo "*zabbix_agentd.win*" deverá estar em uma pasta com o nome de "*Zabbix*" no disco local "*C:*", onde foi incluído o endereço IP do servidor. Em seguida, executou-se o aplicativo "*zabbix_agentd*" para validar o monitoramento do dispositivo.

A configuração do Agente Zabbix, no servidor, foi realizada no arquivo "*/etc/zabbix/zabbix_agentd.conf*", o Quadro 1 exibe as informações que foram incluídas.

Quadro 1: Arquivo zabbix_agentd.conf.

Linha/Definição	Descrição
Server = 10.1.1.36	IP do servidor
StartAgents = 5	Números de instâncias do processo agente que serão iniciadas no host
Hostname = zabbix server	Nome do servidor
Timeout = 3	Valor padrão de 3 segundos, podendo ser até 30

Fonte: Os autores, 2018.

6 Resultados e Discussões

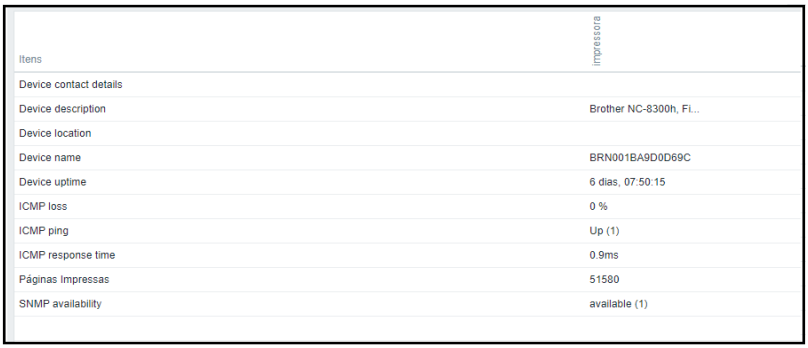
A interface Web do Zabbix possui cinco menus principais, sendo eles: monitoramento, inventário, relatórios, configuração e administração, sendo que cada um apresenta menus secundários, que serão explicados no decorrer da explanação dos resultados, com seção específica para cada um.

6.1 Monitoramento

No item "Dashboard", foi configurado para visualização com as informações dos principais incidentes ocorridos, mapa da rede, status dos hosts e status do Zabbix. Destaca-se que esse item pode ser editado, de acordo com as necessidades do administrador da rede.

É possível ter uma visão geral de cada dispositivo monitorado. A Figura 3 exibe as informações da impressora IP, como descrição e nome do dispositivo, tempo de atividade, porcentagem de perda de ping (0%), disponibilidade (1), tempo de resposta (0.9ms), quantidade de páginas impressas (51580) e se o protocolo SNMP está acessível (1).

Figura 3: Visão Geral da Impressora IP.

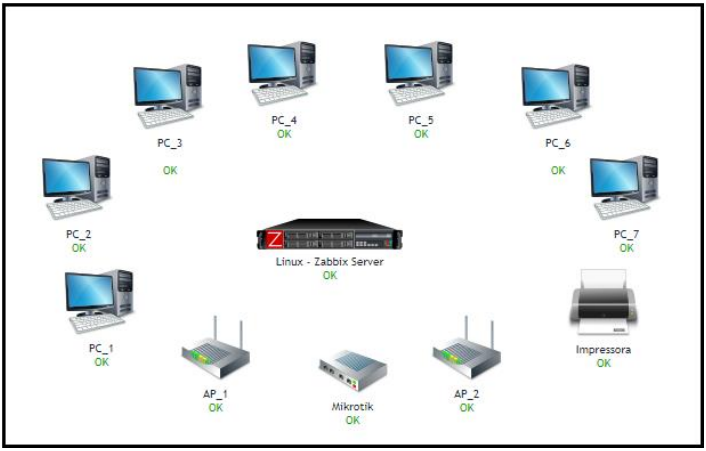


Itens	impressora
Device contact details	
Device description	Brother NC-8300h, Fi...
Device location	
Device name	BRN001BA9D0D69C
Device uptime	6 dias, 07:50:15
ICMP loss	0 %
ICMP ping	Up (1)
ICMP response time	0.9ms
Páginas Impressas	51580
SNMP availability	available (1)

Fonte: Os autores, 2018.

A utilização de um mapa da rede precisa ser feita de forma manual, para se ter uma visão geral de como está o comportamento de toda a rede. Salienta-se que a visualização do mapa é de acordo com os níveis de severidade (Não classificada, Informação, Atenção, Média, Alta e Desastre), esses níveis representam qual é a gravidade de uma trigger. O mapa da rede, apresentado na Figura 4, representa que todos os hosts estão em funcionamento.

Figura 4: Mapa da Rede.



Fonte: Os autores, 2018.

O nível de severidade “Não classificada” representa uma severidade não conhecida. “Informação” apenas informa que um fato ocorreu, um exemplo é quando o servidor Zabbix foi reiniciado. A severidade “Atenção” somente emite um aviso que o administrador da rede poderá ter problemas futuros. “Média” informa que há incidente de grau intermediário. “Alta” quer dizer que algo importante aconteceu e “Desastre” há a perda total de capacidade, sendo que esta última não foi acionada nenhuma vez. O exemplo da Figura 5 mostra três níveis de severidade diferentes.

Visualização de Dados Recentes, Triggers, Gráficos e Descoberta são os demais menus de monitoramento. Cada vez que um dispositivo é monitorado são enviadas informações dos itens gerenciados ao servidor Zabbix, sendo que o item pode ter uma trigger associada a ele. Como exemplos de itens, cita-se o

tráfego de rede na interface de entrada/saída e tempo de resposta do ping.

A trigger é uma regra que vai ser verificada cada vez que houver a coleta de um item. Pode-se definir que será acionada uma trigger quando o armazenamento de um dispositivo atingir 80% de utilização, por exemplo. De acordo com os níveis de severidade, que foram citados nesta seção (7.1), a Figura 5 mostra as triggers que foram configuradas e recebidas, com nível de severidade "Atenção", onde o disco local "C:" encontrava-se menor que 20%, o nível de severidade "Média", onde a interface 2 da Routerboard Mikrotik estava desligada e o nível de severidade "Alta", no qual o PC_4 estava desligado.

Figura 5: Exemplo de Trigger.

Fonte: Os autores, 2018.

Para a utilização do monitoramento por Descoberta, deve-se utilizar o recurso de Autodescoberta. No caso dessa implementação, foi inserido o intervalo de IP's da rede (10.1.1.1 a 10.1.1.254) e a ferramenta buscou automaticamente todos os dispositivos conectados à rede. Esse tipo de monitoramento fez com que se tivesse conhecimento de quantos dispositivos estão conectados em um determinado momento, conforme explanado no final da seção 9 (Monitoramento do Mikrotik).

6.2 Relatórios

Fazem parte do menu "Relatórios", os itens: Status do Zabbix, Relatório de Disponibilidade, Top 100 de Triggers, Auditoria,

Log de Ações e Notificações. O Status do Zabbix informa se o servidor Zabbix está funcionando corretamente, qual a quantidade de hosts, itens e triggers estão habilitados/desabilitados, números de usuários e desempenho requerido do servidor.

No servidor Zabbix foram monitorados a quantidade total de 81 itens, dentre eles está: sistema de arquivos, utilização de memória, interfaces de rede, carga no processador e disponibilidade do dispositivo.

Evidencia-se que os Relatórios de Disponibilidade especificam a porcentagem que um determinado host, item ou serviço ficou disponível. É possível utilizar o recurso de filtros por: grupo de hosts, um host específico e um período de tempo determinado.

O menu Relatórios também reúne as 100 Triggers mais anunciadas. A Trigger que mais gerou alertas foi "*Zabbix discoverer processes more than 75% busy*", obteve um total de 632 vezes. Nessa Trigger, os processos do Servidor Zabbix estavam a mais de 75%, à medida que foram incluídos mais dispositivos, mais informações foram enviadas e aumentou o processamento no servidor, fazendo com que a trigger fosse acionada, gerando esses alertas.

Para corrigir esse problema, tornou-se indispensável realizar alterações no arquivo de configuração "*/etc/zabbix/zabbix_server.conf*", que seguem informadas no Quadro 2.

Quadro 2: Alteração no arquivo /etc/zabbix/zabbix_server.conf.

Linha/Definição	Descrição
StartPollers = 20	Range: 0-1000
StartPingers = 20	Range: 0-1000
StartDiscoverers = 20	Range: 0-250

Fonte: Os autores, 2018.

Para a resolução do problema foi utilizada a orientação do Fórum do Zabbix, o valor utilizado em ambos os casos foi 20. Todavia, o administrador de redes pode alterá-lo de acordo com o ambiente monitorado, a limitação imposta é que o valor esteja no intervalo especificado na coluna da direita.

O item Auditoria informa a data, horário, nome de usuário, endereço IP do dispositivo que realizou a ação, o recurso, a ação, o ID (informado automaticamente pelo Zabbix) e a descrição do que foi realizado, conforme exibição na Figura 6. Pode-se filtrar o relatório de auditoria com um usuário específico, uma ação realizada e o recurso inserido, excluído ou modificado.

Figura 6: Auditoria

Hora	Usuário	IP	Recurso	Ação	ID	Descrição
27-09-2018 15:09:25	mariane	10.1.1.4	Aplicação	Adicionado	1083	Web
27-09-2018 14:18:22	mariane	10.1.1.4	Trigger	Adicionado	15727	High ICMP ping loss
27-09-2018 14:18:22	mariane	10.1.1.4	Trigger	Adicionado	15728	High ICMP ping response time
27-09-2018 14:18:22	mariane	10.1.1.4	Trigger	Adicionado	15726	Unavailable by ICMP ping
27-09-2018 14:14:47	mariane	10.1.1.4	Host	Adicionado	10256	Mikrotik
27-09-2018 14:13:30	mariane	10.1.1.4	Grupo de hosts	Adicionado	15	Windows
27-09-2018 13:31:31	mariane	10.1.1.4	Usuário	Login	0	
26-09-2018 15:39:37	mariane	10.1.1.4	Usuário	Login	0	

Fonte: Os autores, 2018.

O Log de Ações representa a tentativa de notificação gerada pelo Zabbix, informa a data e horário, a ação que foi realizada, o Tipo (E-mail), destinatário, mensagem e Status (podendo ser “enviado” ou “falhou”).

O envio de notificações por e-mail não é de forma instantânea. Para que uma notificação seja enviada, é preciso inserir informações no menu Administração, como: nome e tipo de mídia “*e-mail*”, SMTP (*Simple Mail Transfer Protocol)* server “*smtp.gmail.com*”, SMTP server port “*465*”, SMTP helo “*gmail.com*” e SMTP email “*tcczabbix2018@gmail.com*”.

Além disso, é necessário configurar uma “Trigger”, podendo ser uma trigger específica para um dispositivo, ou ainda uma trigger para um grupo de hosts. A Figura 7 demonstra o recebimento de alguns e-mails de notificações.

Figura 7 – Recebimento de Notificações por E-mail.

Fonte: Os autores, 2018.

6.3 Configuração

Este menu apresenta as configurações que foram utilizadas na implantação: Grupo de Hosts, Templates, Hosts, Ações e Descoberta.

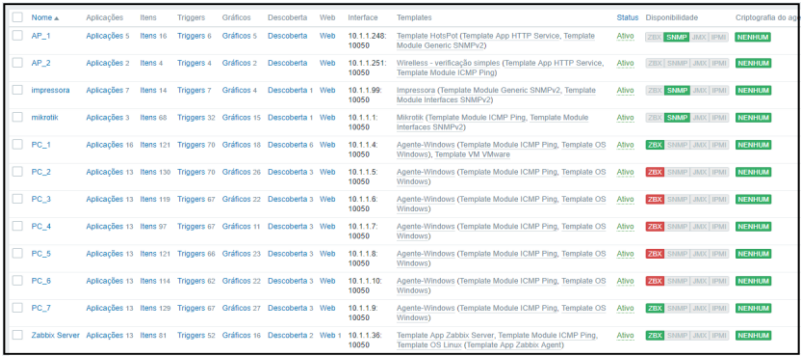
A ferramenta Zabbix disponibiliza Templates prontos para determinados tipos de sistemas operacionais. Como exemplo, pode-se citar alguns dos templates: Zabbix Agente, Zabbix Server, ICMP (*Internet Control Message Protocol*) e Dispositivo SNMP.

Também é possível configurar um grupo de hosts, os quais possuem características semelhantes, como computadores com sistema operacional Windows. A utilização de um grupo de hosts facilita o trabalho do administrador, tendo em vista que é realizada a configuração de apenas um dispositivo e é aproveitada pelos demais.

Para incluir o monitoramento de um dispositivo, deve-se incluir o nome do mesmo, associá-lo a um grupo e informar o endereço IP no campo correspondente ao tipo de monitoramento.

O monitoramento pode ser pela interface do Agente Zabbix (instalado no dispositivo monitorado), por SNMP, IPMI (*Intelligent Platform Management Interface*), JMX (*Java Management Extensions*) ou verificação simples, sendo que neste estudo foi utilizado o gerenciamento por SNMP (AP_1, impressora e Routerboard Mikrotik), Agente Zabbix (PC_1 ao PC_7 e Zabbix Server) e Verificação Simples (AP_2). A Figura 8 representa os hosts cadastrados, visto que na coluna "Disponibilidade" os dispositivos que estão em vermelho, encontram-se desligados.

Figura 8 – Hosts Cadastrados.



Fonte: Os autores, 2018.

6.4 Administração

Por fim, a ferramenta Zabbix possui o menu de Administração, com os seguintes itens: Geral, Proxies, Autenticação, Grupos de Usuários, Usuários, Tipos de Mídias, Scripts e Fila. De forma geral, esse menu permite gerenciar os usuários, controlando a permissão que cada um poderá visualizar, por meio da interface web da ferramenta, sendo que o usuário *Guest* (convidado) foi

desabilitado, uma vez que esse poderia visualizar as informações dos menus Monitoramento, Relatório e Inventário.

Importante destacar sobre a utilização do item “Fila”, este é capaz de controlar a saúde do servidor Zabbix, uma vez que fornece um panorama geral dos itens que ainda precisam ser processados pelo servidor Zabbix.

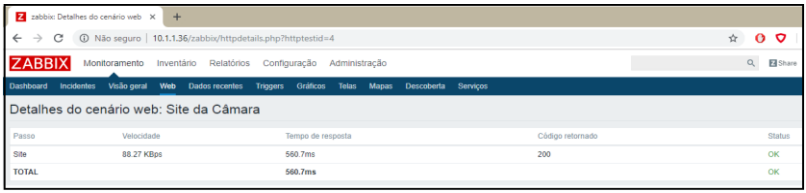
7 Monitoramento Web

O monitoramento Web é uma outra funcionalidade do Zabbix, é possível verificar se uma URL (*Uniform Resource Locator*) está disponível, incluindo a velocidade de download e tempo de resposta.

Usou-se o recurso de configuração de Ações, onde foi adicionada uma Trigger caso existisse a interrupção do site e, diante disso, enviar um alerta ao administrador da rede. No período de utilização deste gerenciamento, compreendido entre 15 de outubro de 2018 e 19 de novembro de 2018, não foi constatado nenhuma interrupção do site. Em alguns momentos, foi apenas constatado um alto tempo de resposta e baixa velocidade de download.

Para o item “Velocidade de Download”, o valor mínimo detectado foi de 3.23Kbps e o valor máximo foi de 214.57Kbps. Já o item “Tempo de resposta”, o valor mínimo detectado foi de 219.2ms e o valor máximo foi 14s 559.6ms. A média de utilização dos recursos do site está representada pela Figura 9.

Figura 9: Cenário Web.



Fonte: Os autores, 2018.

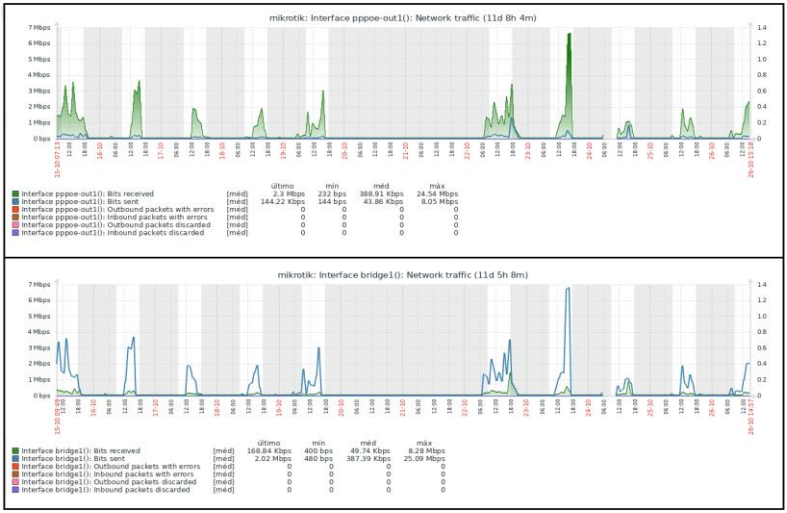
O site da Câmara foi adquirido através do programa Interlegis, disponibilizado pelo Senado Federal. Não há informações de onde o site fica hospedado.

8 Monitoramento do Mikrotik

O acompanhamento do dispositivo Mikrotik foi realizado pelo protocolo SNMP, focando-se principalmente da utilização na internet, sendo que o plano contratado junto ao Provedor de Internet é de 15 Mbps de download e 7 Mbps de upload.

Diante da reclamação dos usuários, informada na seção 5.1, pode-se verificar que eventualmente o link de internet não atingia a taxa contratada. A Figura 10 representa o tráfego de rede na interface de entrada da internet, nomeada como PPPOE e o tráfego na interface de distribuição de internet aos usuários, nomeada como Bridge, no período entre 15 e 26 de outubro de 2018.

Figura 10: Tráfego nas Interfaces PPPOE e Bridge.



Fonte: Os autores, 2018.

Tanto na interface Bridge, quanto na PPPOE, não houve pacotes descartados ou com erros. Ressalta-se que o servidor Zabbix ficou ligado 24 horas por dia, somente em alguns momentos, o mesmo foi desligado para a realização de backup.

A utilização média dos recursos na interface PPPOE e Bridge estão evidenciadas no Quadro 3. Pode-se verificar que houve oscilação do link de internet, diante do valor máximo de dados recebidos/enviados ser superior à média.

Quadro 3: Média do Tráfego de Dados no Mikrotik

Interface Bridge – Distribuição da Internet			
	Mínimo	Médio	Máximo

Bits recebidos	56 bps	82.25 Kbps	20.7 Mbps
Bits enviados	480 bps	601.74 Kbps	25.09 Mbps
Interface PPPOE – Entrada da Internet			
	Mínimo	Médio	Máximo
Bits recebidos	232 bps	586.6 Kbps	25.54 Mbps
Bits enviados	40 bps	71.79 Kbps	20.1 Mbps

Fonte: Os autores, 2018.

Verificou-se que o maior consumo de internet é durante o horário de realização das Sessões Plenárias nas segundas-feiras, no horário compreendido entre 18:00h e 20:00h e também na realização de eventos para terceiros, uma vez que há uma elevação no número de visitantes em ambas as situações.

Utilizou-se o dia 05 e 12 de novembro de 2018 (segundas-feiras) para análise do percentual de utilização de download, sendo que no dia 05, entre 08:00h e 18:00h, o consumo representou 43,38% e das 18:00h às 20:00h o consumo representou 56,62%. No dia 12, o aumento no horário próximo à Sessão Plenária foi maior, entre 08:00h e 18:00h, o consumo representou 37,15% e das 18:00h às 20:00h representou 62,85%.

No dia 21 de novembro de 2018 (quarta-feira) houve a cedência do espaço da Câmara para a realização de uma reunião, entre 13:30h e 18:30h e o consumo foi de 89,42%, já no horário de expediente o percentual foi de 10,58%. Conforme o recurso de autodescoberta do Zabbix, foi observado a quantidade

máxima de 49 dispositivos conectados no horário da reunião (entre 13:30h e 18:30h).

Salienta-se que houve pouca diferença no percentual de upload entre o horário de expediente, de realização da sessão ou cedência do espaço da Câmara para terceiros.

9 Considerações Finais

Após as configurações, coleta e armazenamento dos dados, pode-se verificar na prática como é o funcionamento da ferramenta Zabbix. Com isso, foi possível examinar como é o desempenho normal de cada dispositivo e a partir dessa informação, examinar comportamentos irregulares e tomar as providências necessárias.

Uma vantagem encontrada foi a facilidade de configuração e instalação do servidor Zabbix. Uma das principais dificuldades percebidas foi a de configurar um monitoramento avançado, como por exemplo, os dispositivos monitorados com SNMP, tendo em vista gerenciar itens de forma avançada, como o consumo de rede no Hotspot 300 da INTELBRAS ou nível de toner da impressora. O motivo para este monitoramento não ocorrer foi devido à falta de conteúdo explicativo com a OID (identificador de objeto) para monitoramento de cada situação.

Um recurso bastante útil do Zabbix é a autodescoberta, que permite detectar automaticamente itens de dispositivos a serem monitorados, inclui a associação de triggers a esses itens, facilitando o trabalho do administrador Zabbix. Além disso, registra-se as muitas utilidades da ferramenta Zabbix, sendo que neste trabalho foram apresentadas apenas algumas delas e muito ainda se pode pesquisar.

Para trabalhos futuros, sugere-se explorar mais as funcionalidades do Zabbix, como o monitoramento por IPMI e JMX. Indica-se também o uso da ferramenta em um ambiente com maior fluxo de usuários e aplicações, incluindo o gerenciamento com o Zabbix Proxy.

Além disso, recomenda-se um estudo aprofundado do protocolo SNMP, principalmente para analisar a rede *wireless*, uma vez que não foi possível atender esse indicador, tendo em vista que não foi encontrada a lista com os OID's do dispositivo e a respectiva descrição de cada um identificador de objeto.

No cenário atual, a rede *wireless* e a rede cabeada compartilham a mesma largura de banda. Contudo, não foi possível verificar o monitoramento da rede *wireless* e, por consequência, não dispor desse conhecimento. Diante dessa situação, sugere-se a aquisição de um switch gerenciável, com suporte ao protocolo SNMP e VLAN (*virtual local area network*) e também a divisão da largura de banda em 3 redes, uma para a rede administrativa, a segunda para a rede *wireless* de funcionários e vereadores e a terceira para visitantes.

No entanto, nenhuma das propostas de melhorias sugeridas anteriormente puderam ser implementadas durante a realização deste trabalho, uma vez que depende de orçamento público para aquisição de um novo dispositivo. Ressalta-se que o servidor Zabbix permanecerá em funcionamento, tendo em vista que foi objetivo deste trabalho implantar a ferramenta Zabbix na Câmara Municipal.

Referências

ALMEIDA, Douglas Rodrigues; ROHDEN, Rafael Barasuol. **Utilizando o protocolo SNMP**

para gerenciar ativos de rede no Zabbix. UNICRUZ - Universidade de Cruz Alta, 2017. Disponível em: <http://www.revistaeletronica.unicruz.edu.br/index.php/revistaeletronica/article/view/5452>. Acesso em: 02 set. 2018

BENÍCIO, Washington Ernando Pereira. **Monitoramento e gerenciamento de redes utilizando Zabbix.** Monografia (Tecnologia em análise e desenvolvimento de sistemas) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Capivari/SP, 2015. Disponível em: <http://zabbixbrasil.org/files/Monitoramento_e_Gerenciamento_de_Redes_Utilizando_Zabbix.pdf>. Acesso em: 16 maio 2018.

BLACK, Tomas Lovis. **Comparação de Ferramentas de Gerenciamento de Redes.** Monografia (Curso de Especialização em tecnologias, gerência e segurança de redes de computadores) Universidade Federal do Rio Grande do Sul - Porto Alegres/RS, 2008. Disponível em: <http://www.lume.ufrgs.br/bitstream/handle/10183/15986/000695315.pdf?sequence=1>. Acesso em: 24 maio 2018.

BORGES, Aliguieri Miguel et. al. **Estudo comparativo entre Nagios e Zabbix**. In: FREITAS JUNIOR, Vanderlei et. al. (Org.). Tecnologia e Redes de Computadores: Estudos Aplicados. Sombrio: Instituto Federal Catarinense – Campus Avançado Sombrio, 2015.

CASE, J. et. al.. **Simple Network Management Protocol (SNMP).** Network Working Group, RFC 1157, 1990. Disponível em: <https://www.ietf.org/rfc/rfc1157.txt>. Acesso em: 24 maio 2018.

DÉO, André Luis Boni. **Gerenciamento de Redes com Zabbix.** Universidade de Campinas: São Paulo, 2012. Disponível em: <http://zabbixbrasil.org/files/Apostila_Gerencia_de_Redes_com_SNMP%20v2.zip>. Acesso em: 13 ago. 2018.

GALIANO FILHO, Adilson. **Avaliação da ferramenta Zabbix.** Monografia (Curso de Especialização em Redes e Segurança de Sistemas). Pontifícia Universidade Católica do Paraná - Curitiba/PR, 2010. Disponível em: <https://www.ppgia.pucpr.br/~jamhour/RSS/TCCRSS08B/Adilson%20Galiano%20-%20Artigo.pdf>. Acesso em: 10 ago. 2018.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2010.

KUROSE, James F.; ROSS, Keith W. **Redes de Computadores e a Internet:** uma abordagem top-down. 5. ed. São Paulo: Pearson, 2010.

LIMA, Janssen dos Reis. **Monitoramento de Redes com Zabbix:** monitore a saúde dos servidores e equipamentos de rede. Rio de Janeiro: Brasport, 2014.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Técnicas de Pesquisa.** 7. ed. São Paulo: Atlas, 2012.

MOTA, Levi da Costa. **Uma análise comparativa dos protocolos SNMP, Zabbix e MQTT, no contexto de aplicações de internet das coisas.** Dissertação (Programa de Pós-Graduação em Ciência da Computação). Universidade Federal de Sergipe - São Cristóvão/SE, 2017. Disponível em: <https://repositorio.ifs.edu.br/biblioteca/bitstream/12345678

9/467/1/Disserta%C3%A7%C3%A3o%20Levi%20da%20Costa%20Mota.pdf>. Acesso em: 02 set. 2018.

SCAPIN, Alex Henrique. **Análise de ferramentas de gerência de redes e interfaces Web.** Monografia (Tecnologia em Redes de Computadores) - UFSM/RS, Santa Maria, 2015. Disponível em: <http://www.redes.ufsm.br/docs/tccs/Alex-Scapin.pdf>. Acesso em 09 ago. 2018.

SEVERINO, Antônio Joaquim. **Metodologia do Trabalho Científico.** 23. ed. São Paulo: Cortez, 2007.

SILVA, W. M. C.; MEDEIROS, R. M.; MARTINS, R. S. **Análise e gerenciamento de redes usando uma metodologia proativa com a ferramenta Zabbix.** Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte. Disponível em: <http://www2.ifrn.edu.br/ojs/index.php/HOLOS/article/view/2441/1328>. Acesso em: 03 set. 2018.

SOUSA, Lindeberg Barros de. **Projetos e implementação de redes:** fundamentos, soluções, arquiteturas e planejamento. São Paulo: Érica, 2009.

STALLINGS, William. **Redes e Sistemas de comunicação de dados:** teoria e aplicações corporativas. 5. ed. Rio de Janeiro: Elsevir, 2005.

ZABBIX. **Zabbix Documentation.** 2018. Disponível em: <https://www.zabbix.com/documentation/3.4/pt/manual>. Acesso em: 10 ago. 2018.

OwnCloud + OpenLDAP: Serviço de compartilhamento de dados em nuvem com autenticação centralizada

Yasmim de Matos Nunes[1], Victor Martins de Sousa[2], Sandra Vieira[2]

[1]Acadêmica do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88960-000 – Sombrio – SC – Brasil

[2]Docentes do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88960-000 – Sombrio – SC – Brasil

yasmimmn123@gmail.com, {victor.sousa, sandra.vieira}@ifc.edu.br

Abstract. *This work presents the integration of a cloud data storage service and a centralized authentication service using the OwnCloud and OpenLDAP tools. The article was constituted through research methods: bibliographic, exploratory and qualitative, and practical tests were also adopted to achieve the expected results. For the test environment, 5 virtualized computers were used, among them 3 clients and the other 2 servers, the latter being implementations of the services used in the research. The results allowed us to identify that the integration of the tools was successful, since it enabled the previously registered users in an LDAP server to have the access to manage their folders and files located in the OwnCloud server.*

Keywords. *OwnCloud. LDAP. Integration of LDAP server to the OwnCloud server. Cloud Storage. Centralized Authentication.*

Resumo. *Este trabalho apresenta a integração de um serviço de armazenamento de dados em nuvem e um serviço de autenticação centralizada a partir da utilização das ferramentas OwnCloud e OpenLDAP. O artigo foi constituído através dos métodos de pesquisa: bibliográfico, exploratório e qualitativo, sendo que testes práticos foram adotados também para alcançar os resultados esperados. Para o ambiente de testes foram utilizados 5 computadores virtualizados, dentre eles 3 clientes e os outros 2 servidores, sendo estes últimos, implementações dos serviços utilizados na pesquisa. Os resultados permitiram identificar que a integração das ferramentas foi bem-sucedida, pois possibilitou aos usuários, previamente cadastrados, em um servidor LDAP, terem o acesso para gerenciar as suas pastas e arquivos localizados no servidor OwnCloud.*

Palavras-chave. *OwnCloud. LDAP. Integração do servidor LDAP ao servidor OwnCloud. Armazentamento em Nuvem. Autenticação Centralizada.*

1 Introdução

Computação em nuvem (*Cloud Computing*), conforme Amazon (2018a, s.p.), "[…] é a entrega sobre demanda de poder computacional, armazenamento de banco de dados, aplicações e outros recursos de Tecnologia da informação (TI) […]", sem a necessidade da instalação de *softwares* no equipamento físico, pois a execução e processamento desse serviço é realizado na Internet (ZANUTTO, 2017). Amazon (2018b, s.p.), afirma que existe também o armazenamento em nuvem (*Cloud Storage*), que "[…] é um modelo de computação em nuvem que armazena dados na Internet por meio de um provedor de computação em nuvem, que gerencia e opera o armazenamento físico de dados

como serviço […]", permitindo que os dados armazenados em um servidor remoto na nuvem sejam compartilhados e acessados em qualquer dispositivo conectado à Internet (MACTEC, 2017).

Buscando a autenticação de usuários de maneira centralizada e o acesso e gerenciamento de arquivos, em nuvem, e em uma rede local, este trabalho propõe uma solução para instituições que precisam compartilhar arquivos com usuários pré-determinados.

Para que os usuários não tenham dificuldades de acesso ou falta de armazenamento, o OwnCloud, traz uma interface gráfica amigável com suporte a diferentes dispositivos e sistemas operacionais, possibilitando fazer a transferência e compartilhamento de arquivos em nuvem. O uso do OpenLDAP permitirá aos serviços, que suportam o protocolo LDAP, restrinjirem realmente o acesso apenas aos usuários selecionados, trazendo maior segurança e facilidade para migração deste usuário e senha para outros serviços.

Fogaça (2011, s.p.) afirma que, o OwnCloud é um "serviço de armazenamento na nuvem de código aberto", ele permite a sincronização e compartilhamentos de arquivos *online* de maneira diferenciada. Para Morimoto (2005, s.p.), o LDAP é um protocolo que "permite organizar os recursos de rede de forma hierárquica", ele é prático quanto a sua funcionalidade de localização de arquivos, pois sendo ele distribuído de maneira hierárquica, um dado, como o nome do usuário, acaba mostrando outros dados referentes a ele.

O uso das ferramentas OwnCloud e OpenLDAP surgiu com a ideia de disponibilizar este serviço integrado de forma acessível a todos tipos de empresa, pois como as duas ferramentas são *softwares livres*, a implementação delas será mais amplificada. Outro benefício das ferramentas, é que caso elas não estejam de acordo com os requisitos do usuário, é

possível realizar modificações, pois as duas possuem documentação disponível na Internet, facilitando, assim, a substituição das funcionalidades requeridas ou adequação das mesmas.

O objetivo principal deste trabalho é implantar o OwnCloud como ferramenta de compartilhamento de arquivos, em nuvem, integrado ao serviço OpenLDAP, visando possibilitar a autenticação de maneira centralizada com outros serviços. Para isso, serão realizadas as seguintes etapas:

- Realização de pesquisa bibliográfica sobre o tema;
- Estudo das ferramentas: OwnCloud, OpenLDAP e o protocolo LDAP;
- Virtualização do servidor OwnCloud para o gerenciamento e transferência de arquivos;
- Integração do LDAP ao servidor OwnCloud para a autenticação centralizada;
- Realização da implementação e testes nos computadores clientes.

O trabalho foi dividido em cinco seções, sendo que na seção 2 são explicados os tópicos principais no referencial teórico. A seção 3 traz os materiais e métodos utilizados no trabalho. Na seção 4 tem-se os resultados e discussões da pesquisa e a seção 5 apresenta as considerações finais.

2 Referencial Teórico

Nesta seção serão abordados tópicos necessários visando o entendimento da temática principal do trabalho.

2.1 Rede de Computadores

Segundo Gouveia e Magalhães (2013, s.p.), "[…] rede de computadores é composta por dois ou mais computadores ligados entre si, de modo a poderem compartilhar recursos, dados e programas. […]", sendo que estes dispositivos podem ser conectados via cabo metálico, fibra ótica ou até mesmo em uma ligação sem fio (wireless).

Maya (2016) afirma que o uso das redes de computadores é voltado: para a facilitação da comunicação dos usuários através de diferentes tipos de aplicações como o *e-mail* e videoconferência; permite o compartilhamento de dispositivos como uma impressora e também torna possível o compartilhamento de arquivos como documentos de texto, vídeos, fotos, entre outros.

2.2 Serviços De Redes

Conforme Rios (2011), os serviços de redes podem ser oferecidos por diferentes tipos de protocolos em uma rede de computadores, visando cumprir as tarefas requeridas pelo usuário.

Maziero (2008, s.p.) descreve os serviços de rede como "[…] uma aplicação distribuída, que executa em dois ou mais computadores conectados por uma rede. […]", e eles devem conter pelo menos quatro elementos: servidor (usa seus recursos locais para realizar a parte central do serviço); cliente (computador solicitante do serviço), protocolo (conjunto de dados necessários para que haja uma boa comunicação entre o computador cliente e o servidor, fazendo assim com que o servidor consiga realizar o serviço requerido) e *middleware* (responsável por fazer o suporte para a construção e execução dos serviços).

2.3 Computação em Nuvem (*Cloud Computing*)

De acordo com Silva (2010), a computação em nuvem ou *Cloud Computing* é um modelo novo de computação que permite ao usuário acessar aplicações e serviços, independente da plataforma escolhida e em qualquer lugar, bastando, para isso, ter um terminal conectado à "nuvem". Para o entendimento geral deste modelo de computação, é necessário identificar seus três agentes participantes, que podem ser divididos em: Provedor de serviço, Desenvolvedor e Usuário. O provedor tem como principal função o gerenciamento, disponibilização e monitoramento de toda a infraestrutura da nuvem, garantindo assim, um nível do serviço e segurança adequados dos dados e aplicações. Já o desenvolvedor, através do uso da infraestrutura disponibilizada pelo provedor de serviço, deve ter a capacidade de prover os serviços para o usuário final. Enquanto o usuário final é quem utilizará e consumirá os recursos oferecidos.

E segundo Hurwitz *et al.* (2010), a "nuvem", é representada de maneira geral pela Internet, sendo que Mell e Grance (2009) afirmam que sua implantação segue quatro modelos, são eles: nuvem pública, privada, comunidade e híbrida.

De acordo com Sousa, Moreira e Machado (2009), os quatro modelos de implantação podem ser definidos da seguinte maneira: "nuvem pública": a infraestrutura é disponibilizada para o público de maneira geral, e não podem ser implantadas políticas de restrição de acesso ou em relação ao gerenciamento das redes; "nuvem privada": neste modelo a infraestrutura é disponibilizada apenas para uma organização, podendo ser acessada localmente ou remotamente, e administrada pela própria empresa e ou terceiros; "nuvem comunidade": sua implantação ocorre com o compartilhamento desta nuvem por diversas empresas, tendo o suporte a partir de uma comunidade específica, que possui interesses em comum. Neste modelo a nuvem pode ser gerenciada por terceiros ou por empresas

presentes nessa comunidade; “nuvem híbrida”: na composição deste tipo de nuvem podem haver várias nuvens diferentes, onde elas podem ser privadas ou públicas ou comunidade, não perdendo a característica, pois elas são conectadas por uma tecnologia padronizada ou privada, que possibilita a portabilidade dos seus dados e aplicações

2.3.1 OwnCloud

De acordo com OwnCloud (2018), o OwnCloud é um *software* livre/aplicação *web* que foi escrito na linguagem PHP e que possibilita uma configuração ágil e segura de uma nuvem privada. Este serviço possui as seguintes características: suporte às diferentes plataformas disponíveis tanto na versão *desktop* quanto *mobile*, como Linux, Windows, MacOS, Android e iOS; possibilidade de armazenamento e compartilhamento de arquivos em nuvem; possibilidade de criptografia dos dados armazenados para melhor segurança e também possui documentação disponível na Internet, possibilitando, assim, uma melhor adaptação de suas funcionalidades.

Venezuela (2014) apresenta as seguintes vantagens do uso do OwnCloud: a segurança – pois no momento da instalação da ferramenta é possível escolher criptografar os dados, além da opção de conexão segura; ganho de produtividade para empresas – pois a possibilidade de armazenamento, compartilhamento de arquivos de maneira remota e acesso disponível em diferentes plataformas, torna o acesso mais fácil e a disseminação das informações úteis e necessárias na empresa, mais rápida.

2.4 Protocolo de rede

Segundo Sousa (2010), protocolo de rede é um *software* responsável por receber ou enviar os dados que precisam ser transmitidos, para que não ocorram erros na transmissão ou falhas na segurança, agregrando informações importantes no início e no final destes dados.

LinuxDictionary (2004, s.p.) define protocolo de rede como:

> […] Um conjunto de regras que define exatamente como as informações devem ser trocadas entre dois sistemas. Isso permite que diferentes tipos de máquinas se comuniquem em um formato que ambos compreendam. […].

Moraes (2013) afirma que, "protocolos são regras e procedimentos de comunicação".

De acordo com Fernandes (2016, s.p.):

> Protocolo é uma convenção que controla e possibilita uma conexão, comunicação, transferência de dados entre dois sistemas computacionais. De maneira simples, um protocolo pode ser definido como "as regras que governam" a sintaxe, semântica e sincronização da comunicação. Os protocolos podem ser implementados pelo *hardware*, *software* ou por uma combinação dos dois. Existem diversos tipos de protocolos e cada um com uma função específica.

2.4.1 Protocolo LDAP

O LDAP (*Lightweight Directory Access Protocol*) é um protocolo que permite a troca de mensagens entre clientes e

servidores de diretórios. Nele o cliente envia um pedido de informações ao servidor que faz o processamento delas. Além disso, o servidor realiza as operações necessárias em seu diretório de busca, para então estar apto para enviar respostas. Em uma resposta pode conter as informações solicitadas pelo cliente ou os erros que ocorreram ao longo do pedido do serviço (RFC 2251, 1997).

Conforme Scrimger, Lasalle e Parihar (2002), o LDAP por ser rápido, leve e de fácil implementação, não é classificado apenas como um protocolo, mas também um diretório de serviço escalonável, portanto, pode-se, assim, aceitá-lo como o padrão Internet para serviços de diretórios que sejam executados sobre a arquitetura TCP/IP[24].

De acordo com Carter (2009), a criação do LDAP está relacionada ao aumento da densidade de padrões apresentados pelo serviço de diretório X.500. Por conta disso, este serviço acabou sendo considerado muito "pesado" para fazer a comunicação entre cliente e servidor e então "[…] O LDAP foi projetado originalmente como um protocolo de *desktop* mais leve usado para intermediar a comunicação entre solicitações e servidores X.500 […]".

2.5 Ferramenta OpenLDAP

De acordo com OpenLDAP (2018), o OpenLDAP é um software com código aberto voltado a implementação do protocolo LDAP.

Segundo Augusto (2017), o OpenLDAP é distribuído para os principais sistemas operacionais do mercado como o

[24] De acordo com Palma e Prates (2000), o TCP/IP "é um conjunto de protocolos usados em redes de computadores".

Unix, Microsoft Windows e MacOS, e ele oferece os seguintes recursos: integração com os principais protocolos de comunicação utilizados; integração de chaves criptográficas e banco de dados, além de fazer a autenticação centralizada dos usuários cadastrados por meio da sua base de dados.

Conforme Oliveira (2017, s.p.):

> Entre as características principais do OpenLDAP estão a segurança durante o processo de transporte, a capacidade de alta performance durante a realização de múltiplas chamadas, o controle de acessos, a habilidade de poder atender a diferentes – e múltiplos – bancos de dados ao mesmo tempo, a replicação de base e muitas outras funcionalidades.

2.6 Ferramenta phpLDAPadmin

Segundo phpLDAPadmin (2011), essa é uma ferramenta voltada para o gerenciamento *web* e administração do LDAP, além disso, é capaz de:

> "gerenciar registros em um servidor LDAP, incluindo a criação, modificação e exclusão destes. Possui padrões abertos em conformidade com o protocolo LDAP, para que seja possível o gerenciamento a partir de qualquer servidor LDAP compatível".

3 Materiais e métodos

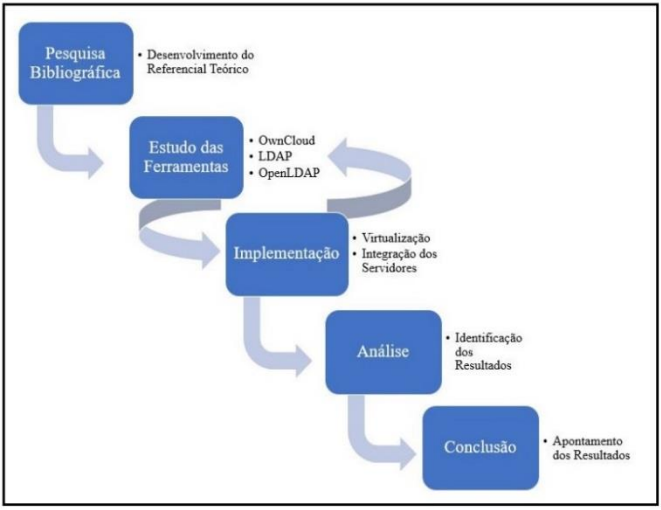
A fim de cumprir todos os objetivos apresentados neste trabalho, foram utilizados os métodos de pesquisa bibliográfico, exploratório e qualitativo para melhor descrever a metodologia utilizada. A pesquisa bibliográfica visa o reconhecimento dos dados bibliográficos vindos de diferentes materiais já

publicados, como revistas, livros, artigos, entrevistas, entre outros (MARCONI; LAKATOS, 2009).

Cervo, Bervian e Silva (2007) descrevem que a pesquisa exploratória tem como objetivo realizar descrições precisas entre os dados e/ou situações estudados, com planejamento flexível, para poder, assim, considerar e reconhecer todas as possibilidades e então descobrir as relações existentes.

De acordo com Minayo, Deslandes e Gomes (2011, s.p.), a pesquisa qualitativa "[…] trabalha com o universo dos significados, dos motivos, das aspirações, das crenças, dos valores e das atitudes. […]". O detalhamento das principais atividades desenvolvidas neste trabalho é representado no fluxograma exposto na Figura 1.

Figura 1 – Fluxograma das atividades



Fonte: Os Autores, 2018.

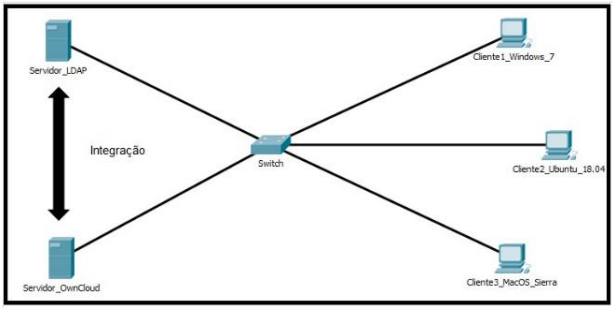
Para a realização da etapa de pesquisa bibliográfica foram utilizadas como fontes de pesquisa artigos científicos e livros. Os materiais foram recolhidos tanto na Internet quanto em bibliotecas físicas visando esclarecer os principais tópicos explanados e descritos neste trabalho.

Na etapa de estudo das ferramentas, foram identificadas quais seriam as mais apropriadas para a realização do estudo, isto foi feito, considerando os seguintes critérios: se os *softwares* eram livres, se existia a possibilidade de integração entre as ferramentas e se existia documentação oficial das ferramentas com informações referentes à instalação e integração dos softwares utilizados. Para a implementação foi utilizado como material um *notebook*, com as seguintes configurações: processador Intel Core i7-7700HQ, 16 GB de memória RAM DDR4 e 480 GB de SSD. Além disso, foi utilizado o *software* VirtualBox na versão 5.2.22, para realizar a virtualização das máquinas necessárias para os testes. O ambiente configurado conta com 5 máquinas virtuais, sendo que 2 máquinas contém o sistema operacional Debian 9 para os servidores LDAP e OwnCloud; e 3 clientes, onde uma máquina contém o sistema operacional Windows 7, outra o Ubuntu 18.04 e outra com o MacOS Sierra. Para o funcionamento das máquinas virtuais e da conexão, foi necessário realizar a instalação dos sistemas operacionais, identificar os *hosts*[25], clientes e servidores, através de seus IPs[26] e placas de rede. Na Figura 2 é apresentada a topologia desta rede.

[25]De acordo com Viana (2012), "*host* é qualquer computador ou dispositivo conectado a uma rede, que conta com número de IP e nome definidos".

[26]Sousa (2010) afirma que o endereço IP identifica de maneira única os equipamentos ligados em uma rede, para que eles consigam receber e enviar dados para redes diferentes.

Figura 2 – Topologia da rede



Fonte: Os autores, 2018.

Antes de definir os IPs das máquinas, foram selecionadas as placas de rede, sendo que todas as máquinas seguiram o padrão de ter uma placa com Rede NAT e outra placa como Rede Interna. Feito isso, foram selecionados os IPs estáticos para as placas da Rede Interna de cada máquina, os *hosts* e seus respectivos IPs são apresentados na Tabela 1.

Tabela 1 – Identificação dos *hosts* e seus IPsFonte: Os autores, 2018.

Servidor/Cliente - Sistema Operacional	**IP Definido**
Servidor LDAP – Debian 9	192.168.0.251
Servidor OwnCloud – Debian 9	192.168.0.252
Cliente 1 – Windows 7	192.168.0.10
Cliente 2 – Ubuntu 18.04	192.168.0.20
Cliente 3 – MacOS Sierra	192.168.0.30

3.1 Configuração do Servidor OwnCloud

Após a identificação dos computadores, foi realizada a instalação e configuração do servidor OwnCloud na versão 10.0, no sistema operacional Debian versão 9. Para isso, foi necessário instalar algumas ferramentas adicionais, que são pré-requisitos para que ocorra o bom funcionamento geral do OwnCloud, como o servidor HTTP Apache[27] na versão 2, pacotes específicos da linguagem PHP[28] na versão 7.0, voltada principalmente para o desenvolvimento *web*, o SQLite[29] na versão 3, que é uma biblioteca que implementa um mecanismo de banco de dados SQL, e foi instalado também o MySQL Server, que é um sistema para gerenciar banco de dados que usa a linguagem SQL como interface. No Quadro 1 é apresentada a sequência de comandos necessários para fazer a instalação e configuração do OwnCloud e de outros pacotes necessários.

[27]https://apache.org/foundation/

[28]http://secure.php.net/manual/pt_BR/intro-whatis.php

[29]https://www.sqlite.org/about.html

Quadro 1 – Comandos da instalação do OwnCloud e de configuração de outros pacotes necessários

```
root@owncloud-server:~# apt-get install -y apache2 mariadb-server libapache2-mod-php7.0 openssl php-imagick
    php7.0-common php7.0-curl php7.0-gd php7.0-imap php7.0-intl php7.0-json php7.0-ldap php7.0-mbstring php7.0-
    mysql php7.0-pgsql php-smbclient php-ssh2 php7.0-sqlite3 php7.0-xml php7.0-zip

root@owncloud-server:~# wget -nv
    https://download.owncloud.org/download/repositories/production/Debian_9.0/Release.key -O Release.key

root@owncloud-server:~# apt-key add - < Release.key

root@owncloud-server:~# echo 'deb http://download.owncloud.org/download/repositories/production/Debian_9.0/ /'
    > /etc/apt/sources.list.d/owncloud.list

root@owncloud-server:~# apt-get update

root@owncloud-server:~# apt-get install owncloud-files
```

Fonte: Os autores, 2018.

Feito isso, é necessário configurar o arquivo "/etc/apache2/sites-available/owncloud.conf" para que o Apache, servidor *web*, reconheça qual o diretório que os arquivos do OwnCloud ficarão armazenados, sendo definido o diretório: "/var/www/html/owncloud", para este propósito. Além disso, foram definidos alguns parâmetros de configuração voltados ao funcionamento da integração das duas ferramentas. Na Figura 3 encontra-se a configuração do arquivo "owncloud.conf".

Figura 3 – Configuração do arquivo “owncloud.conf”

Fonte: Os autores, 2018.

Com as configurações definidas, foi necessário realizar a ativação através do comando “a2enmod”, dos seguintes módulos do Apache: *rewrite*, *headers*, *env*, *dir*, *mime*, *ssl* e *default-ssl*; Em seguida foi realizada a reinicialização destes módulos a partir do comando: “service apache2 restart”.

Terminada a configuração do apache, foi necessário configurar o grupo do diretório e suas permissões de uso, onde os arquivos do OwnCloud ficam localizados. Para isso, foi usado o seguinte comando: “chown -R www-data:www-data /var/www/html/owncloud/”. Depois é preciso configurar um módulo específico no PHP chamado: “pdo_mysql”. Para realizar esta configuração foi utilizado o comando: “phpenmod pdo_mysql”. E para que as configurações fossem aceitas pelo servidor, o serviço apache foi novamente reiniciado.

Com os serviços Apache e PHP configurados, é iniciada a configuração do MySQL. Primeiramente foi realizado o *login* no MySQL, com usuário padrão chamado root, através do seguinte comando: “sudo mysql -u root”. Depois criou-se um

banco de dados específico para armazenar os dados dos usuários OwnCloud. Para a criação deste banco de dados usou-se o comando: “create database owncloud;”. Criado o banco de dados, é necessário criar um usuário administrador que o gerencie, e que tenha acesso a todas as informações. Para a criação do usuário administrador “owncloudadm”, foram executados os seguintes comandos:

- CREATE USER "owncloudadm" IDENTIFIED BY "123";
- GRANT ALL PRIVILEGES ON owncloud.* TO owncloudadm;
- FLUSH PRIVILEGES;

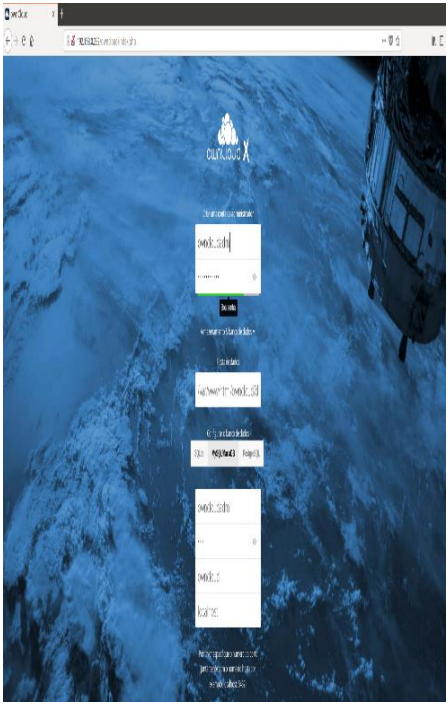
Configurados os serviços pré-requisitados, partiu-se para a etapa de configuração do OwnCloud, fazendo o uso da interface gráfica do navegador do cliente 2, com sistema operacional Ubuntu 18.04. Para isso, foi necessário digitar: “http://192.168.0.252/owncloud” na barra de endereços do navegador, e então preencher os seguintes dados requisitados na tela inicial do OwnCloud para o reconhecimento do servidor:

- Criar um usuário administrador do OwnCloud e uma senha para ele;
- Identificar o diretório onde ficam localizados os dados dos usuários do OwnCloud, neste servidor eles ficam no diretório: “/var/www/html/owncloud/data”;
- Selecionar qual serviço de banco de dados será utilizado, neste servidor foi configurado o MySQL;
- Identificar qual o usuário e senha do usuário administrador do banco de dados configurado, e o nome deste banco de dados;

- Identificar qual o *host* que é responsável pelo acesso e configuração principal do OwnCloud, neste caso foi indicado o *host* localhost.

Na Figura 4 é possível visualizar as etapas dessa configuração inicial para acesso do OwnCloud.

Figura 4 – Configuração inicial para acesso do OwnCloud



Fonte: Os autores, 2018.

3.2 Configuração do Servidor LDAP

Após o acesso ao servidor OwnCloud, através da interface gráfica, foi possível criar pastas e usuários testes para reconhecimento do ambiente. Após isso, foi iniciada a

configuração do servidor LDAP. A Tabela 2 apresenta os comandos utilizados para este propósito e qual a função de cada um deles.

Tabela 2 – Comandos para a configuração do servidor LDAP

Função	Comando
Instalação da ferramenta OpenLDAP	sudo apt-get install slapd ldap-utils
Configuração detalhada do OpenLDAP	sudo dpkg-reconfigure slapd
Lista as configurações indicadas para o servidor	sudo ldapsearch -xb dc=testetcc, dc=com, dc=br
Instalação da ferramenta phpLDAPadmin	sudo apt-get install phpldapadmin

Fonte: Os autores, 2018.

Após a instalação do OpenLDAP, é preciso realizar o seguinte comando: “sudo dpkg-reconfigure slapd”, pois ele é necessário para definir alguns dados de configuração que são importantes para o servidor, como: o domínio DNS da rede, que terá os usuários cadastrados no servidor LDAP, neste servidor foi configurado o seguinte domínio: “testetcc.com.br”; o nome da organização, que neste servidor foi definido como: “Teste TCC” e a senha do usuário administrador, dentre outras configurações.

Para a compreensão básica da configuração do servidor LDAP e do phpLDAPadmin, é necessário notar que existem algumas nomenclaturas específicas, que de acordo com a RFC 4519 (2006), elas são descritas como: “Domain Component” ou “Componente de Domínio” (dc) – atributo que armazena um rótulo ou um nome de domínio DNS; “Common Name” ou “Nome Comum” (cn) – atributo que contém o nome de um objeto, se for uma pessoa, este objeto armazena seu nome completo; “Organizational Unit Name” ou “Nome da Unidade

Organizacional" (ou) – atributo que contém os nomes de uma unidade organizacional.

Após a configuração detalhada do OpenLDAP é necessário listar e confirmar se estas configurações informadas anteriormente funcionarão e estão corretas. O responsável por essa verificação é o comando: "sudo ldapsearch -xb dc=testetcc, dc=com, dc=br". Feito isso, foi realizada a instalação da ferramenta phpLDAPadmin na versão 1.2.2 visando configurar o servidor através do uso da interface gráfica de uma máquina cliente. Para que o phpLDAPadmin identifique corretamente os dados cadastrados no servidor LDAP é necessário acessar e editar o seguinte arquivo de configuração através do comando: "sudo nano /etc/phpldapadmin/config.php". O Quadro 2 mostra as configurações editadas no arquivo "config.php".

Quadro 2 – Configuração do arquivo "config.php"

```
$servers->setValue('server','name','LDAP Teste TCC');
$servers->setValue('server','host','192.168.0.251');
[...]
$servers->setValue('server','base',array('dc=testetcc,dc=com,dc=br'));
[...]
$servers->setValue('login','bind_id','cn=admin,dc=testetcc,dc=com,dc=br');
```

Fonte: Os autores, 2018.

Para acessar o phpLDAPadmin, foi digitado na barra de endereços do navegador do cliente 1, com sistema operacional Windows 7, a seguinte linha: "http://192.168.0.251/phpldapadmin". Fazendo o acesso com o usuário administrador, foi definido uma hierarquia para criar e organizar os grupos de usuários; primeiramente foi criada a unidade organizacional (ou) com o nome de IFC, e depois dois subgrupos, classificados pelo LDAP de Grupo Posix (cn), um com nome de Alunos e outro com o nome de Docentes; dentro

de cada Grupo Posix foram criados 2 usuários, todos configurados com o atributo: “o” (Nome da organização), este atributo é necessário para que o servidor OwnCloud consiga identificar os usuários criados. Na Figura 5 é possível observar a hierarquia criada.

Figura 5 – Hierarquia de grupos criada no phpLDAPadmin

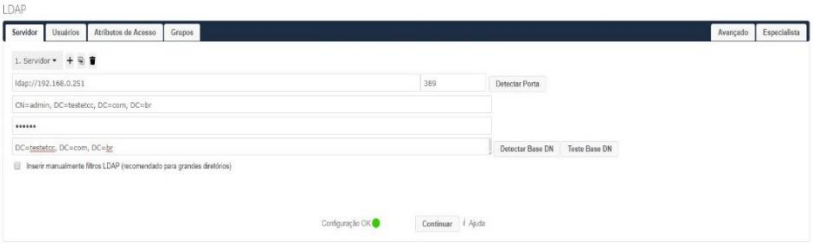
Fonte: Os autores, 2018.

3.3 Integração dos servidores

Finalizada as configurações dos servidores OwnCloud e LDAP de maneira individual, foi necessário acessar, através do navegador da máquina cliente 1, com sistema operacional Windows 7, o servidor OwnCloud para realizar a integração dos servidores. Para isso foi necessário entrar no sistema com o usuário administrador do OwnCloud, ir no menu localizado na parte superior esquerda da tela e selecionar a aba “market”; dentro desta aba existem diversas aplicações para instalar, mas para esta integração foi selecionada e instalada a aplicação de nome “Integração LDAP”.

Após esta etapa, a integração já pode ser iniciada, sendo necessário ir no menu configurações do usuário administrador, localizado na parte superior direita da tela, depois no menu autenticação de usuário. Nesta parte aparece uma tela inicial com 4 abas. Na primeira aba, chamada Servidor, são requeridos alguns dados requeridos pelo servidor, como: IP, usuário e senha do administrador e o DNS do servidor LDAP, todos eles foram preenchidos de acordo com as configurações já realizadas na seção anterior, e antes de prosseguir a configuração é preciso clicar nos botões: “Detectar Base DN”, e “Teste Base DN”, pois eles identificam se a base de dados criada no LDAP está sendo reconhecida pelo servidor OwnCloud. Na Figura 6 é possível observar como foi configurada essa parte da integração dos servidores.

Figura 6 – Tela de configuração dos dados requeridos do servidor LDAP

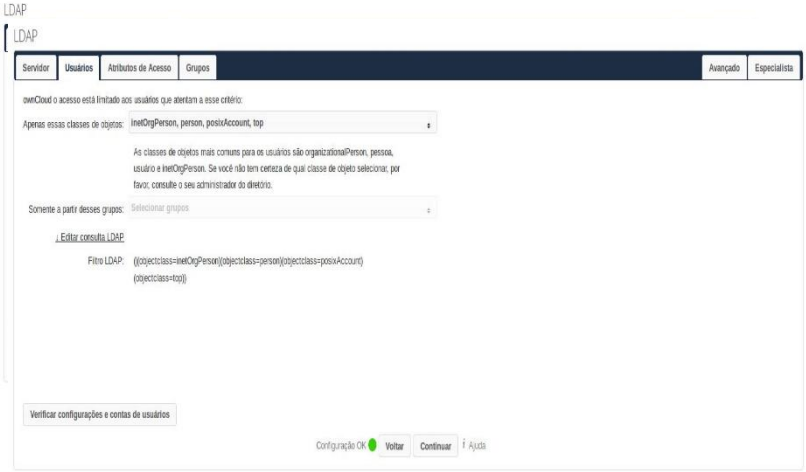


Fonte: Os autores, 2018.

Nas outras três abas: Usuários, Atributos de Acesso e Grupos, foram definidos quais os atributos/classes de objetos que devem ser reconhecidos e que foram realmente utilizados no servidor LDAP. Na aba “Usuários” foram definidas as seguintes classes de objetos: “inetOrgPerson”, “person”, “posixAccount” e “top” pois todos os usuários cadastrados pertenciam e eram configurados com essas classes. Na aba “Atributos de Acesso”,

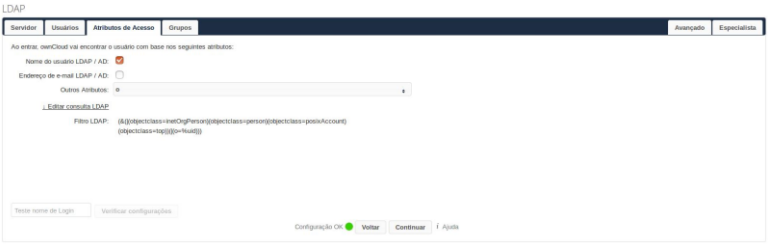
foi selecionada a opção: “Nome de usuário LDAP/AD”, e definido como atributo principal, o atributo “o” (Nome da organização); como todos os usuários possuem seus *login* e senha cadastrados no servidor LDAP, o atributo “o” permite que o OwnCloud consiga identificá-los e integrá-los para que eles tenham sua conta para armazenar seus arquivos. E na aba “Grupos”, foram definidos as classes de objeto: “posixGroup” e “top”; os grupos selecionados foram: “Alunos” e “Docentes”; as configurações dessa aba seguiram a hierarquia dos grupos criados no servidor LDAP e seus atributos selecionados no momento de sua criação. Nas Figuras 7, 8 e 9 é possível observar as telas de configurações das 3 abas descritas acima.

Figura 7 – Tela de configuração da aba Usuários



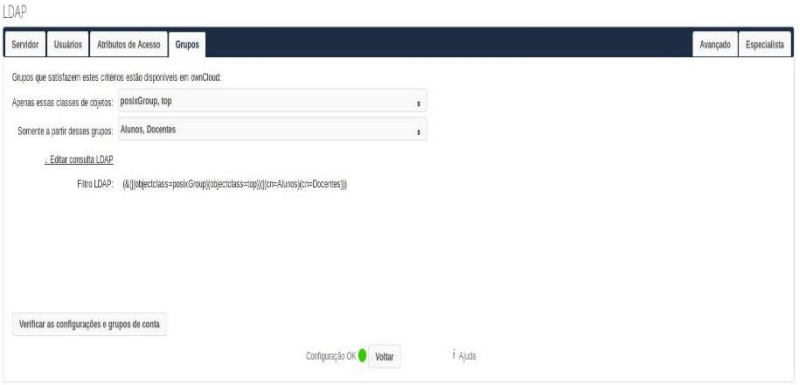
Fonte: Os autores, 2018.

Figura 8 – Tela de configuração da aba Atributos de Acesso



Fonte: Os autores, 2018.

Figura 9 – Tela de configuração da aba Grupos



Fonte: Os autores, 2018.

A etapa identificada como análise foi realizada posteriormente e os dados levantados serão descritos na seção seguinte.

4 Resultados e discussões

Após essa etapa de integração basta identificar se os usuários criados na base de dados do servidor LDAP realmente foram reconhecidos pelo servidor OwnCloud. Para isso, deve-se acessar o OwnCloud no

navegador, com o usuário administrador, depois clicar no menu superior direito e ir na aba Usuários. A Figura 10 apresenta os usuários identificados pelo servidor OwnCloud.

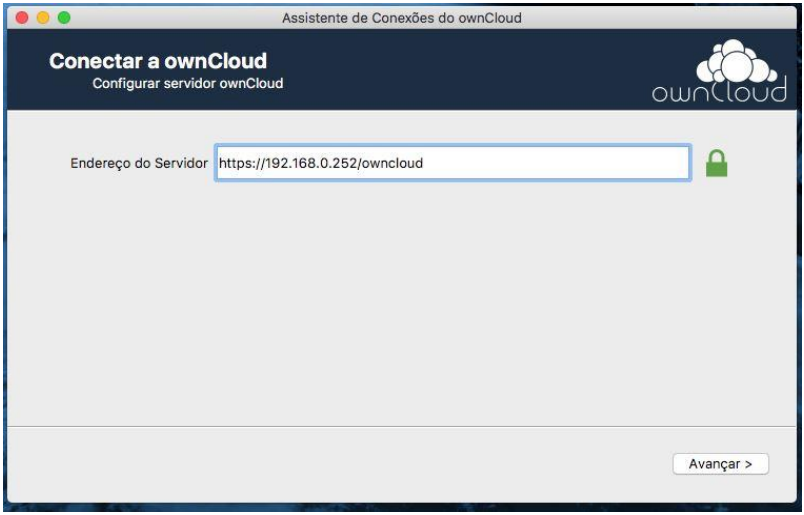
Figura 10 – Tela de identificação dos usuários no servidor OwnCloud

Fonte: Os autores, 2018.

Feito isso, os usuários já podem ter seus acessos liberados aos seus arquivos, tanto através do navegador, digitando: “http://192.168.0.252/owncloud” e digitando seus usuários e senhas definidos na base de dados do servidor LDAP.

Para instalação do aplicativo OwnCloud nas máquinas, que está disponível para download nos três tipos de sistemas operacionais utilizados neste trabalho, basta indicar qual o IP da máquina do servidor OwnCloud, a pasta principal de arquivos do servidor e seu usuário e senha. A Figura 11 apresenta a tela principal de configuração que aparece na instalação das três máquinas clientes.

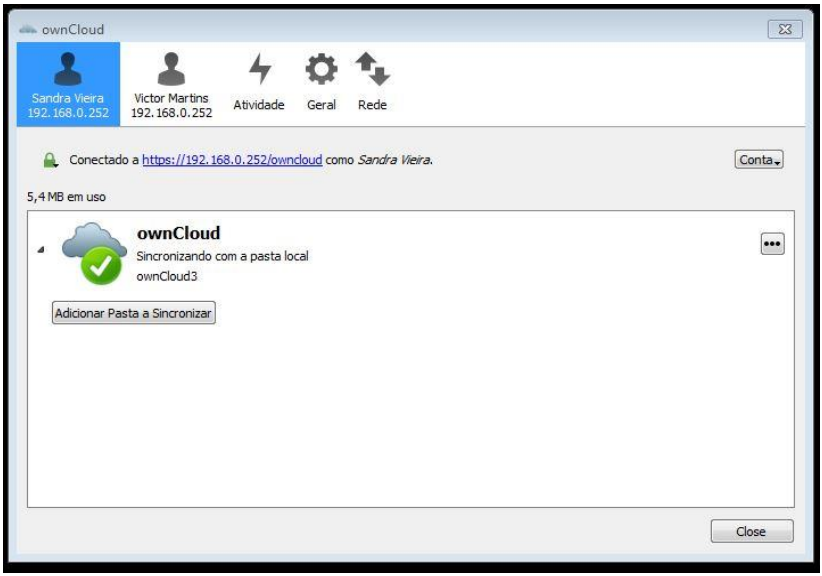
Figura 11 – Tela de identificação do servidor OwnCloud nas máquinas clientes



Fonte: Os autores, 2019

Depois de ter feito o acesso através de seu usuário e senha dentro da aplicação, pode-se fazer o *login* de múltiplos usuários em uma mesma máquina para fazer a transferência de seus arquivos para o servidor OwnCloud. Na Figura 12 é apresentada a tela principal de configuração da aplicação utilizada pelas máquinas clientes.

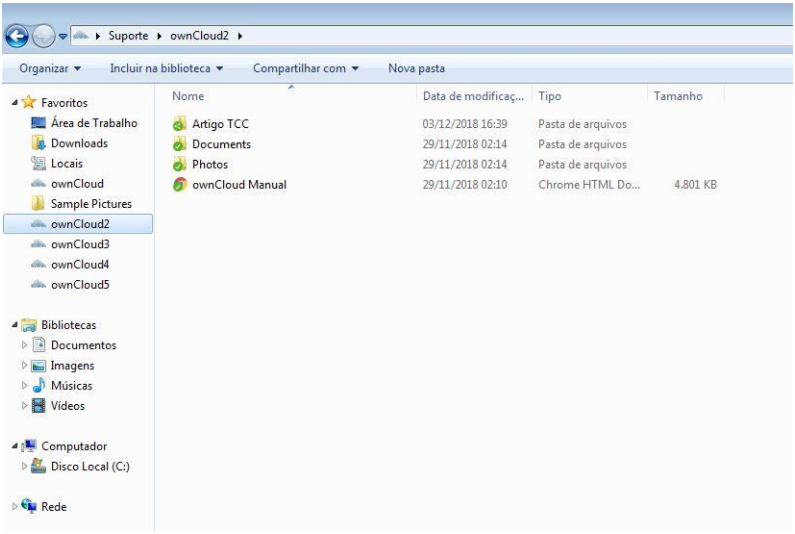
Figura 12 – Tela principal de configuração da aplicação OwnCloud nas máquinas clientes



Fonte: Os autores, 2018.

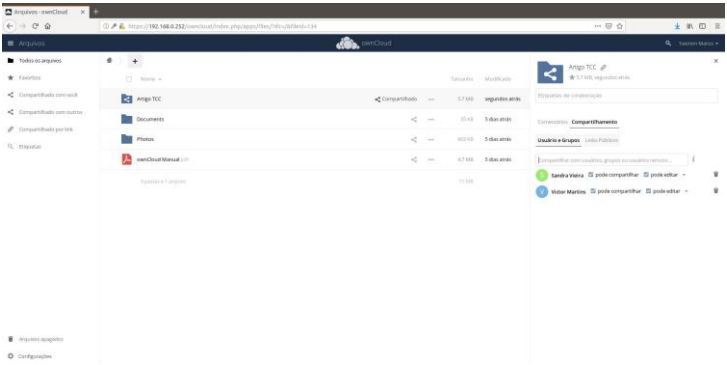
Tendo o acesso à tela principal de sincronização de arquivos, os clientes já podem fazer a transferência dos seus dados para o servidor OwnCloud. Nas Figuras 13, 14 e 15 é possível visualizar a tela de acesso dos clientes Windows 7, Ubuntu 18.04 e MacOS Sierra dos seus arquivos através do navegador e pastas sincronizadas no computador. Neste exemplo utilizamos o usuário “Yasmim Matos”, com a pasta criada “Artigo TCC” que está sendo compartilhada com os usuários “Sandra Vieira” e “Victor Martins”.

Figura 13 – Tela de acesso da pasta sicronizada do OwnCloud no cliente Windows 7



Fonte: Os autores, 2018

Figura 14 – Tela inicial de acesso do OwnCloud no navegador do cliente Ubuntu 18.04



Fonte: Os autores, 2018.

5 Considerações finais

O desenvolvimento do trabalho, a partir das suas etapas, permitiram atingir o objetivo que era desenvolver um processo de implementação de serviços, integrando uma ferramenta de armazenamento em nuvem e uma base de dados para autenticação centralizada de usuários. Para isso, foram utilizados o OwnCloud e o serviço OpenLDAP, visando disponibilizar o serviço de autenticação de maneira centralizada.

As informações levantadas a partir da integração do servidor OpenLDAP ao servidor OwnCloud para a autenticação centralizada e a transferência de arquivos, mostraram-se suficientes para atingir o objetivo proposto, visto que, as ferramentas selecionadas para a implementação, foram capazes de executar as ações sem apresentar problemas.

Para trabalhos futuros, recomenda-se que sejam realizados estudos sobre sistema de *backup* e identificação e contingenciamento de redundâncias no armazenamento dos arquivos gerenciados pelo OwnCloud. Também é sugerido a realização de testes com clientes de plataformas *Mobile* para *Smartphones* e *Tablets*, com sistemas operacionais Android e IOS, além da possível implantação em ambiente real das ferramentas aplicadas neste trabalho, já que a última etapa realizada foi a execução dos testes nos computadores dos clientes virtuais, visto que os testes realizados, foram executados em um ambiente virtual.

Referências

AMAZON. **O que é a computação em nuvem?** 2018. Disponível em: <https://aws.amazon.com/pt/what-is-cloud-computing/>. Acesso em: 28 Jul. 2018a.

_____. **O que é o armazenamento em nuvem?** 2018. Disponível em: <https://aws.amazon.com/pt/what-is-cloud-storage/>. Acesso em: 29 Jul. 2018b.

AUGUSTO, Cassio. **OpenLDAP – O que é e para que serve?** 2017. Disponível em: <http://ninjadolinux.com.br/openldap-o-que-e-e-para-que-serve/>. Acesso em: 26 Set. 2018.

CARTER, Gerald. **LDAP: Administração de Sistemas**. 2009.

CERVO, Amado L.; BERVIAN, Pedro A.; SILVA, Roberto da. **Metodologia Científica**. 2007.

FERNANDES, André. **Informática Facilitada para Concursos**. 2016. Disponível em: <https://3dconcursos.com.br/arquivos/1476991687.98-arquivo-N.pdf>. Acesso em: 31 Jul. 2018.

FOGAÇA, André. **Conheça o OwnCloud, o serviço de armazenamento em nuvem privado**. 2011. Disponível em: <http://www.techtudo.com.br/artigos/noticia/2011/10/conheca-o-owncloud-o-servico-de-armazenamento-em-nuvem-privado.html>. Acesso em: 29 Out. 2018.

GOUVEIA, José; MAGALHÃES, Alberto. **Redes de Computadores**. 2013.

HURWITZ, Judith; BLOOR, Robin; KAUFMAN, Marcia; HALPER, Fern. **Cloud Computing for Dummies**. 2010.

LINUXDICTIONARY. **Protocol**. 2004. Disponível em: <http://tldp.org/LDP/Linux-Dictionary/html/p.html>. Acesso em: 14 Ago. 2018.

MACTEC. **Cloud Computing e Cloud Storage: Qual a diferença?** 2017. Disponível em: <http://blogmactec.com.br/?p=120>. Acesso em: 12 Set. 2018.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Metodologia do Trabalho Científico**. 2009.

MAYA, Alcides. **O que são redes de computadores?** 2016. Disponível em: <http://www.alcidesmaya.com.br/blog/o-que-sao-redes-de-computadores/>. Acesso em: 09 Set. 2018.

MAZIERO, Carlos. **Introdução aos Serviços de Rede**. 2008. Disponível em: <http://wiki.inf.ufpr.br/maziero/doku.php?id=espec:introducao>. Acesso em: 09 Out. 2018.

MELL, P.; GRANCE, T. **Draft NIST Working Definition of Cloud Computing**. 2009. Disponível em: <http://csrc.nist.gov/groups/SNS/cloud-computing>. Acesso em: 12 Set. 2018.

MINAYO, Maria Cecília de Souza; DESLANDES, Suely Ferreira; GOMES, Romeu. **Pesquisa Social: Teoria, método e criatividade**. 2011.

MORAES, Alexandre Fernandes de. **Redes de Computadores: Fundamentos**. 2013.

MORIMOTO, Carlos E. **LDAP**. 2005. Disponível em: <http://www.hardware.com.br/termos/ldap>. Acesso em: 16 Set. 2018.

OLIVEIRA, Paulo. **Conheça o OpenLDAP: seu próximo serviço de diretórios de rede**. 2017. Disponível em: <https://www.escolalinux.com.br/blog/conheca-o-openldap-seu-proximo-servico-de-diretorios-de-rede>. Acesso em: 08 Ago. 2018.

OPENLDAP. **OpenLDAP**. 2018. Disponível em: <https://www.openldap.org/>. Acesso em: 15 Ago. 2018.

OWNCLOUD. **OwnCloud Features**. 2018. Disponível em: <https://owncloud.org/features/>. Acesso em: 10 Ago. 2018.

PALMA, Luciano; PRATES, Rubens. **Guia de Consulta Rápida: TCP/IP**. 2000. Disponível em: <http://www.martinsfontespaulista.com.br/anexos/produtos/capitulos/143086.pdf>. Acesso em: 29 Nov. 2018.

PHPLDAPADMIN. **About: What is phpLDAPadmin**. 2011. Disponível em: <http://phpldapadmin.sourceforge.net/wiki/index.php/About>. Acesso em: 20 Out. 2018.

RFC 2251. **Lightweight Directory Access Protocol**. 1997. Disponível em: <https://www.ietf.org/rfc/rfc2251.txt>. Acesso em: 23 Set. 2018.

RFC 4519. **Lightweight Directory Access Protocol (LDAP): Schema for User Applications**. 2006. Disponível em: <https://tools.ietf.org/html/rfc4519#page-5>. Acesso em: 20 Nov. 2018.

RIOS, Renan Osório. **Protocolos e Serviços de Redes: Curso Técnico em Informática**. 2011. Disponível em: <http://redeetec.mec.gov.br/images/stories/pdf/eixo_infor_comun/tec_inf/081112_protserv_redes.pdf>. Acesso em: 17 Set. 2018.

SILVA, F. H. R. **Um estudo sobre os benefícios e os riscos de segurança na utilização de Cloud Computing**. 2010.

SOUSA, Flávio R. C.; MOREIRA, Leonardo O.; MACHADO, Javam C. **Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios**. 2009.

SOUSA, Lindeberg Barros de. **Redes de Computadores: Guia Total (Tecnologia, Aplicações e Projetos em Ambiente Corporativo)**. 2010.

SCRIMGER, Rob; LASALLE, Paul; PARIHAR, Mridula. **TCP/IP: A Bíblia**. 2002.

VENEZUELA, Sandro. **O que é o OwnCloud**. 2014. Disponível em: <https://www.inovatize.com.br/site/o-que-e-o-owncloud/>. Acesso em: 14 Set. 2018.

VIANA, Gabriela. **O que é um Host?** 2012. Disponível em: <https://www.techtudo.com.br/artigos/noticia/2012/02/o-que-e-um-host.html>. Acesso em: 10 Nov. 2018.

ZANUTTO, Bruno G. **Segurança em Cloud Computing**, 2017, Disponível em: <https://dcomp.sor.ufscar.br/verdi/topicosCloud/Artigo-Seguranca-Cloud.pdf>. Acesso em: 25 Ago. 2018.

Raspberry + Samba: Um servidor para compartilhamento de arquivos e ponto de acesso

Joyce de Sant'Ana Silvano[1], William Tiago da Silva Vargas[1], Guilherme Klein da Silva Bitencourt[2], Jéferson Mendonça de Limas[2]

[1]Acadêmicos do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88960000 – Sombrio – SC – Brasil

[2]Docentes do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88960000 – Sombrio – SC – Brasil

{js1842, w_tiago_silva}@hotmail.com,
{guilherme.bitencourt, jeferson.limas}@ifc.edu.br

Abstract: *The use of computing resources and the advancement of technologies, allows a data sharing service. It becomes essential for the improvement of information management. This work aims to implement a Samba file server, using Raspberry Pi 3 hardware, along with the phpLDAPadmin tool, serving as a file and user controller, as well as providing a wireless access point. For this work, it was used the technological and applied research, besides practical tests, to analyze the performance of the proposed server. Finally, it is concluded that the expected results have been achieved and it is possible to use Raspberry Pi 3 as a low-cost Samba server to aid in data and user management in both enterprise and residential applications.*

Keywords: *File Server, Data sharing, Raspberry Pi, Samba, Access Point.*

Resumo: *A utilização de recursos computacionais e o avanço das tecnologias permite que, um serviço de compartilhamento de dados, torne- se essencial para a melhoria do gerenciamento de informações. Este trabalho tem o objetivo de implementar um servidor de arquivos Samba, utilizando um hardware Raspberry Pi 3, em conjunto com a ferramenta phpLDAPadmin, servindo como controlador de arquivos e usuários, além de fornecer um ponto de acesso sem fio. Na elaboração desse trabalho, foi utilizada a pesquisa tecnológica e aplicada, além de testes práticos, para analisar o desempenho do servidor proposto. Por fim, conclui-se que os resultados esperados foram alcançados, sendo possível utilizar o Raspberry Pi 3 como um servidor Samba, de baixo custo, para auxiliar na gerência de dados e usuários, tanto em aplicações empresariais quanto residenciais.*

Palavras-chave: *Servidor de arquivos, Compartilhamento de dados, Raspberry Pi, Samba, Ponto de Acesso.*

1 Introdução

No momento atual, a Tecnologia da Informação (TI) encontra-se presente em diferentes áreas, o que a torna algo indispensável em qualquer organização, utilizando de todos os recursos, tais como confiabilidade e facilidade de armazenamento, para seu melhor funcionamento (SILVA, 2010). Busca, também, uma forma segura e estável de compartilhar arquivos, sem o uso de serviços de terceiros ou investimento em uma grande

infraestrutura para o armazenamento destes dados, em uma rede, seja doméstica ou empresarial de pequeno, médio ou grande porte.

Nesse cenário, a Tecnologia da Informaçao visa cada vez mais ter o controle dos dados de usuários, seus serviços e seus acessos. Sabendo-se disso, é necessário a busca de uma solução para ter o próprio controle destes dados, mantendo monitoramento e gerenciamento destes serviços e contas.

O gerenciamento dos serviços visa distribuir e gerenciar os recursos de forma integrada, evitando problemas na operação dos serviços de TI. Assim, um bom gerenciamento e compartilhamento de arquivos possibilita administrar e distribuir as informações, buscando soluções derivadas pela área, unindo Tecnologia e Informação (MAGALHÃES, 2007).

Para Ross (2008), o servidor de arquivos é responsável pelo armazenamento de arquivos, que necessitam ser compartilhados com os usuários de rede, para o propósito de armazenamento e gerenciamento dos arquivos e informações, onde estão disponíveis o Samba e o LDAP – *Lightwight Directory Access Protocol*.

O Samba é um servidor e um conjunto de ferramentas que permite que máquinas Linux e Windows comuniquem-se entre si, compartilhando arquivos, diretórios, por meio do protocolo SMB/CIFS. Com o servidor Samba, é possível realizar o controle de acesso do usuário, compartilhamento de arquivos, entre outros serviços (BOAS; MENDONÇA, 2006).

O LDAP é um protocolo utilizado para autenticação e armazenamento de informações sobre usuários e grupos. É um protocolo executado no topo do protocolo TCP/IP, permitindo ao administrador executar diversas operações (LDAP, 2018).

Com o avanço das tecnologias, há vários equipamentos de hardware que cumprem com as necessidades necessárias para elaboração do projeto e nosso objetivo é oferecer um projeto com bom desempenho e de menor investimento sendo que entre as tecnologias disponíveis, foi utilizado o hardware Raspberry Pi, com hardware e software suficientes para cumprir os requisitos deste projeto.

De acordo com Halfacree e Upton (2013), o Raspberry Pi é uma placa de circuito, ou seja, um minicomputador do tamanho de um cartão de crédito, desenvolvida em 2016 no Reino Unido pela Raspberry Pi Fundation, com o intuito de auxiliar na aprendizagem da linguagem de programação.

O objetivo principal deste projeto é utilizar o hardware Raspberry Pi 3, implementando um servidor de arquivos Samba, com o propósito de compartilhar arquivos entre usuários, juntamente com a disponibilização de um ponto de acesso sem fio. Partindo deste objetivo, será implementado uma interface gráfica WEB, para auxiliar o gerenciamento do servidor de arquivos; implantar um servidor com melhor desempenho e de menor custo possível ao administrador; e utilizar um ponto *Wirelless*, a fim de facilitar o acesso da rede aos usuários.

2 Referencial Teórico

Para auxiliar na compreensão do cenário proposto, que inclui o gerenciamento de um servidor de arquivos, além de um ponto de acesso, as seções 2.1, 2.2 e 2.3 apresentam os conceitos, as ferramentas e os protocolos utilizados neste estudo. A seção 2.4 discorre sobre o Raspberry Pi. Por fim, a seção 2.5 exprime o funcionamento do ponto de acesso.

2.1 Sistema e Servidor de Arquivos

Um arquivo é um conjunto de bytes que pode conter um programa executável, um código fonte, uma planilha, um texto ou um conjunto de arquivos compactados (CUNHA, 2017).

De acordo com Maziero (2017), um sistema de arquivos pode ser visto como uma ampla estrutura de dados armazenados em um dispositivo. Na implementação de um sistema de arquivos, calcula-se que cada arquivo possui dados e metadados. Os dados de um arquivo são as informações, seu conteúdo em si, como um texto ou uma planilha, enquanto os metadados são seus atributos, como nome, data, permissão e todas as informações de controle necessárias.

Segundo Bruschi (2016), um servidor de arquivos tem por objetivo armazenar e disponibilizar informações para usuários de rede. Este serviço fornece uma melhor agilidade no processo de compartilhamento de dados, podendo ser acessado de forma remota, por um computador que possua acesso à internet.

Atualmente, existem vários tipos de servidores de arquivos. Nesse projeto será implementado o servidor Samba.

2.2 Samba

O compartilhamento de arquivo é a aplicação fundamental de redes departamentais. Servidores Linux são excelentes plataformas para servidores de arquivos, pois são rápidos e estáveis, além de fornecer escolhas de serviços de arquivos que outros sistemas operacionais não possuem (HUNT, 2004).

Os serviços Samba estão implementados com dois Daemons, sendo ele o SMB (SMBD - *Server Message Block Deamons*), que fornece serviços de compartilhamento de arquivos e de impressão e o NetBIOS Name Server (NMBD - *NetBIOS Message Block Daemon),* que fornece serviço de nome NetBIOS para endereço IP (HUNT, 2004).

O *Server Message Block – SMB*, é um protocolo para compartilhamento de arquivos, portas seriais e abstrações de comunicações. O SMB é um servidor cliente, um protocolo de solicitação-resposta (SAMBA, 2002).

A função do Samba é executar os quatro serviços CIFS - *Common Internet File System*, que são: serviços de arquivo e impressão, autenticação e autorização, resolução de nomes e anúncio de serviço (SAMBA, 2001). O CIFS é um sistema de arquivos de rede e um conjunto de serviços auxiliares suportados por vários protocolos (HERTEL, 2003).

O samba consiste em dois programas principais, que são o SMBD e o NMBD. O SMBD fornece os serviços de arquivo e

impressão, que é a base da suíte CIFS, e também autenticação e autorização do modo de compartilhamento e usuários, ou seja, que preserva os serviços compartilhados solicitando senhas. Já os serviços CIFS de resolução de nomes e navegação, são fornecidas pela NMBD, envolvendo o gerenciamento e a distribuição de listas de nomes NetBIOS (SAMBA, 2001).

2.3 LDAP

O LDAP foi criado por Tim Howes, Steve Kille e Wengyik Yeong, sendo definido pelo IETF na RFC2251. Este protocolo é usado em servidores Proxy para gerenciar o acesso dos usuários da rede e a pesquisa de informações (SOUSA, 2017).

O LDAP é um protocolo de acesso a diretório que é executado em TCP/IP. As entradas de informações no diretório LDAP são organizadas em uma estrutura hierárquica em forma de árvore e o serviço de diretório é baseado em um modelo cliente-servidor (OpenLDAP, 2000).

O Samba utiliza o LDAP para informações e autenticação de contas de usuários, grupos e máquinas, permitindo que um cliente LDAP consulte ou altere dados no diretório, comunicando-se com o servidor LDAP (GIL, 2012).

Um servidor LDAP tem por função analisar as informações do cliente e verificar se os dados solicitados estão armazenados no servidor, onde um ou mais servidores LDAP contém os dados que compõem a árvore do diretório. O servidor LDAP também realiza a função de permitir ou não que um

cliente consulte ou modifique os dados armazenados (GIL, 2012).

Para gerenciar registros em um servidor LDAP, como criação, modificação e exclusão, foi utilizada o phpLDAPadmin que é uma ferramenta web de administração do LDAP (PHPLDAPADMIN, 2011). A ilustração 1 apresenta o usuário logado ao servidor LDAP, além dos usuários e grupos cadastrados.

Figura 1 – Usuários e grupos cadastrados

Fonte: Os Autores, 2018.

O phpLDAPadmin foi projetado para ser utilizado por administradores LDAP, ou seja, pode ser utilizado por

administradores que gerenciam todo o banco de dados LDAP ou por administradores que somente gerenciam uma parte do banco de dados (PHPLDAPADMIN, 2011). A figura 2 demonstra os recursos disponibilizados pelo phpLDAPadmin para criação e alterações de grupos e usuários.

Figura 2 - Recursos para criação de objetos

Fonte: Os Autores, 2018.

2.4 Raspberry Pi 3

Conforme Merces (2013), o Raspberry Pi é um computador construído sobre uma placa composta por um processador, uma memória, entre outros componentes que auxiliam em seu funcionamento.

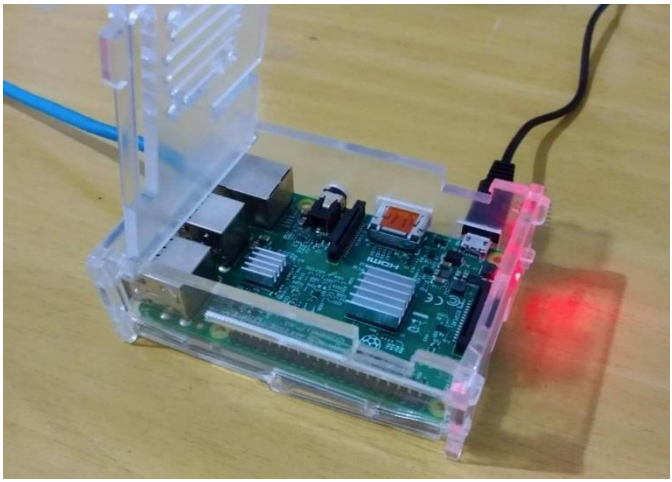
O Raspberry Pi foi desenvolvido em 2006 no Reino Unido pela Raspberry Pi Fundation, com o objetivo de estimular as pessoas a aprender linguagem de programação. Porém, somente após o ano de 2008, o objetivo foi alcançado, devido os

dispositivos móveis começarem a ficar mais acessíveis (HALFACREE; UPTON, 2013).

O Raspberry Pi possui 3 modelos, classificados entre modelo A, que foi o primeiro modelo fabricado, o modelo B e o modelo B+ produzido em 2014.

Neste projeto foi utilizado o modelo B+, que já possui o certificado Anatel, além de atualizações, como uma melhoria na utilização de energia, sendo mais econômica, facilitando em projetos que utilizem baterias. A figura 3, demonstra o Raspberry Pi utilizado, possuindo um processador Quad-Core ARM 1.2GHz de 64bits, memória de 1GB LPDDR2, um processador gráfico VídeoCore IV3D com 400MHz (RASPBERRY PI, 2018).

Figura 3 - Hardware Raspberry Pi 3



Fonte: Os Autores, 2018.

Segundo Harrington (2015), existem vários sistemas operacionais diferentes para o Raspberry Pi, incluindo o RISOS, o Pidora, o Arch Linux e o Raspbian. O sistema operacional (SO) utilizado neste projeto é o Raspbian.

De acordo com a Raspberry Pi (2018), o Raspbian já vem pré-instalado com algumas linguagens de programação, como por exemplo, o Python, Scratch, Java, entre outros.

O sistema operacional Raspbian inclui personalizações que são desenvolvidas para tornar o Raspberry Pi mais fácil de usar e inclui muitos pacotes de software prontos para uso (HARRINGTON, 2015).

2.5 Ponto de Acesso

As tecnologias sem fio (*wireless*) são cada vez mais importantes, preenchendo um espaço na vida dos usuários. Um dos pontos fracos na comunicação sem fio é a baixa segurança. Porém, com o avanço da tecnologia, é possível criar pontos de acesso ou até mesmo configurar WLANs utilizando chaves de criptografia, protegendo dados de quem acessa, como também de quem fornece a conexão. Os pontos de acesso são dispositivos de uma rede sem fio que executa a interconexão entre os dispositivos móveis. Geralmente os pontos de acesso conectam-se a uma rede cabeada, servindo como um ponto de partida (JOBSTRAIBIZER, 2010).

De acordo com Jobstraibizer (2010), estabelecimentos comerciais, como aeroportos, cafés, bibliotecas, utilizam de um ponto de acesso como parte do serviço. Para utilizar esse serviço,

o usuário deve estar no raio de cobertura do sinal do ponto de acesso. Este raio pode ser configurado no momento da distribuição do sinal, ou poderá ser limitado pela distância do ponto de acesso até o ponto em que se encontra o usuário.

3 Materiais e Métodos

Nas seções, a seguir, serão abordados os métodos e os materiais utilizados para o estudo e desenvolvimento do projeto.

3.1 Materiais

Decorrendo dos objetivos propostos no estudo, na tabela 1 consta os equipamentos utilizados durante a implementação do projeto.

Tabela 1. Equipamentos utilizados

Servidor	Raspberry Pi 3
Clientes	Notebook Windows 10 Desktop Windows 8.1 Dispositivo móvel

Fonte: Os Autores, 2018.

Utilizou-se do hardware Raspberry Pi 3 como servidor, em conjunto com a ferramenta Samba para gerenciamento de arquivos. O Raspberry Pi 3 utiliza o sistema operacional Raspbian GNU/Linux 9.4. Como máquinas clientes, utilizou-se de um Notebook com Windows 10, um Desktop com Windows 8.1 e um dispositivo móvel.

A ferramenta phpLDAPadmin tornou o servidor LDAP facilmente gerenciável de qualquer local, permitindo uma administração e navegação intuitiva. A configuração, instalação e testes foram realizadas na residência de um dos autores, sendo utilizado o notebook para acessar o Raspberry Pi.

3.2 Métodos

Neste artigo foi utilizada pesquisa tecnológica e aplicada em relação ao tema proposto, aos equipamentos e aos softwares.

Fleury (2017) define pesquisa aplicada como atividades em que conhecimentos previamente adquiridos são utilizados para coletar, selecionar e processar fatos e dados, a fim de obter e confirmar resultados.

Junior (2014) define uma pesquisa tecnológica como um estudo cientifico do artificial, onde a tecnologia pode ser vista como o campo do conhecimento relativo ao projeto de artefatos e ao planejamento de sua realização, operação, ajuste, manutenção e monitoramento, à luz do conhecimento científico.

Com base nos objetivos deste projeto e compreensão dos métodos empregados, no capitulo de resultados, serão apresentados os problemas enfrentados e as soluções adotadas.

4 Resultados e Discussões

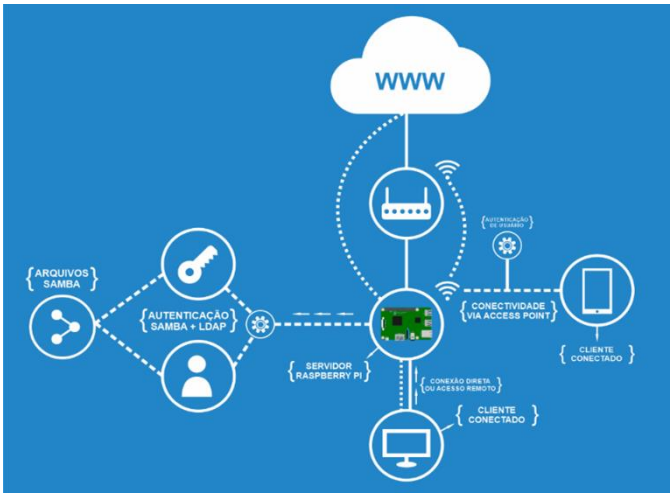
Para permitir o funcionamento do Samba com o LDAP no Raspberry Pi é necessário executar a instalação e configuração dessas ferramentas. As etapas executadas podem ser visualizadas na figura 4, separadas por seções, em conjunto

com a figura 5, que demonstra a estrutura utilizada para sua implementação.

Figura 4 - Etapas para implementação do projeto

Fonte: Os Autores, 2018.

Figura 5 - Estrutura utilizada para a implementação



Fonte: Os Autores, 2018.

A seção 4.1 apresenta a instalação do sistema operacional Raspbian no Raspberry Pi 3. Na seção 4.2 estão expostos os

procedimentos para a instalação e configuração da ferramenta Samba, seguida pela seção 4.3, que apresenta a instalação e configuração do LDAP e do phpLDAPadmin. A seção 4.4 demonstra a configuração do ponto de acesso. A seção 4.5 contém os resultados obtidos no estudo.

4.1 Instalação do Raspbian no Raspberry Pi 3

Inicialmente, a instalação do sistema operacional Raspbian foi realizada em um cartão MicroSD classe 10 de 32Gb de armazenamento, com a ajuda do Etcher, um aplicativo que grava imagens .iso em cartões SD para torná-los inicializáveis.

Com o auxílio de periféricos de entrada e saída (mouse, teclado e monitor), a configuração no Raspberry Pi 3 foi iniciada a partir da atribuição manual de um endereço IP (192.168.1.50) e da utilização do SSH como principal meio de acesso remoto através do endereço IP de 192.168.1.99. Foi utilizada a ferramenta VNC viewer, um software de compartilhamento de tela que permite conexão por SSH, ou seja, conecta-se a um computador remoto como se estivesse em frente à própria máquina.

A lista de repositórios do sistema foi atualizada utilizando o comando *apt-get update* e *apt-get upgrade*, deixando o software e suas ferramentas com as versões atuais, a fim de prevenir erros posteriores com aplicações e/ou pacotes antigos.

4.2 Instalação do Samba

Com a instalação do sistema operacional Raspbian e o acesso remoto SSH configurado, é necessário realizar a

instalação/configuração do serviço Samba, além dos respectivos backups da sua configuração padrão, decorrente dos comandos representados abaixo:

```
#sudo apt-get install samba samba-common-bin
#sudo cp /etc/samba/smb.conf  /etc/samba/smb.conf.old
```

Com o backup realizado, é necessário a configuração do arquivo *smb.conf*, para criação das pastas de compartilhamento com suas respectivas permissões de acesso. O acesso a esse arquivo foi realizado através do comando *sudo nano /etc/samba/smb.conf.* A primeira pasta a ser atribuída ao arquivo é o suporte, com o intuito de gerenciar os arquivos disponibilizados para o compartilhamento, seguido pela pasta de usuário e finalizado por uma pasta denominada total, correspondente aos administradores. As configurações executadas no arquivo *smb.conf* estão representados abaixo:

```
[global]
#definindo o grupo ou domínio a qual a máquina pertence
workgroup = raspberry
#nome que será exibido no ambiente de rede do windows
server string = servidor de arquivos
#forma de autenticação
#share (qualquer um), user (login e senha)
security = user
#local de armazenamento dos perfils
#onde %L (nome netbios do servidor) e %U (usuário)
logon path = \\%L\profiles\%U
#ativando a criptografia de senhas
encrypt password = yes
```

```
#permitindo acesso de maquinas ao domínio #host allow =
ip_da_maquina (ex.: 192.168.1.50)
#fazendo com que o samba carregue as impressoras
    Load printers = yes
#definindo o tipo de servidor de impressão
    printing = cups
#definindo local de arquivo de log do samba
    log file = /var/log/samba/samba.%
#definindo as redes que podem logar no samba
    interfaces = 192.168.1.0
#tornando o samba um controlador de domínio
#local master = yes
    #domain master = yes
    #preferred master = yes #
    domain logons = yes

#:::::::::::::: PASTAS COMPARTILHADAS ::::::::::::::::::
    [suporte]
    comment = arquivos de
     suporte path =
    /home/arquivos/suporte
    create mode = 0666
    directory mode = 0755
    write list =
     @suporte,@total read
    list = @suporte,@total
    valid users = @suporte,@total
    browseable = yes
    writeable = yes

    [usuario]
```

```
    comment = arquivos de
    usuario path =
    /home/arquivos/usuariocre
    ate mode = 0777 directory
    mode = 0444 writable =
    true security = user
    browseable = true
    write list =
    @usuario,@total read list =
    @usuario,@total valid
    users = @usuario,@total
    public = yes


    [total]
    comment = arquivos de administradores
    path = /home/arquivos/total
    create mode =
    0666 directory
    mode = 0777
    write list = @total
    read list = @total
    valid users = @total
    browseable = yes writeable = yes
    writeable = yes

#para isso não se tornar "duplicado", devido ter que,
primeiramente, cadastrar um usuário no Linux e depois
associá-lo no samba
    unix password sync = Yes
```

```
    passwd program = /usr/bin/passwd %u
    passwd chat =
    *New*UNIX*password* %n\n
    *Retype*new*UNIX*password* %n\n
     *passwd:*all*authentication*tokens*updated*
      successfully*
   #:::::::::::::::::::: LDAP ::::::::::::::::::::::::
include                 /etc/ldap/schema/core.schema include
                        /etc/ldap/schema/cosine.schema
include                 /etc/ldap/schema/nis.schema include
                        /etc/ldap/schema/inetorgperson.
schema include          /etc/ldap/schema/samba.schema


passdb backend =
ldapsam:"ldap://localhost.com/phpldapadmin/"

ldap ssl = off
ldap admin dn = cn=admin,dc=pi3,dc=com ldap suffix =
dc=pi3,dc=com
```

Finalizada sua configuração, foi reinicializado o serviço com o comando:

```
#sudo /etc/init.d/samba restart
```

4.3 Instalação e configuração do LDAP e do PHPLDAPADMIN

Assim como no Samba, durante a instalação do LDAP e do PhpLDAPadmin, é necessário instalar dependências essenciais

para seu funcionamento. Estas dependências e a ferramenta Vim, para edição de arquivos, foram baixados através do comando:

```
#sudo apt install –y php ph-xml php-ldap php-common php7.0- ldap php7.0-common php7.0 php7.0-fpm php7.0-cgi php7.0-cli php7.0-json php7.0-opcache php7.0-readline

  #sudo apt install –y vim
```

Finalizada a instalação das dependências, foi iniciada a instalação do LDAP. Na etapa de configuração de acesso, foi atribuída a senha "*admin"*.

A instalação do LDAP foi realizada pelo comando *sudo apt –y install slapd ldap-utils.* O primeiro arquivo a ser editado após o fim do processo de instalação foi o arquivo de configuração *ldap.conf,* localizado no caminho */etc/ldap*. No arquivo são alteradas as linhas de configuração de autenticação para acesso ao phpLDAPadmin, sendo a primeira linha a base de autenticação e a segunda linha, o endereço de acesso à interface web, conforme demonstrados a seguir:

```
BASE dc=pi3,dc=com

URI ldap://localhost:389
```

Terminada a instalação do LDAP, é feita a configuração de acesso, através do comando *sudo dpkg-reconfigure slapd,* marcando as sete (7) opções representadas abaixo:

```
#if you enable      no
#DNS domain name  pi3.com
#organization name   raspberry
#admin password   admin
```

```
#database backend   HDB
#do you want the database to be removed   no
#move old database?   Yes
```

Com a instalação do LDAP concluída, deu-se início a instalação do serviço phpLDAPadmin versão 1.2.2-6.1, que fará a gerência dos arquivos, grupos e usuários do servidor Samba, através dos comandos representados:

```
#wget
http://ftp.br.debian.org/debian/pool/main/p/phpldapadmin/p
hpldapad min_1.2.2-6.1_all.deb

#sudo apt install –y ./phpLDAPadmin_1.2.2-6.1_all.deb
```

Com o processo de instalação executado, é editado o arquivo de configuração *config.php,* com o acesso realizado pelo comando *sudo vim /etc/phpLDAPadmin/config.php.* As linhas de comando alteradas são mostradas abaixo:

```
Linha original:
$servers
setValue(‘server’,’base’,array(‘dc=example,dc=com’));
Linha alterada:
$servers
setValue(‘server’,’base’,array(‘dc=pi3,dc=com’));

Linha original:
$ servers-
>setValue('server','base',array('dc=default,dc=com'));
Linha alterada:
$servers-
>setValue('server','base',array('dc=pi3,dc=com'));

Linha original:
```

```
$servers-
>setValue('login','bind_id','cn=admin,dc=default,dc=com'
);
Linha alterada:
$servers-
>setValue('login','bind_id','cn=admin,dc=pi3,dc=com');
```

Por fim, é alterada a linha de configuração *$config custom appearance ['hide_template_warning']=false,* modificando a palavra *false* para *true.*

Com as alterações finalizadas, foi instalado o lighttpd, um servidor web similar ao Apache, porém mais leve, além da criação do arquivo *.conf* para configuração, conforme demonstrado a seguir:

```
#sudo apt install –y apache2-utils lighttpd
#sudo vim /etc/lighttpd/conf-available/50-
phpldapadmin.conf
```

Dentro do arquivo *.conf* são adicionados quatro linhas de comando:

```
#alias for phpLDAPadmin directory
Alias.url += (
        "/phpLDAPadmin" => "/usr/share/phpldapadmin",
)
```

Em seguida, é habilitado o arquivo de configuração *.conf* e o arquivo *fastcgi*, através do comando:

```
#sudo In –s /etc/lighttpd/conf-available/50-
phpldapadmin.conf
```

```
/etc/lighttpd/conf-enabled/
#sudo In –s /etc/lighttpd/conf-available/10-fastcgi.conf
/etc/lighttpd/conf-enabled/
#sudo lighttpd-enable-mod fastcgi fastcgi-php
```

Após as configurações, é necessário reiniciar alguns serviços, como demonstrado abaixo:

```
#sudo service lighttpdforce-reload
#sudo service lighttpd restart
#sudo systemct start slapd
#sudo systemctlenable slapd
```

Com os serviços reiniciados, já é possível acessar, via navegador web, o serviço do phpLDAPadmin, que fará a gerência dos arquivos, grupos e usuários do serviço Samba através do endereço http://localhost/phpldapadmin.

4.4 Configuração do ponto de acesso Hotspot

Com a instalação do sistema operacional Raspbian, do serviço Samba, LDAP e phpLDAPadmin finalizadas, inicia-se a instalação do ponto de acesso. Primeiramente, atribui-se um endereço IP estático a interface *wireless*, para isso, é acessado o arquivo */etc/dhcpcd.conf* introduzindo a seguinte linha de comando:

```
#denyinterfaces wlan0
```

Realizada a alteração no arquivo *dhcpcd.conf* é atribuído o IP estático, da interface *wireless,* dentro do diretório */etc/network/ interfaces*, como demonstrado:

```
auto wlan0
iface wlan0 inet static
hotsapd
/etc/hostapd/hostapd.conf address
192.168.1.
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
```

Com o IP da interface atribuído, é realizado a instalação e configuração do *hostapd* e *dnsmasq*. O hostapd é o daemon Host Access Point, que fornece criptografia WPA2 e autenticação em pontos de acesso *wireless* baseados em Linux. Sua instalação é feita através do comando:

```
#sudo apt-get install hotsapd
```

Após concluída a instalação, é editado o arquivo de configuração *hostapd.conf*, localizado no diretório */etc/hostapd*, para criação da rede *wireless* e sua senha de acesso. O comando para acesso do arquivo e as respectivas linhas de comando podem ser visualizados abaixo:

```
# sudo vim /etc/hostapd/hostapd.conf
```

```
interface=wlan0
driver=nl80211
ssid=Rasp3
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=raspberry
wap_key_
mgmt=WPA-PSK
rsn_pairwire=CCMP
```

Concluída a modificação, é acrescentado ao arquivo */etc/default/hostapd* o caminho para localização do arquivo de configuração, através do comando:

```
#DAEMON_CONF=”/etc/hostapd/hostapd.conf”
```

Outro ponto importante durante a implementação do ponto de acesso é a instalação do *dnsmasq*, que combina funções de DHCP e DNS. A instalação e configuração do *dnsmasq* é executada pelos seguintes comandos:

```
#sudo apt-get install dnsmasq

#sudo vim /etc/dnsmasq.conf
```

```
interface=wlan0
listen-address=192.168.1.51 bind-interfaces server=8.8.8.8
domain-needed bogus-priv
dhcp-range=192.168.1.52, 192.168.1.254, 12h
```

Com o IP da interface *wireless*, o *hostapd* e o *dnsmasq* instalados e configurados, é realizada a configuração do encaminhamento dos pacotes que são encaminhados à interface *wireless* e tem como destino a interface Ethernet. Com isso, primeiramente, ativa-se o encaminhamento IP, desmarcando a linha de comando *net.ipv4.ip_forward=1*, localizado no arquivo */etc/sysctl.conf*. Por fim, para sua ativação imediata é realizado o comando:

```
#sudo sh –c “echo 1 > /proc/sys/net/ipv4/ip_forward”
```

Mesmo com todo esse processo já realizado, para transformar o Raspberry Pi num roteador, é necessário executar três regras de configuração e armazená-las num arquivo, como demonstrado a seguir:

```
#sudo iptables –t nat –A POSTROUTING –o eth0 –j MASQUERADE

#sudo iptables –A FORWARD –i eth0 –o wlan0 –m state –state RELATED, ESTABLISHED –j ACCEPT

#sudo iptables –A FORWARD –i wlan0 –o eth0 –j ACCEPT #sudo sh –c “iptables-save > /etc/iptables.ipv4.nat”
```

Após os comandos anteriores concluídos, é necessário adicionar ao final do arquivo */etc/network/interfaces* o comando abaixo, responsável por executar as regras de *iptables* sempre que o Raspberry iniciar:

```
pre-up iptables-restore < /etc/iptables.ipv.nat
```

Para concluir a instalação e configuração do ponto de acesso, inicia-se os serviços *hostapd* e *dnsmasq*, através dos comandos:

```
#sudo service hostapd start
#sudo service dnsmasq start
```

5 Resultados

Com a implementação deste dispositivo, é oferecido um servidor para o compartilhamento de arquivos entre os usuários, utilizando uma autenticação segura com *login* e senha, que faz a verificação da identidade do usuário, liberando assim o acesso aos arquivos compartilhados.

Para manter todos os usuários conectados e com acesso à internet, disponibilizou-se um ponto de acesso, com uma conexão estável e com velocidade balanceada para melhor atender quem esteja conectado ao dispositivo.

Para o gerenciamento de todos os recursos, está disponível um painel de controle dos usuários permitidos, que foram previamente cadastrados, no arquivo do servidor Samba, com a interface web do phpLDAPadmin. Essa ferramenta também oferece um monitor de recursos detalhado para melhor visualização do uso de cada processo que está executando no dispositivo.

A ideia deste estudo foi apresentar um servidor Samba com o hardware Raspberry Pi, sendo uma opção de baixo custo, quando comparado `as máquinas próprias de servidores de

arquivos. Assim, este projeto oferece um sistema de compartilhamento de arquivos e gerenciamento de usuários com bom desempenho para empresas comerciais ou para uso pessoal.

Para a realização dos testes foi utilizada a ferramenta Jmeter, que é um software de código aberto, utilizado para a execução de testes e medição de desempenho (JMETER, 2018).

Foram realizados dois testes de cargas, executados em três dispositivos diferentes, todos com um período de duração de 60 segundos. Os dispositivos utilizados foram um Desktop, um Notebook e o Raspberry Pi 3, todos funcionando como servidor de arquivos.

Nos testes foram filtrados os dados de interesse, classificados por uso de disco, uso de memória e uso de CPU. As configurações realizadas nas três máquinas podem ser visualizadas na tabela abaixo, como também, o investimento necessário para o funcionamento das máquinas. Os preços apresentados na tabela foram extraídos de sites de busca em novembro de 2018.

Tabela 2 - Configurações e investimento dos equipamentos utilizados para teste

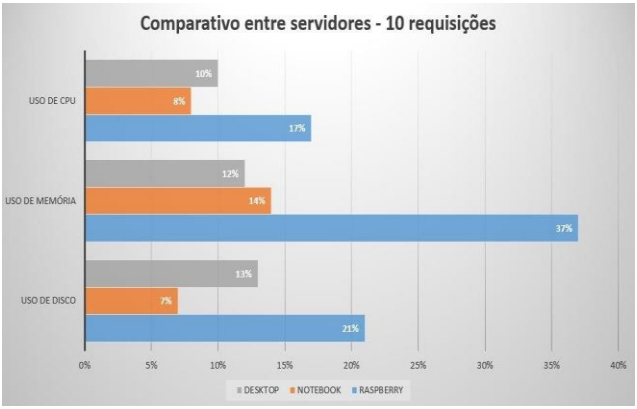
	Raspberry Pi 3	**Desktop**	**Notebook**
SO	Raspbian 9.4	Windows 8.1	Windows 10
Processador	Quad-Core ARM	Intel Core I3-2100	Intel Core I5 7200U
Mémoria	1gb LPDDR2	8gb DDR3	8gb DDR3

Disco	MicroSD 32gb Classe 10	HD Seagate 500gb 7200 rpm	HD 1TB 5400 rpm
Investimento	R$325,00	R$2748,00	R$2400,00

Fonte: Os Autores, 2018.

O primeiro teste realizado, foi a partir de 10 requisições de usuários ao servidor, como pode ser visualizado no gráfico da figura 6.

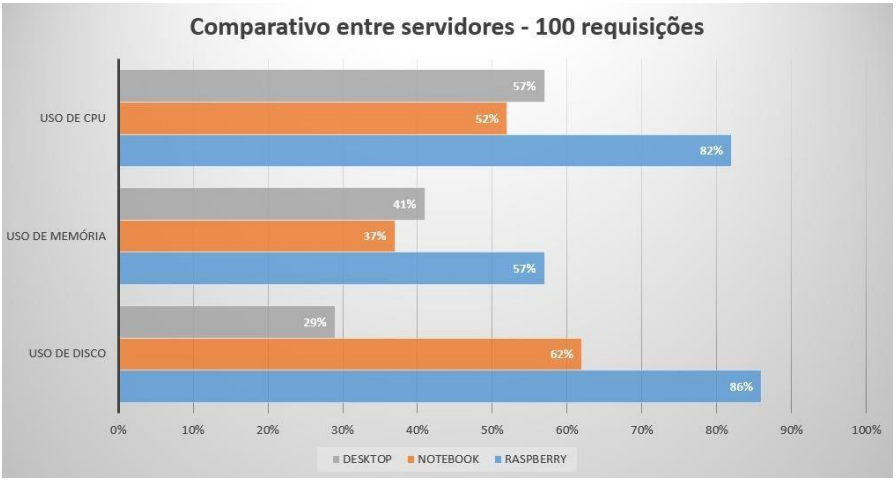
Figura 6 - Testes realizados com 10 requisições de usuários ao servidor



Fonte: Os Autores, 2018.

O segundo teste realizado, foi a partir de 100 requisições de usuários ao servidor, como pode ser visualizado no gráfico da figura 7.

Figura 7 - Testes realizados com 100 requisições de usuários ao servidor



Fonte: Os Autores, 2018.

A partir dos testes realizados, é possível visualizar a diferença entre um Desktop, um Notebook e um hardware Raspberry Pi 3. Conforme visualizado nos gráficos, nota-se que os serviços utilizam mais recursos no Raspberry Pi, devido sua limitação de hardware. Entretanto, apesar de um desempenho inferior quando comparado aos outros dois servidores, o Raspberry Pi possui um comportamento satisfatório, principalmente levando em consideração o custo reduzido para sua implementação.

Para uma empresa de pequeno e médio porte, com uma demanda de acesso razoável, o Raspberry Pi pode ser uma escolha interessante, oferecendo um custo benefício maior, se comparado a um servidor próprio para este objetivo.

Da mesma forma, para uma empresa de grande porte e com demanda de acesso elevada, pode se tornar uma opção viável, com relação ao custo financeiro da empresa em questão.

6 Considerações Finais

Com os resultados obtidos neste trabalho, percebe-se que foram alcançados os objetivos propostos inicialmente. O servidor e o ponto de acesso se apresentaram estáveis e eficientes durante o processo de implementação, como também, durante os testes executados. Desta forma, o dispositivo é uma proposta válida para empresas de pequeno e médio porte que não dispõem de um custo financeiro alto. Pode ser utilizado também por empresas de grande porte que optam por reduzir custo e que necessitam de um servidor de arquivos compartilhados, na rede local, que seja seguro, prático e portátil.

Ao utilizar o equipamento, o usuário pode encontrar dificuldades no início, entretanto, após algumas instruções e um determinado tempo de interação com a ferramenta, a tendência é que essa tarefa se torne fácil. Muitas empresas que não disponibilizam de um profissional capaz para operar na área de TI podem se beneficiar de um servidor como este, otimizando o desempenho, como também reduzindo os custos.

No decorrer do estudo, dentre as várias dificuldades encontradas, a que se destaca foi a falta de informações, tanto em relação à conexão entre o Samba/LDAP com o phpLDAPadmin, como também, para configuração e criação de processos no próprio phpLDAPadmin.

Ao longo do processo de implementação, algumas alterações foram necessárias, pois a ideia inicial seria utilizar a ferramenta Docker para portabilidade e mobilidade, porém, devido às dificuldades encontradas, não foi possível realizá-la.

Para trabalhos futuros, sugere-se a utilização da ferramenta Docker em conjunto com o Raspberry Pi, para facilitar a portabilidade e mobilidade das aplicações utilizadas neste projeto, permitindo, assim, backups armazenados em nuvem, que agilizem a restauração e otimização do tempo na resolução de eventuais problemas. Do mesmo modo, sugere-se ampliar o espaço de armazenamento do servidor, instalando um HD ou SSD, aliando capacidade e velocidade de leitura, quando comparado ao cartão SD de Classe 10 com 32Gb, que foi utilizado nesse estudo.

Referências

BRUSCHI, G.; FERNANDES, V.; LOUZANO, V. **Teste de Desempenho de um Servidor de Arquivos Samba, Utilizando Raspberry Pi com Diferentes Dispositivos de Armazenamento**. SP: Fatec Bauru, 2016. Disponível em: <http://www.fatecbauru.edu.br/ojs/index.php/CET/article/view/208/179>. Acesso em 18 jun. 2018.

CUNHA, G.; MACEDO, R.; SILVEIRA, S. **Informática Básica.**Santa Maria, RS: UFSM, NTE, 2017.

FLEURY, M.; WERLANG, S. **Pesquisa Aplicada – Reflexões sobre conceitos e abordagens metodológicas.** FGV, 2017. Disponivel em: <https://bibliotecadigital.fgv.br/dspace/bitstream/

handle/10438/18700/A_pesquisa_aplicada_conceito_e_abordagens_metodol%C3%B3gicas.pdf?sequence=6&isAllowed=y>. Acesso em 24 jun. 2018.

GIL, A. **OpenLDAP Extreme**. Rio de Janeiro: Brasport, 2012.

HARRINGTON, W. **Learning Raspbian.** Birmingham**:** Packt Publishing LTDA, 2015. Disponível em: <https://tentacle.net/~prophet/raspberrypi/Raspberry%20Pi/1784392197%20%7B8127D82E%7D%20Learning%20Raspbian%20%5BHarrington%202015-02-27%5D.pdf>. Acesso em 24 jun 2018.

HERTEL, C. **Implementing CIFS: The Common Internet File System**. Editora Prentice Hall. 2003.

HUNT, C. **Linux: servidores de rede**. Rio de Janeiro: Ciência Moderna, 2004.

JMETER. **Apache Jmeter.** 2018. Disponível em: <https://jmeter.apache.org/>. Acesso em 03 dez 2018.

JOBSTRAIBIZER, F. **Desvendando as Redes Sem Fio**. São Paulo: Digerati Books, 2010.

JUNIOR, V. **A Pesquisa Científica e Tecnológica**. 2014. Disponível em: <http://www.revistaespacios.com/a14v35n09/14350913.html>. Acesso em 16 dez 2018.

LDAP. **Sobre o LDAP.** 2018. Disponivel em: <https://ldap.com/about/>. Acesso em 17 set 2018.

MAGALHÃES, I.; PINHEIRO, W. **Gerenciamento de Serviços de TI na Prática.** São Paulo: Novatec, 2007.

MAZIERO, C. **Sistemas Operacionais: Conceitos e Mecanismos.** DINF–UFPR, 2017. Disponivel em:

<http://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=so:so-livro.pdf>. Acesso em 23 jul 2018.

MERCES, R. **Raspberry Pi: Conceito e Prática**. RJ: Ciência Moderna, 2013.

OPENLDAP. **O que é o LDAP**. 2000. Disponivel em:<https://www.openldap.org/doc/admin20/intro.html#What%20is%20 LDAP>. Acesso em 04 ago 2018.

PHPLDAPADMIN. **O que é phpLDAPadmin**. 2011. Disponivel em:<http://phpldapadmin.sourceforge.net/wiki/index.php/About>. Acesso em 12 ago 2018.

RASPBERRY PI. **Rapberry Pi 3 Model B+.** 2018. Disponivel em:<https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>. Acesso em 07 jun 2018.

RASPBERRY PI. **Raspbian**. 2018. Disponivel em: <https://www. raspberrypi.org/downloads/raspbian/>. Acesso em 17 mai 2018.

ROSS, J. **Redes de Computadores**. Editora Livro Tec, 2008. SAMBA.

SAMBA. **Samba: uma introdução.** 2001. Disponivel em: <https://www.samba. org/samba/docs/SambaIntro.html>. Acesso em 23 jul 2018.

SAMBA. **Apenas, o que é SMB?.** 2002. Disponivel em: <https:// www.samba.org/cifs/docs/what-is-smb.html#What_Is_ SMB>. Acesso em 04 ago 2018.

SILVA, C. **Monitoramento de Redes Windows com servidor Linux utilizando o Samba**. - Curso de Sistemas de

Informação, Universi- dade do Planalto Catarinense, Lages, 2010.

SOUSA, L. **Gerenciamento e Segurança de Redes**. SP: Senai, 2017.

UPTON, E.; HALFACREE, G. **Raspberry PI: Manual do Usuário**. SP: Novatec, 2013.

Rastreamento e monitoramento veicular utilizando Raspberry Pi.

Giuvan Santos Rodrigues[1], Marco Antonio Silveira de Souza[2]

[1]Acadêmico do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88960-000 – Sombrio – SC – Brasil

[2]Docente do Instituto Federal Catarinense – *Campus* Avançado Sombrio – 88960-000 – Sombrio – SC – Brasil

giuvan.rodrigues@gmail.com[1],

marco.souza@ifc.edu.br[2]

Abstract. *The demand for cargo and fleet tracking and monitoring systems has been growing substantially and making them common use by transport companies because it allows the exact location of their vehicles. This article has as main objective to present the tools that were used to develop a monitoring system in real time using Raspberry Pi. In order to reach the proposed objective, an applied research was carried out by means of a case study for a company from Extremo Sul Catarinense, using a route that takes students from the city of Sombrio / SC to UNESC (Criciúma / SC) to carry out the tests. The operation of the application is demonstrated through that scenario.*

Resumo*. A procura por sistemas de rastreamento e monitoramento de cargas e frotas de transporte vem crescendo substancialmente e tornando comum a sua utilização pelas empresas de transporte por possibilitar a localização exata de seus veículos. Este*

artigo tem como principal objetivo apresentar as ferramentas que foram utilizadas para desenvolver um sistema de monitoramento em tempo real utilizando Raspberry Pi. A fim de alcançar o objetivo proposto, foi realizada uma pesquisa aplicada por meio de estudo de caso para uma empresa do Extremo Sul Catarinense, utilizando uma rota que leva estudantes da cidade de Sombrio/SC para a UNESC (Universidade do Extremo Sul Catarinense) (Criciúma/SC) para a realização dos testes, a fim de que, através deste cenário, seja demonstrado o funcionamento da aplicação.

1 Introdução

O monitoramento e rastreamento veicular é uma prática que permite acompanhar a localização de um determinado veículo através de um sistema que recebe e transmite dados via satélite, e com grande proporção do mercado, cada vez mais procura por empresas para gestão de frotas, trazendo assim benefícios como: segurança, controle e monitoramento em tempo real.

Segundo Mantuano (2018), o rastreamento veicular além de proporcionar segurança para o veículo e para quem estiver a bordo, pode oferecer informações em tempo real para que medidas preventivas possam evitar prejuízos, já que qualidade e produtividade são importantes para atingir a excelência em um serviço prestado (MISTRETTA; DELMANTO JUNIOR., 2012).

Há diversos sistemas de rastreamento disponíveis no mercado, onde o mais comum é contratar um serviço com equipamento privado e pagar mensalidade, ou optar por adquirir sistemas e aparelhos com preço inferior, porém sem oferecer suporte. Diante dessa realidade, uma empresa de fretamento encara a seguinte questão: é possível contratar ou obter um

serviço confiável, com disponibilidade e baixo custo? No cenário atual de tecnologia há uma imensa disponibilidade de equipamentos e soluções que podem ser usados para abordar este problema. Uma delas é a utilização de plataformas embarcada como RBP, com uma relação custo benefício excelente para projetos de prototipagem. O RBP é um dispositivo de baixo custo, porém, com poder de processamento suficiente para atender uma vasta gama de aplicações práticas.

Assim, este trabalho tem por objetivo demonstrar o uso de ferramentas *Open Source* que podem ser configuradas para atender a necessidade de uma empresa de fretamento de ônibus e, através disso, desenvolver um sistema utilizando um hardware de baixo custo, e um aplicativo para ser utilizado em conjunto com um dispositivo (computador, *smartphone*, etc) para monitorar e gerenciar a sua frota. Para o alcance deste objetivo delineiam-se os seguintes objetivos específicos:

- Pesquisar a funcionalidade do Raspberry Pi;
- Relatar as ferramentas disponíveis para a criação e desenvolvimento de aplicações, através da IBM Cloud;
- Desenvolver e avaliar um protótipo de rastreamento com base em uma rota específica.
- Apresentar o resultado obtido.

Levou-se a termo uma pesquisa do tipo aplicada, com estudo de caso aplicado a uma empresa do Extremo Sul Catarinense que trabalha no ramo de fretamento e busca por uma tecnologia que ofereça maior segurança para seus colaboradores e clientes, com um baixo investimento.

Este trabalho está organizado da seguinte forma: a primeira seção apresenta uma introdução do tema e seus objetivos, na segunda seção a fundamentação teórica utilizada na elaboração deste artigo. As subseções, por sua vez, abordam

conceitos e significados das tecnologias utilizadas. A terceira seção aborda os materiais e métodos utilizados para a elaboração do trabalho, uma apresentação sucinta da empresa que o estudo de caso foi aplicado e o modelo proposto para elaboração do projeto. Na sequência, outras questões como forma de implementação, ambiente para os testes, os resultados obtidos, pontos positivos e negativos são analisados e por último as considerações finais, trazendo uma visão geral do protótipo e seu funcionamento.

2 Referencial Teórico

Nesta seção são apresentados temas como: conceito de rastreamento e monitoramento, o funcionamento do sistema global de navegação geográfica, o conceito de rede móvel, a apresentação do micro computador Raspberry Pi e uma breve apresentação da plataforma Android.

2.1 Rastreamento e Monitoramento

Rastreamento no ambiente veicular é um sistema de monitoramento para poder localizar um veículo durante sua movimentação (ASSIS, 2010). Segundo Branco (2009), o rastreamento chegou ao Brasil em 1994, motivado por questões de segurança.

Aguilera *apud* Prado *et al.* (2010) elaboraram um estudo em uma grande empresa prestadora de serviços e chegaram à conclusão que o principal fator para empresas adotarem monitoramento e rastreamento é a prevenção ao roubo de veículos e cargas. Os autores consideram que no Brasil a evolução constante de sistemas de rastreamento desenvolveu-se principalmente por possuir um alto índice de roubo. Os sistemas integrados vêm sendo utilizados há bastante tempo para diminuir custos de seguros, roubos de cargas e dar segurança para transportadores (LUIZ, 2008).

Atualmente, as funcionalidades destes sistemas têm sido ampliadas e tornaram-se cada vez mais precisas e através destes sistemas, dados podem ser coletados e permitem a realização de operações de controle logístico, controle de risco, gerenciamento de frotas e gestão de transporte (ASSIS, 2010).

Monitoramento significa coletar e analisar um conjunto de dados e informações, e com resultado pode-se chegar a uma solução, ou evitar alguma ação, ou até mesmo buscar melhoras no desempenho de determinada função (CÂNDIDO *et al.*, 2015).

De acordo com Cândido *et al.* (2015), o processo de monitoramento auxilia a empresa a melhorar sua habilidade de organizar, resolver problemas e de atender as necessidades de seus clientes e usuários, fornecendo assim, um serviço de qualidade superior. Cândido *et al.* (2015) afirma que a Tecnologia da Informação, junto com mecanismos e métodos adequados, são indispensáveis para garantir qualidade e pontualidade nesse processo para auxiliar tomadas de decisões através de sistemas.

As informações que serão monitoradas neste artigo são coordenadas geográficas. Segundo Carvalho e Araújo (2008),o sistema de mapeamento da terra através de coordenadas geográficas expressa qualquer posição no planeta através de duas coordenadas.

Campos (2012) define coordenadas geográficas como um conjunto de paralelos e meridianos, que servem de referência para qualquer localização, mais conhecidas como latitude e longitude. De acordo com Carvalho e Araújo. (2008) latitude é a distância em graus de qualquer ponto da terra até a Linha do Equador. Longitude é a distância em graus de qualquer ponto na terra até o Meridiano de Greenwich. Através das coordenadas

geográficas e com um sistema de navegação por satélite podemos localizar qualquer ponto na terra.

2.2 Sistemas Global de Navegação por Satélite

A sigla GNSS vem de *Global Navigation Satellite System*, ou seja, Sistema Global de Navegação por Satélite. É um sistema de navegação e posicionamento. Conhecido popularmente como GPS, os principais sistemas globais que compõem o GNSS são: o Russo (GLONASS) o Europeu (GALILEO) e o Chinês (BEIDOU).

Esta rede de satélites é a responsável por enviar sinais de rádio que possibilitam a realização desta orientação através de coordenadas geográficas e cálculos de distância para localizar determinado objeto. No início das operações, apenas o GPS figurava no cenário dos GNSS, e é o termo mais popularizado, utilizado e disponibilizado de forma gratuita em nosso país (ASSIS, 2010). Os GNSS foram criados inicialmente para serem usados para fins militares, além de serem disponíveis para o uso gratuito de todos ainda é financiado pelas forças aéreas americanas.

O GPS permite que o usuário localize o posicionamento de seu objeto através de coordenadas geográficas (latitude, longitude) (ASSIS, 2010). O satélite fica enviando sinais de rádio constantemente para que seus receptores captem. Para obter mais precisão, os satélites são equipados com relógios atômicos, com um par de relógio de rubídio e/ou de césio[30], com precisão de microssegundos e seus horários são sincronizados.

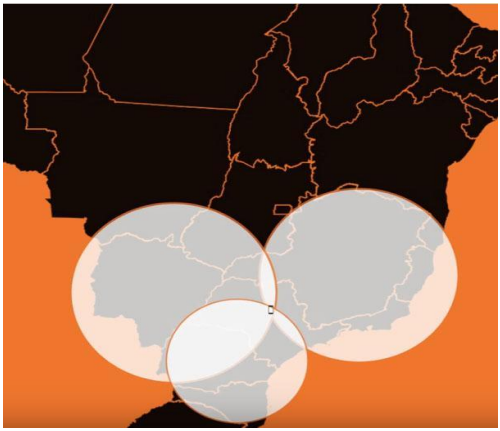
No mínimo, três satélites têm que enviar sinais para que o receptor possa calcular em que distância e hora cada satélite se encontra, e com base nessas informações é determinado a longitude e a latitude (HASEGAWA *et al.*, 1999). Essa operação

[30]São relógios de alta precisão controlados por mecanismos atômicos.

é baseada em um princípio matemático chamado trilateração (ARVUS *apud* GALON, 2014, p. 1).

A Figura 1 exemplifica a analogia do princípio matemático de trilateração em 2D (longitude e latitude).

Figura 1 - Princípio matemático.



Fonte: Poeira cósmica, 2016.

O protocolo NMEA 0183 é utilizado para a comunicação dessas informações. Ele foi criado pela National Marine Electronics Association, e define requisitos de sinal elétrico, protocolo e tempo de transmissão de dados e formatos de frases específicos, destinado a suportar transmissão de dados em série unidirecional de um único locutor para vários ouvintes, esses dados são mensagens no formato ASCII que incluem posição, velocidade, profundidade, alocação de frequência. (NATIONAL MARINE ELECTRONICS ASSOCIATION, 2008).

Esses dados são transmitidos através de sentenças. Neste artigo será trabalhado com o padrão de mensagem $GPGGA, que decodifica essas strings NMEA.

2.3 Rede Móvel

De acordo com Galon (2014), as telecomunicações surgiram com sistema de mensagens telégrafo, enviando sinais elétricos codificados, e posteriormente, aperfeiçoou-se conseguindo transmitir sons pelo telégrafo falante, mais tarde sendo conhecido por telefone (SILVA, 2010 *apud* GALON, 2014).

Os serviços oferecidos, na primeira geração, eram serviços simples, basicamente serviços de voz. Os primeiros equipamentos funcionavam com recursos de redes 1G, que era uma tecnologia analógica que apenas permitia a comunicação por voz, não suportando a transmissão de dados. Na década de 80 acontece a evolução para o sistema digital, possibilitando tráfego de dados e recursos multimídia ficaram conhecidos como 2G e 3G. (GALON, 2014).

Atualmente, a rede móvel mais usada é conhecida comercialmente como 4G, embora LTE (*Long Term Evolution*) seria o termo técnico adequado. Ela traz uma evolução considerável nas taxas de transmissão em relação ao 3G e está em constante aprimoramento. O LTE *Advanced* e o LTA *Advanced Pro*, conhecidos por 4G+ e 4,5G respectivamente, são a evolução desta tecnologia.

Através da rede móvel disponibilizada por um roteador 4G, o dispositivo Raspberry Pi 3 se mantém conectado com a internet, conservando a comunicação com o servidor.

2.4 Raspberry Pi

Os sistemas embarcados são uma tendência tecnológica, e estão cada vez mais complexos e poderosos, com maior capacidade de processamento (VARGAS, 2018). De acordo com Pereira *et al.* (2011), os sistemas embarcados possuem como potencial uma flexibilidade para executar determinadas funções que podem ser pré-programadas para atuar em sistemas maiores.

Uma definição para sistema embarcado é colocar uma capacidade computacional dentro de um circuito integrado, que tem características de independência de operação, e possibilita que usuários possam interagir através de interfaces (CUNHA, 2015).

De um modo geral, os sistemas embarcados realizam um conjunto de tarefas pré-definidas com objetivos e tarefas específicas. Existem diversas arquiteturas disponíveis no mercado, nesse artigo foi escolhido trabalhar com o Raspberry Pi 3 modelo B, por possuir recursos tecnológicos agregados, conta com processador quad-core Broadcom BCM2837 de 1,2 GHz (Cortex-A53), 1 GB de RAM, Wi-Fi 802.11n, saída de vídeo HDMI, quatro portas USB 2.0, entrada para cartão de memória e conexão Fast Ethernet, pesando 45 gramas, que o possibilitam otimizar seu tamanho e o custo do produto equivalente a 114,00 Reais.

Segundo Hein (2013), o engenheiro britânico Eben Upton e uma equipe de hackers de hardware começaram o projeto Raspberry Pi como um meio de fornecer tecnologia de computador acessível para jovens interessados no assunto. O objetivo era desenvolver e comercializar um computador de placa única, do tamanho aproximado de um cartão de crédito e compatível com o orçamento apertado do público alvo.

O Raspberry Pi considera como sistema operacional padrão o Raspbian, que é uma distribuição Linux baseada em Debian, otimizado para o hardware do Raspberry Pi e conta com programas básicos e utilitários como ferramentas de desenvolvimento (RASPBIAN ORGANIZATION, 2018).

2.5 Plataforma Android

O Android é um Sistema Operacional da Google e é uma plataforma móvel baseada em Linux, composta por um ambiente de desenvolvimento de software.

Devido à evolução de recursos de *smartphones* e *tablets*, surgiu a programação de aplicativos para que o usuário pudesse interagir com seu dispositivo móvel através de interface gráfica. Uma das principais vantagens dos dispositivos com a tecnologia Android é a interação com os serviços Google, como por exemplo Google Maps. (GALON, 2014). O Android Studio é o ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de aplicativos Android.

3 Materiais e métodos

A pesquisa aplicada foi do tipo estudo de caso, aplicado à empresa Eusantur Viagens, que foi fundada no ano de 2010 e está localizada na cidade de Sombrio/SC e mantêm o transporte universitário através de veículos próprios e conveniados.

Uma pesquisa aplicada tem como objetivo atuar em torno de problemas presentes nas atividades de empresas, organizações ou instituições. A pesquisa concentrada na elaboração de diagnósticos, identificação de problemas e busca soluções (FLEURY e WERLANG, 2010).

Esse tipo de pesquisa utiliza de toda informação disponível para a criação de novas tecnologias, possuindo resultados palpáveis e percebidos pela população (CERVI, 2018).

Para a elaboração desse artigo apresentam-se os materiais e métodos empregados com o objetivo de desenvolver uma aplicação e um sistema de rastreamento e monitoramento, de baixo custo e confiável para a empresa mencionada.

Inicialmente a fundamentação teórica dos principais tópicos que foram utilizados no decorrer do trabalho através de uma pesquisa bibliográfica, trazendo conceitos e explicações.

De acordo com Fleury e Werlang (2010), os objetivos de uma pesquisa podem ser diversos: criar uma visão geral de uma determinada condição, gerar novas ideias e conhecer os fatos básicos que circundam uma situação.

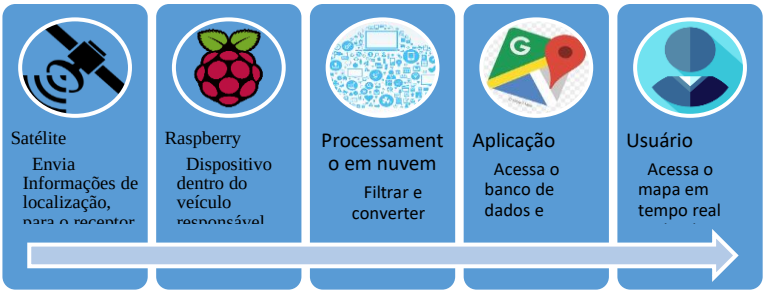
Da pesquisa realizada, verificou-se que a problemática da empresa é não possuir um controle específico de embarque-desembarque e conhecimento das rotas estabelecidas. A partir disso, há aqui a intenção de buscar uma solução de baixo custo, onde possa melhorar a segurança de sua frota, e de seus passageiros, e disponibilizar esse serviço para que seus clientes possam otimizar seu tempo através do monitoramento, e com gerenciamento do mesmo visando evitar problemas.

Após uma análise das possibilidades disponíveis para desenvolvimento, a opção de escolha ocorreu por se possuir conhecimento prévio das ferramentas. Para a realização do projeto, foi escolhido como hardware: um embarcado Raspberry Pi 3 modelo B, por possuir recursos tecnológicos necessários, e um módulo GPS GY-GPS6MV2 pela compatibilidade com o Raspberry. Como softwares: o Raspbian sistema operacional *Open Source* nativo do Raspberry, Node-RED e componentes da nuvem da IBM Cloud.

3.1 Modelo Proposto

O modelo de funcionamento proposto consiste em um ciclo de processos, desde o recebimento da localização via GPS, até ser tratada, gravada, e exposta em forma de mapa para o acesso do usuário final. a Figura 2 apresenta como funcionará a criação do nosso protótipo de rastreamento:

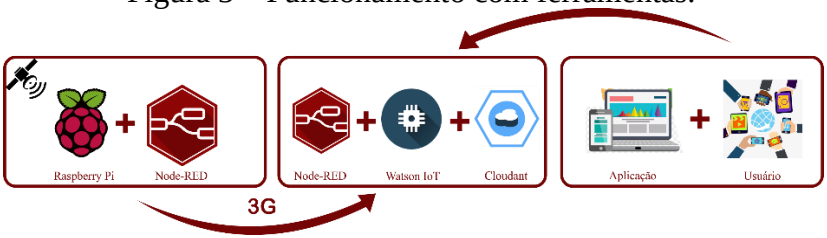
Figura 2 – Modelo proposto.



Fonte: Autor, 2018.

Abaixo, o funcionamento da aplicação com as ferramentas utilizadas:

Figura 3 – Funcionamento com ferramentas.



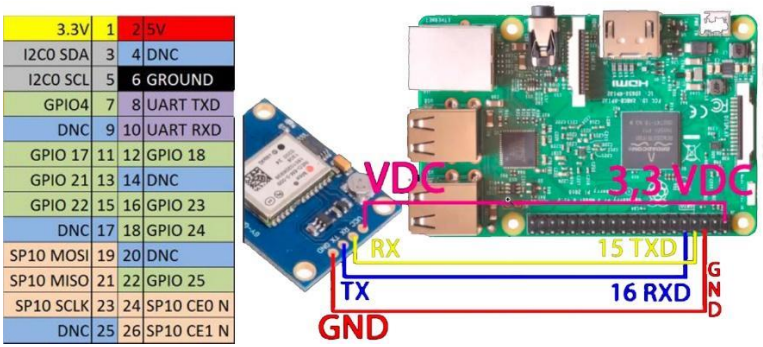
Fonte: Autor, 2018.

O Raspberry Pi recebe informações geográficas via satélite, através do módulo GPS GY-NEO6MV2, com Node-RED instalado e configurado para manipulação dos dados, que são enviados via rede 3G à nuvem da IBM, que possui uma outra instalação Node-RED para receber, e através do Watson IoT armazenar no banco de dados, onde então a informação fica pronta para o acesso da aplicação, para que então fique disponível ao usuário.

O módulo utiliza comunicação serial na GPIO do Raspberry Pi 3, onde TX é o que transmite e RX é o que recebe os dados. Opera numa tensão de 3,3V e precisa de apenas 4 fios para ser ligado. Para ligar o GPS ao

Raspberry Pi é necessário utilizar o GPIO. Na Figura 4 abaixo à esquerda há a tabela de pinos ilustrando as ligações dos fios de RX e TX na porta 8 e 10, o GROUND na porta 6, e o 3,3V na porta 1. Na imagem à direita o diagrama de ligação entre o módulo GPS e o Raspberry Pi.

Figura 4 – Tabela de pinos, Diagrama de ligação.



Fonte: GPIO do Raspberry Pi B, Raspberry.org, 2018.

3.1.1 IBM Cloud

Para desenvolvimento da aplicação, após análise, selecionou-se a IBM Cloud, cujos motivos primordiais foram: uso gratuito para criar aplicativos até 256MB de memória e por prestar um serviço profissional onde podemos contar com a garantia e integridade de alto nível, de sofisticação e escalabilidade para futura expansão.

A IBM Cloud (2018) se auto define como uma plataforma em nuvem alimentada por projetos de código aberto, com modelos de implantação integrados, sendo públicas ou dedicadas, e oferece aos desenvolvedores acesso instantâneo a mais de 150 serviços.

3.1.2 Ferramentas e Serviços

De acordo com NODERED.ORG (2018), o Node-RED é uma ferramenta de programação baseada em fluxo, originalmente desenvolvida pela equipe de Serviços de Tecnologia da IBM. Cada etapa deste fluxo de processamento corresponde a um nó que executa uma tarefa específica.

É um modelo de programação visual, acessado via navegador, que permite criar aplicativos onde seu tempo de execução é baseado em Node.js, e utiliza o protocolo MQTT para se conectar entre os nós (NODERED.ORG, 2018). A IBM junto com a JS Foundation, o transformaram em um projeto de código aberto (NODERED.ORG, 2018).

Para os dispositivos e serviços se comunicarem é necessário que ambos estejam com conexão à internet. O MQTT (Message Queue Telemetry Transport) é o protocolo padrão para comunicações de IoT (Yuan, 2017). O mesmo autor afirma que o MQTT foi inventado e desenvolvido inicialmente pela IBM, a sua principal função é comunicar hardware com a rede banda larga.

O IoT (Internet Of Things), ou a Internet das Coisas, é uma extensão da internet atual, proporciona conectar objetos do dia-a-dia com capacidade computacional a comunicar-se com a internet, embarcados, microeletrônicos e sensores, permitem ser acessados como provedores de serviços (SANTOS *et al.*, 2016).

A IBM trabalha com a plataforma Watson IoT, que é um serviço hospedado na nuvem projetado para conectar dispositivos, permitindo que os serviços interajam com os dados, processem esses dados em tempo real e sejam gravados na nuvem (IBM CLOUD, 2018).

Utilizou-se um banco de dados Cloudant, para permitir que o Watson IoT pudesse gravar informações em um local para

ser acessado pela aplicação, ou seja, aproximar os dados do usuário. O IBM Cloudant é um serviço de banco de dados NoSQL JSON, disponível na nuvem da IBM Cloud.

JSON (*JavaScript Object Notation* ou Notação de Objetos JavaScript), segundo JSON.ORG (2018) é uma formatação em forma de texto leve para troca de dados, fácil para seres humanos poderem ler e para máquinas interpretar e gerar. Basicamente, é constituído em uma coleção de pares nome/valor, e uma lista ordenada de valores.

4 Implementação do Modelo Proposto

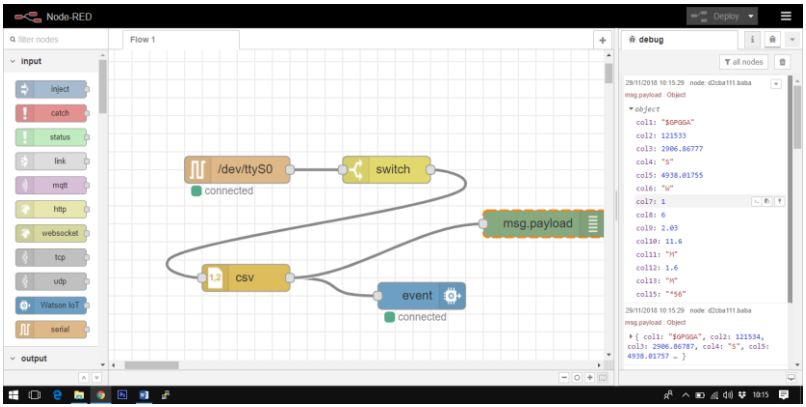
Após apresentar as principais características dos hardwares, ferramentas e serviços, foi criado um ambiente para a realização dos testes práticos com intenção de analisar os resultados da rota proposta.

O objetivo dessa implementação é apresentar o protótipo e identificar na prática os resultados, vantagens e desvantagens na utilização.

4.1 Montagem do fluxo no Raspberry Pi

A versão do Node-RED instalada é a versão 0.19.3, foi utilizado dentro do dispositivo para coletar as informações do GPS. A Figura 5 mostra como foi montado o fluxo de processamento no Node-RED dentro do Raspberry.

Figura 5 – Fluxo Node-RED Raspberry.



Fonte: Autor 2018.

O objetivo deste fluxo é coletar as informações que estão chegando do módulo GPS, filtrar as informações necessárias, convertê-las em formato JSON e através do IoT, enviar para a nuvem. O primeiro node é responsável por capturar o que trafega na porta serial que o módulo está conectado, ou seja, tudo que está chegando em formato de protocolo NMEA.

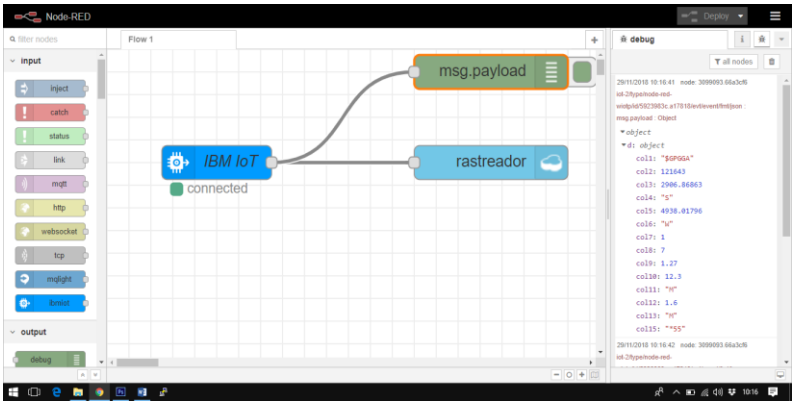
As strings que devem ser capturadas são as strings de padrão de mensagem $GPGGA, para isso foi utilizado o segundo nó, chamado switch para filtrar somente o que conter a string $GPGGA.

Para enviar essas informações filtradas, é necessário converter o formato de vírgulas. O terceiro node é responsável por esta tarefa: transformar os parâmetros para chave/valor em formato JSON. Esta informação chega ao último node que é o Watson IoT. Este componente é responsável por transmitir a informação para a nuvem.

4.2 Montagem do fluxo utilizando Node-RED na nuvem.

Dentro da nuvem é necessário um serviço de Node-RED responsável por receber essas informações e gravar em um banco de dados. A Figura 13 demonstra como foi montado os nós dentro do serviço.

Figura 6 – Fluxo Node-RED na nuvem.



Fonte: Autor 2018.

A função desse fluxo é receber essa informação proveniente do Raspberry Pi e gravar em um banco de dados Cloudant em formato JSON.

Através desse caminho fim-a-fim, essa informação fica gravada de uma forma persistente dentro da nuvem disponível e, em simultâneo, uma aplicação pode capturar esses dados no Cloudant e exibir em forma de coordenadas no mapa com a última informação gravada no banco de dados.

5 Ambiente de Testes

Com objetivo de validar o modelo proposto, uma série de testes foram realizados na prática, através do monitoramento da rota completa que leva alunos de Sombrio/SC para a UNESC em Criciúma/SC. Foi utilizado uma

aplicação criada em Android Studio 3.2.1 que busca as informações no banco de dados na nuvem e converte em forma de mapa usando uma API do Google.

O Smartphone usado para instalação do aplicativo foi um Motorola MotoG4 PLAY, modelo XT1609, processador Quad-Core, 2GB de memória RAM, tela de 5 polegadas, rede de 4G LTE Speed, com sistema operacional versão 6.0.1 Marshmallow.

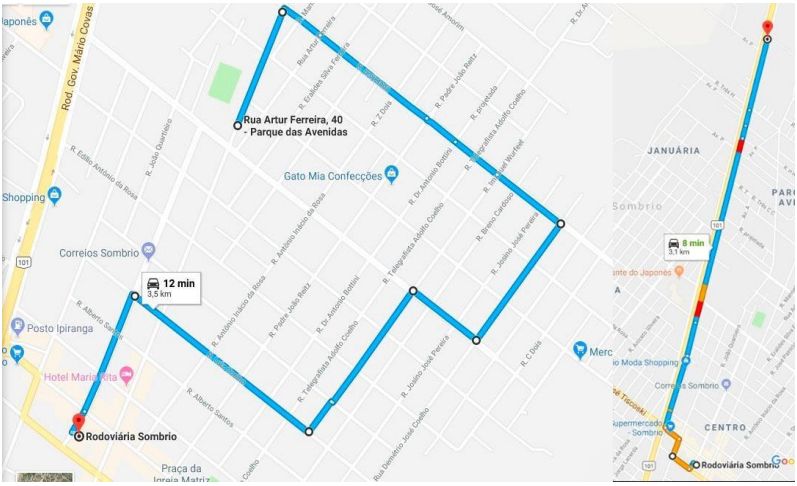
A rede utilizada para os testes foi um link de internet móvel em um plano empresarial de 5GB da operadora VIVO.

5.1 Cenário dos Testes

Para análise da rota, foram selecionados dois pontos importantes, são eles: O início da rota, onde o ônibus inicia a trajetória, passando nas paradas definidas dentro da cidade para a entrada dos passageiros/alunos e o meio da rota, que é onde contém informações, que ajudam na localização do veículo até o final do percurso.

O início é um dos pontos mais importantes, pois é um dos objetivos a serem alcançados. Através do monitoramento os usuários poderão acompanhar o deslocamento do veículo até chegar a sua parada, ou se já passou. A Figura 7 mostra a rota definida pela empresa.

Figura 7 – Rota inicial.

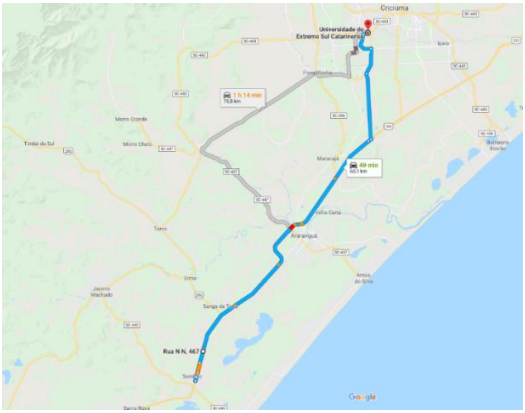


Fonte: Google Maps.

O trajeto inicial começa pela Avenida Nereu Ramos ao lado do ginásio de esportes Rogério Valerim às 17:30 e percorre os bairros Parque das Avenidas, São Luiz e Centro. Tem como parada principal a rodoviária de Sombrio por 10 minutos, sendo no bairro Guarita a última parada dentro da cidade.

O meio do trajeto entre a cidade de Sombrio/SC até Criciúma/SC informa dados de localização, em que se pode monitorar o veículo até o final da rota, que acontece na UNESC, onde o veículo faz uma parada de três horas e meia, até o final de período de aula. Após todos alunos retornarem para o veículo, o mesmo inicia o percurso de volta. Abaixo a Figura 8 mostra o caminho Meio/Fim.

Figura 8 – Trajeto meio/fim.



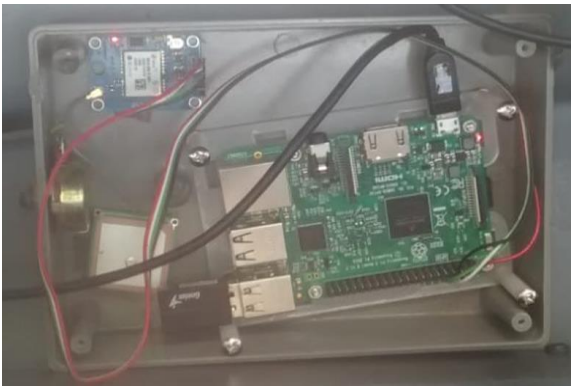
Fonte: Google Maps.

Ao retornar à cidade o trajeto inicial acontece novamente, porém, do modo inverso.

6 Resultados e Discussões

O Raspberry Pi, utilizado neste trabalho, mostrou-se uma excelente ferramenta para prototipagem de alto nível, além de sua versatilidade e portabilidade. Por possuir um tamanho reduzido e sua conexão com a internet ser pelo Wi-Fi, possibilitou ser embutido em um compartimento do veículo, próximo ao adaptador conectado que converte a energia de 24V para 5V. Antes ele foi acondicionado junto com o GPS em um compartimento de plástico para proteção. Abaixo a Figura 9 mostra o compartimento:

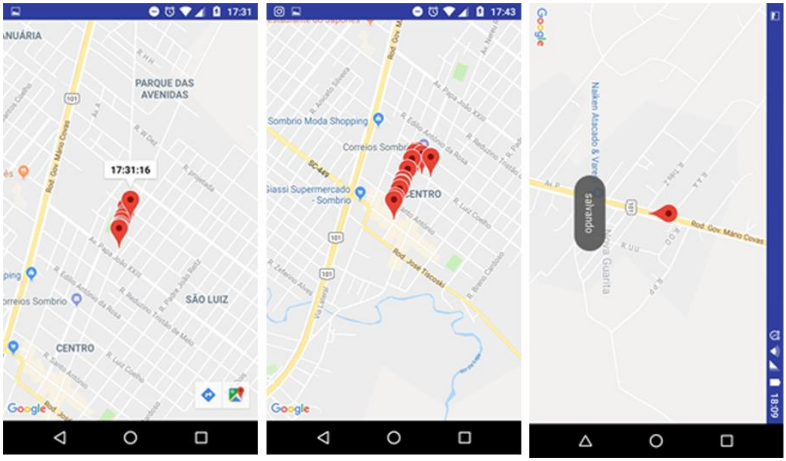
Figura 9 – Compartimento.



Fonte: Autor 2018.

Após estar alocado e conectado na rede Wi-Fi do ônibus, a rota inicial começa e o protótipo apresentou excelente funcionamento, mostrando através do aplicativo a localização do veículo em tempo real, de acordo com o trajeto esperado. Abaixo, a Figura 10 mostra a imagem das telas capturadas do aplicativo no smartphone durante o teste.

Figura 10 – Inicio da rota durante o teste.

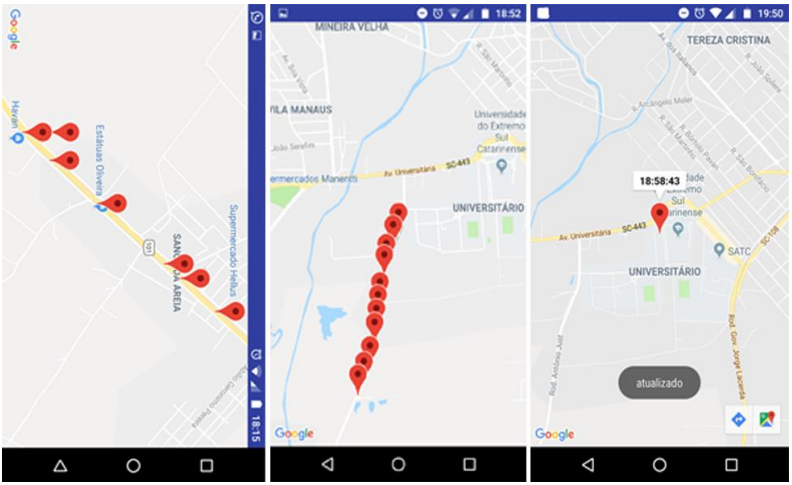


Fonte: Autor, 2018.

Como se pode observar na Figura 10, os três pontos importantes para a rota inicial são o começo da rota, onde marca o horário de partida, a rodoviária onde, embarca a maior parte dos estudantes e também onde fica a maior parte do tempo estacionado e, após a partida da rodoviária, a última parada acontece no bairro Guarita.

Após o embarque de todos alunos, o ônibus inicia o trajeto Meio/Fim, onde a maior parte do trajeto é na Rodovia Governador Mário Covas (BR 101). Até a universidade são 60 (sessenta) quilômetros, e leva em torno de 50 (cinquenta) minutos de viagem, não efetuando nenhuma parada neste período. Abaixo imagens da aplicação, na Figura 11, durante o caminho de Meio/Fim.

Figura 11 – Meio/Fim da rota durante o teste.



Fonte: Autor 2018.

Durante o período de aula, o veículo fica estacionado, então mostra no marcador a hora em que o veículo foi desligado. Ao fim do período de aula quando o veículo é ligado novamente a hora é atualizada, e após o embarque de todos os passageiros,

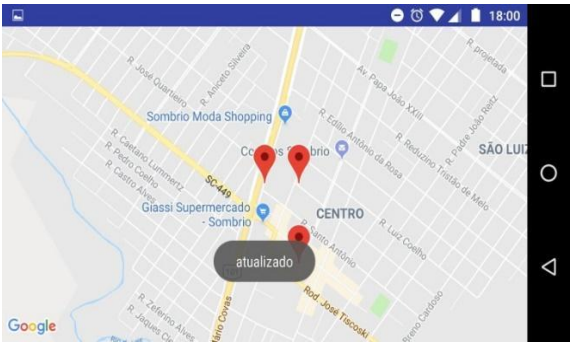
iniciasse o trajeto de volta, efetuando o mesmo caminho no modo inverso.

6.1 Pontos Positivos e Negativos

O ponto positivo foi o funcionamento do protótipo de maneira eficiente dentro da proposta da empresa, onde os pontos essenciais eram: dentro da cidade mostrar a localização de forma que os usuários possam identificar o deslocamento do veículo (e assim poder otimizar seu tempo de organização até o embarque), bem como os proprietários poderem monitorar de forma que proporcione o controle de deslocamento da sua frota. São pontos importantes, que trazem maior segurança para a empresa e credibilidade com os clientes.

Outro ponto positivo foi o baixo investimento da empresa relacionado à compra de equipamentos. Apesar de ter funcionado da forma esperada, o protótipo ainda conta com alguns problemas no sistema. Mesmo atendendo às necessidades imediatas, o protótipo depende totalmente da rede móvel para seu funcionamento. Alguns pontos, com a ausência do sinal de telefonia ou com alto pico de consumo, já que o protótipo estava conectado a mesma rede dos passageiros, fizeram com que acontecesse algumas falhas na comunicação, com o banco de dados ficando sem gravar durante esses locais.

Figura 12 – Alto consumo da rede.



Fonte: Autor 2018.

O bairro Sanga da Toca, localizado na cidade de Araranguá, possui uma ausência de sinal de rede móvel que também faz com que aconteçam falhas. Como mostra a Figura 13, o localizador fica um período de tempo sem atualizar a localização.

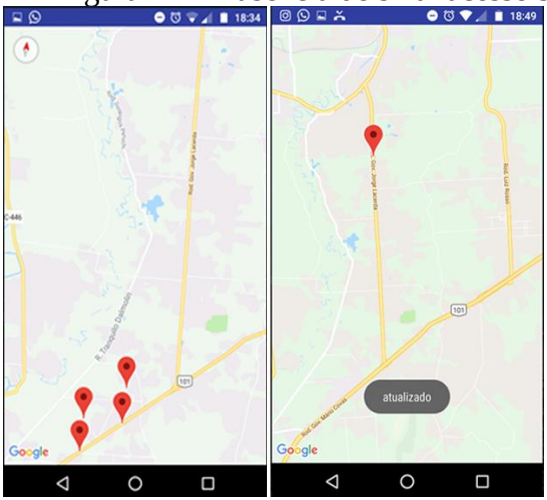
Figura 13 – Ausência de sinal Sanga da Toca.

Fonte: Autor 2018.

Outro ponto é a Rodovia Governador Jorge Lacerda, que é o acesso sul à cidade de Criciúma/SC, esta rodovia liga a BR 101 à via lateral da UNESC. Abaixo, a figura 14 mostra o trecho

com maior dificuldade de comunicação, ficando aproximadamente um período de 10 minutos sem sinal.

Figura 14 – Ausência de sinal acesso sul.



Fonte: Autor 2018.

Podemos observar na imagem o início da rodovia, onde acontece a perda do sinal e onde acontece a volta do sinal.

7 Agradecimentos à empresa Eusantur Viagens

Um dos pontos positivos da pesquisa aplicada ao estudo de caso da empresa foi a colaboração do empreendimento para o desenvolvimento do protótipo, disponibilidade de tempo para efetuar os testes e análise do ambiente onde seria aplicado, mostrando-se, assim, interessada na colaboração do projeto.

Fundada em 2010, a Eusantur Viagens está localizada na cidade de Sombrio, sul do Estado de Santa Catarina e vem desde então crescendo e se firmando no transporte de passageiros. A escolha da Eusantur Viagens para aplicação desse projeto foi devida à afinidade com os colaboradores da empresa.

8 Considerações finais

O monitoramento e rastreamento veicular pode trazer diversos benefícios, essenciais para uma empresa que procura por segurança, inovação e desenvolvimento tecnológico de comunicação. Esses fatores são de fundamental importância para as organizações de pequeno, médio ou grande porte.

Assim, diante desta realidade, o objetivo deste trabalho foi desenvolver um sistema de monitoramento veicular em tempo real, utilizando Raspberry Pi. Para isso, foi feito uma pesquisa aplicada a estudo de caso de uma empresa de fretamento.

Considerando o que foi apresentado nesse trabalho, pode-se concluir que com um baixo investimento de hardware e o uso de ferramentas gratuitas é possível desenvolver um sistema de monitoramento funcional que auxilie o administrador da empresa à coleta de informações.

Após o desenvolvimento do sistema para monitoramento, realizou-se testes através de uma rota. O protótipo atende parcialmente, ou de forma satisfatória, a todos requisitos que a empresa procura, podendo ajudar na segurança dos veículos, através do monitoramento em casos de roubo e de acidentes, onde a empresa pode tomar as devidas providências. Por sua vez, os usuários podem utilizá-lo para otimizar seu tempo, monitorando a localização e deslocamento do veículo.

O protótipo está em sua fase inicial, possui alguns problemas e portanto para trabalhos futuros propõe-se ampliar as funcionalidades e reparar as falhas. Medidas serão adotadas para evolução, como: quando houver ausência de sinal de rede gravar off-line, para quando o sinal voltar marcar o trajeto no mapa, também, utilizar um link de internet próprio para o protótipo, entre outras futuras alterações de layout.

Referências

ASSIS, Paulo Ueiner Moreira de. **Sistema de rastreamento de veículos para empresas de transporte utilizando navegação por satélite**. 2010. 121 f. Trabalho de Conclusão de Curso (Graduação em Engenharia da Computação) – Centro Universitário de Brasília. Disponível em: < http://www.repositorio.uniceub.br/bitstream/123456789/3383/3/20516524.pdf>. Acesso em: 10 set. 2018.

BRANCO, A. **Monitoramento e Rastreamento. Como escolher?** Artigos. Comunidade adm 2009. Disponivel em: http://www.administradores.com.br/artigos/marketing/monitoramento-e-rastreamento-como-escolher/29532/

CAMPOS, Antônio Carlos. **Redes geográficas e coordenadas geográficas**. Disponível em:< http://www.cesadufs.com.br/ORBI/public/uploadCatalago/111193804042012Cartografia_Basica_Aula_7.pdf >. Acesso em: 04 set. 2018.

CANDIDO, Luciano Manoel; CRUZ ,Hélio Alves da. **Análise do sistema de rastreamento e monitoramento de rota em um centro de distribuição**. In: SIMPÓSIO DE EXCELÊNCIA NA GESTÃO E TECNOLOGIA, 7., 2015, Rio de Janeiro. **Anais...** Rio de Janeiro: AEDB, 2015, p. 1-10. Disponível em: < https://www.aedb.br/seget/arquivos/artigos15/35722526.pdf > .Acesso em: 04 set. 2018.

CARVALHO, Edilson Alves de; ARAÚJO, Paulo César de. **Localização: Coordenadas geográficas**. Universidade do Rio Grande do Norte e Universidade Estadual da Paraíba. Rio Grande do Norte, 2008. (Apostila). Disponível em: <http://www.ead.uepb.edu.br/ava/arquivos/cursos/geografia/leituras_cartograficas/Le_Ca_A08_J_GR_260508.pdf>. Acesso em: 04 set. 2018.

CERVI, Marina. **Pesquisa básica e pesquisa aplicada***: o que são e suas importâncias*. Galoá Journal, São Paulo: Campinas, 2018. Disponível em: <https://galoa.com.br/blog/pesquisa-basica-e-pesquisa-aplicada-o-que-sao-e-suas-importancias> .Acesso em: 14 out 018.

CUNHA, Alessandro F. **O que são sistemas embarcados?** 2015. Disponível em: <https://files.comunidades.net/mutcom/ARTIGO_SIST_EMB.pdf>. Acesso em: 21 nov. 2018.

FLEURY, Maria Tereza Leme; WERLANG, Sérgio. **Pesquisa aplicada – reflexões sobre conceitos e abordagens metodológicas**. FGV, Rio de Janeiro, p. 1 – 5, 2010.Disponível em: <https://bibliotecadigital.fgv.br/dspace/bitstream/handle/10438/18700/A_pesquisa_aplicada_conceito_e_abordagens_metodol%C3%B3gicas.pdf> .Acesso em: 29 out. 2018.

GALON, Handrey Emanuel. **Sistema de rastreamento e controle de recursos de um veículo utilizando um smartphone android**. 2014. 79 f. Trabalho de Conclusão de Curso (Graduação em Engenharia da Computação) – Universidade Tecnológica Federal do Paraná, Pato Branco, 2014. Disponível em: < http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/4068/1/PB_COENC_2014_1_04.pdf>. Acesso em: 04 set. 2018.

HASEGAWA, Júlio, K.; GALO, Mauricio; MONICO, João, F. G.; IMAI, Nilton, N., Sistema de localização e navegação apoiado por GPS, XX Congresso Brasileiro de Cartografia, Recife – PE, 1999.Disponivel em: <https://www.researchgate.net/publication/228585655_Sistema_de_Localizacao_e_Navegacao_apoiado_por_GPS> Acesso em 02 dez. 2018.

HEIN, Werner. PI, Raspberry. Raspberry Pi aplicado a projetos do mundo real. **Linux Magazine**, mar. 2013. Disponível em: < http://www.linuxnewmedia.com.br/images/uploads/pdf_aberto/LM_100_60_65_06_tut_raspberry_pi.pdf>. Acesso em: 01 out. 2018.

IBM Clound. About IBM: **IBM CLOUD**. 2018. Disponível em: <https://www.ibm.com/cloud/>. Acesso em: 21 nov. 2018.

JSON.ORG. **Introducting JSON.** 2018. Disponível em: <https://www.json.org/>. Acesso em: 21 nov. 2018.

Luiz. Paulo. Gestão de Operação e logística. **IMPACTO DOS SISTEMAS INTEGRADOS DE RASTREAMENTO NA LOGÍSTICA EMPRESARIAL**. Art. 2008. Disponível em: <http://www.administradores.com.br/producao-academica/impacto-dos-sistemas-integrados-de-rastreamento-na-logistica-empresarial/562/> Acesso em: 02 dez. 2018.

MANTUANO, Gabriela. **Rastreamento veicular entenda a importância**! SOFTRUCK, Argentina, 2018. Disponível em: < http://blog.softruck.com/a-importancia-do-rastreamento-veicular/>. Acesso em: 27 out. 2018.

MISTRETTA, Larissa Franco; JÚNIOR, Osmar Delmanto. Implantação de sistema de rastreamento e monitoramento de frota e simulação de rota de uma empresa de bebidas. **Tékhne ε Lógos**, Botucatu, SP, v.3, n.2, Julho, 2012. Disponível em: < http://www.fatecbt.edu.br/seer/index.php/tl/article/view/133 > .Acesso em 29 out. 2018.

NATIONAL MARINE ELETRONICS ASSOCIATION. Nmea: **About the NMEA.** 2008. Disponivel em: <

https://www.nmea.org/content/about_the_nmea/about_the_nmea.asp> Acesso em: 02 dez. 2018.

NODE-RED. Node-Red: **Flow-based programming for the internet of things**. JS Foundation, 2018. Disponível em: <https://nodered.org/>. Acesso em: 21 nov 2018.

PEREIRA, Luiz Arthur Malta et al. Software embarcado, o crescimento e as novas tendências deste mercado. **Revista de ciências exatas e tecnologia**, Londrina, v. 6,n.6, p.85-94, 2011. Disponível em: <http://www.pgsskroton.com.br/seer/index.php/rcext/article/view/2308> .Acesso em: 21 nov. 2018.

PRADO, Jaime; PEINADO, Jurandir; GRAEML, Alexandre Reis. Percepção dos benefícios do uso de sistemas de rastreamento de veículos pelos transportadores rodoviários. **Brazilian Business Review,** Vitória, v. 7, n. 2, p. 1 – 20, mai/ago, 2010. Disponível em: < http://www.scielo.br/pdf/bbr/v13n6/pt_1808-2386-bbr-13-06-0210.pdf> .

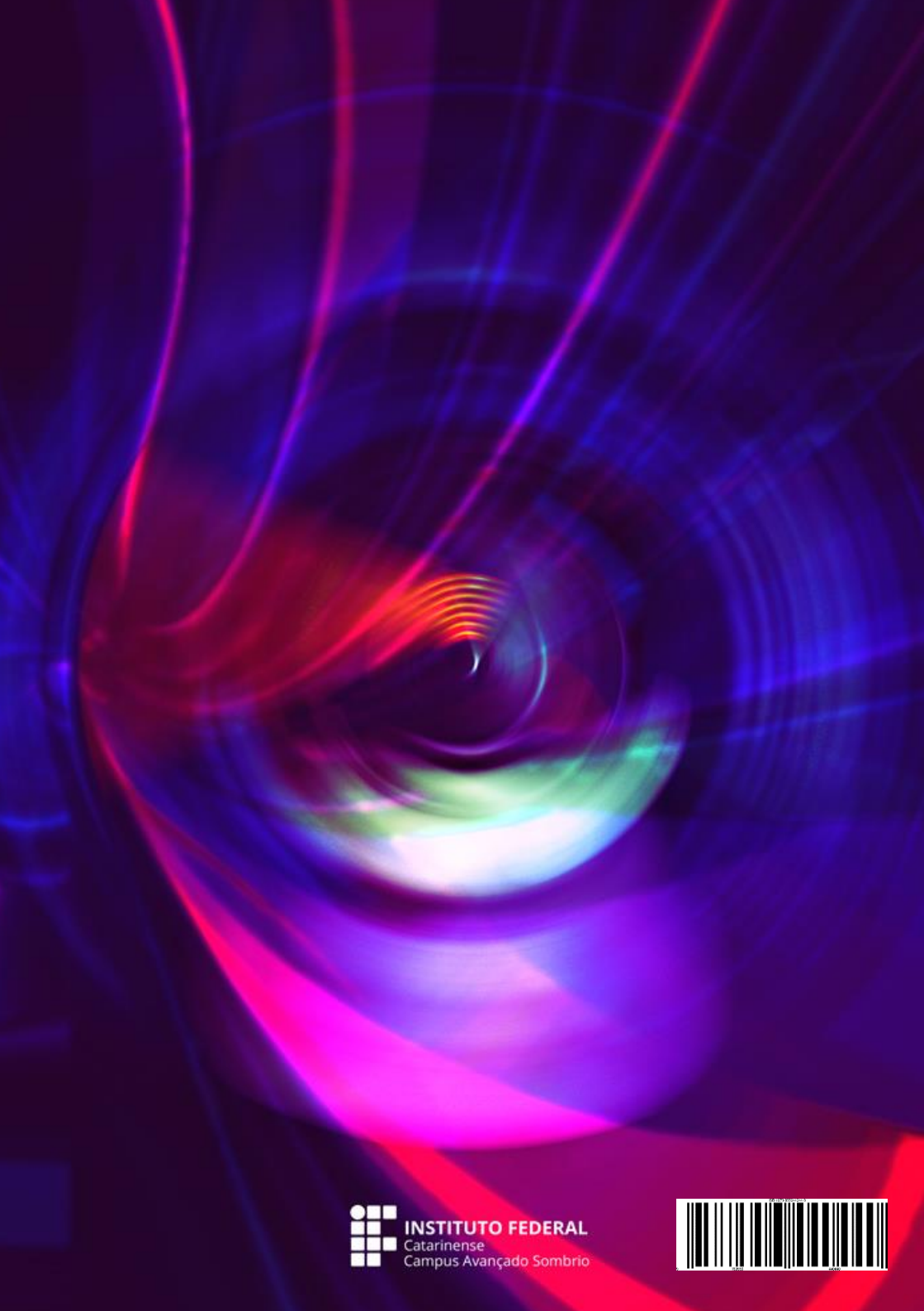
RASPBERRY PI. **Sobre nós**. Disponível em: <https://www.raspberrypi.org/about/>. Acesso em: 01 out. 2018.

RASPBIAN. **Welcome to Raspbian**. Reino Unido, 2012. Disponível em: <https://www.raspbian.org> .Acesso em: 27 out 2018.

SANTOS, Bruno. et al. Internet das Coisas: da Teoria à Prática. In: SIQUEIRA, Frank Augusto, et al. (Orgs). **Livro de Minicursos** *SBRC 2016*, Porto Alegre: SBC, 2016. Disponível em: <http://www.sbrc2016.ufba.br/downloads/anais/MinicursosSBRC2016.pdf> .Acesso em: 27 out. 2018.

VARGAS, Rafael et al. *Sistemas embarcados:* **acoplamento do soft-core plasma ao barramento opb de um powerpc 405**. Florianópolis: Universidade Federal de Santa Catarina, 2007. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_440/Artigo%20-%20Sistemas%20Embarcados.pdf>. Acesso em:21 nov. 2018.

YUAN, Michael. **Conhecendo o MQTT**. IBM . Nova York: Armonk, 2017. Disponível em: < https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html> .Acesso em: 27 out 2018.



INSTITUTO FEDERAL
Catarinense
Campus Avançado Sombrio