UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE ESPECIALIZAÇÃO SEMIPRESENCIAL EM CONFIGURAÇÃO E GERENCIAMENTO DE SERVIDORES E EQUIPAMENTOS DE REDES

RAINER TESTA MEDRADO

MONITORAMENTO DE ATIVOS DE REDE UTILIZANDO SOFTWARES OPEN-SOURCE

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA
2018

RAINER TESTA MEDRADO

MONITORAMENTO DE ATIVOS DE REDE UTILIZANDO SOFTWARES OPEN-SOURCE

Monografia de Especialização, apresentada ao Curso de Especialização Semipresencial em Configuração e Gerenciamento de Servidores e Equipamentos de Redes, do Departamento Acadêmico de Eletrônica – DAELN, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. M.Sc. Juliano de Mello Pedroso

CURITIBA

2018

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Curitiba

Diretoria de Pesquisa e Pós-Graduação
Departamento Acadêmico de Eletrônica
Curso de Especialização Semipresencial em Configuração e Gerenciamento de Servidores e Equipamentos de Redes

TERMO DE APROVAÇÃO

MONITORAMENTO DE ATIVOS DE REDE UTILIZANDO SOFTWARES OPEN-SOURCE

por

RAINER TESTA MEDRADO

Esta monografia foi apresentada em 20 de novembro de 2018 como requisito parcial para a obtenção do título de Especialista em Configuração e Gerenciamento de Servidores e Equipamentos de Redes. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

_____
Prof. M.Sc. Juliano de Mello Pedroso
Orientador

_____
Prof. Dr. Kleber Kendy Horikawa Nabas
Membro titular

_____
Prof. M. Sc. Omero Francisco Bertol
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

RESUMO

MEDRADO, Rainer Testa. **Monitoramento de ativos de rede utilizando softwares open-source**. 2018. 49 p. Monografia de Especialização em Configuração e Gerenciamento de Servidores e Equipamentos de Redes, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

Este trabalho apresenta um estudo de caso prático sobre o uso de soluções de monitoramento em um ambiente de *datacenter*. São apresentados conceitos essenciais sobre o protocolo *SNMP*, protocolo este que foi desenvolvido para facilitar o monitoramento e gerenciamento de redes. Também são apresentados os softwares Zabbix *e o* Cacti*,* com seu plugin PHP Network Weathermap, onde será documentado todos procedimentos realizados, da instalação a configuração do serviço de monitoramento, da configuração de mapas e uso de *templates e* desenvolvimento de mapa de uso dos links da dados, tudo de acordo com a documentação oficial dos softwares.

Palavras-chave: SNMP. Métricas de rede. Gerenciamento.

ABSTRACT

MEDRADO, Rainer Testa. **Monitoring network assets using open-source software**. 2018. 49 p. Monografia de Especialização em Configuração e Gerenciamento de Servidores e Equipamentos de Redes, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

This paper presents a practical case study on the use of monitoring solutions in a datacenter environment. Essential concepts are presented on the SNMP protocol, which was developed to facilitate the monitoring and management of networks. Also presented are Zabbix and Cacti software, with its PHP Network Weathermap plugin, where all procedures performed will be documented, from installation to configuration of the monitoring service, configuration of maps and use of templates and development of link use map of the data, all according to the official documentation of the software.

Keywords: SNMP. Network metrics. Management.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

ASN.1	*Abstract Syntax Notation.1*
CDP	*Cisco Discovery Protocol*
GNU	*Gnu Not Unix*
ICMP	*Internet Control Message Protocol*
IP	*Internet Protocol*
LLD	*Low Level Discovery* (ou Descoberta de Baixo Nível)
MIB	*Managment Information Base* (ou Base de Informação de Gerenciamento)
NMS	*Network Managment Systems* (ou Sistema Gerenciador de Redes)
RFC	*Request for Comments*
SNMP	*Simple Network Management Protocol*
SNMP	*Simple Network Managment Protocol* (ou Protocolo de Gerência Simples de Rede)
TCP/IP	*Transmission Control Protocol/Internet Protocol* (ou Protocolo de Controle de Transmissão/Protocolo da Internet)
TI	Tecnologia da Informação
UDP	*Usre Datagram Protocol*
XMPP	*eXtensible Messaging and Presence Protocol*

SUMÁRIO

1 INTRODUÇÃO

Realizar o monitoramento dos ativos de um datacenter é essencial para uma administração eficaz, em que se espera um tempo mínimo na resolução de problemas e alta disponibilidade no uso dos recursos oferecidos. Falhas nos sistemas podem trazer prejuízos imensuráveis, podendo até, serem irrecuperáveis.

O monitoramento de uma rede operacional pode fornecer ao administrador de rede as informações para gerenciar a rede dinamicamente e relatar as estatísticas de uso de rede. Atividade do link, taxas de erro e status do link são alguns dos fatores que ajudam o administrador de rede a determinar a integridade e a utilização de uma rede. A coleta e a análise dessas informações ao longo do tempo permitem que o administrador de rede visualizar e projete o crescimento, e podem possibilitar que o administrador encontre e substitua peças defeituosas antes de uma falha total.

O gerenciamento eficiente de uma rede de computadores permite que falhas possam ser identificadas e prevenidas rapidamente, com o intuito de minimizar o impacto sobre os usuários e diminuir os prejuízos da instituição. Realizar auditoria em logs de maneira rápida é imprescindível para um administrador de redes assim como buscar e minimizar as vulnerabilidades encontradas no seu ambiente.

Para facilitar e otimizar essa administração, diversas ferramentas foram desenvolvidas com este intuito, gerando alertas em tempo real de problemas apresentados, administrando todos os recursos em uso desses equipamentos, como uso de memória, espaço em disco, resposta a consultas ICMP.

Neste trabalho serão apresentadas duas ferramentas de monitoramento. A primeira será o popular Zabbix, ferramenta amplamente utilizada pela comunidade de software livre, onde será abordado desde sua instalação até a coleta das métricas. Outra ferramenta será o Cacti, em conjunto com seu *plugin* para construção de mapas de monitoramento de links, o PHP Network Weathermap.

1.1 OBJETIVOS

Para a realização deste trabalho, são apresentados os objetivos gerais e específicos nas seções seguintes.

1.1.1 Objetivo Geral

Apresentar solução e implementação para monitoramento dos ativos de um *datacenter*, de modo a realizar o monitoramento de diversos itens, como *AccessPoints*, *Switchs*, Roteadores, *Links* de dados e Servidores.

1.1.2 Objetivos Específicos

Para atender ao objetivo geral neste trabalho de conclusão de curso os seguintes objetivos específicos serão abordados:

- Instalar e configurar um servidor de monitoramento, utilizando o software Zabbix em ambiente *Linux Debian*, bem como, dos agentes nos dispositivos clientes.
- Realizar a configuração nos equipamentos monitorados de modo a buscar os principais parâmetros do equipamento a ser gerenciado.
- Gerar mapas para facilitar a localização dos ativos na rede em caso de problemas e *templates* que explorem melhor as características dos equipamentos monitorados.
- Criar mapa de uso do *link* de dados utilizando a ferramenta *PHPNetworkWeathemap*.

1.2 JUSTIFICATIVA

O ambiente de um datacenter possui uma gama de equipamentos que necessitam de constante acompanhamento, seja para uma melhor gestão dos recursos disponibilizados, seja na resolução de forma eficaz e rápida dos problemas apresentados. O correto gerenciamento de falhas nos serviços ofertados, é fundamental para que a rede de um datacenter consiga se manter estável, disponível e com qualidade na entrega dos serviços disponibilizados.

A relevância desse trabalho se dá pela importância que as ferramentas de monitoramento dos ativos de uma rede possuem, onde ter acesso fácil a essas informações otimiza o trabalho da equipe técnica e do gestor na tomada de decisões.

1.3 ESTRUTURA DO TRABALHO

Este trabalho foi desenvolvido com o uso das soluções de monitoramento *open source Zabbix e Cacti (plugin PHP Network Weathermap).* A versão do Zabbix utilizada foi a última versão estável, que atualmente (ago. 2018) está na versão 4 (ZABBIX, 2018). A versão do Cacti utilizada foi a 0.8.8, por questões de compatibilidade com o plugin de mapas *PHPNetworkWeathermap.*

As seguintes etapas serão realizadas durante o desenvolvimento do trabalho:

1. Pesquisar:
   - O protocolo SNMP.
   - O software Zabbix.
   - O sftware Cacti e o *plugin PHP Network Weathermap*
2. Especificar:
   - Os equipamentos que serão monitorados.
   - Os parâmetros de monitoramento destes equipamentos.
3. Implementar:
   - Realizar a instalação do sistema de monitoramento.
   - Configurar parâmetros necessários para bom funcionamento.
   - Instalação dos agentes SNMP nos equipamentos clientes.
   - Mapas do ambiente monitorado.
4. Testar:
   - Testar e validar instalações realizadas.

2 REFERENCIAL TEÓRICO

2.1 GERENCIAMENTO DE REDES

Segundo Lessa (1999), “Estatisticamente, enquanto 30% dos custos de uma infraestrutura computacional estão diretamente associados à aquisição de hardware, os 70% restante dizem respeito à manutenção e suporte aos recursos e serviços nela contida.” Portanto, o monitoramento da infraestrutura computacional, torna-se uma atividade que contribui decisivamente para o funcionamento contínuo dos serviços oferecidos, garantindo que a qualidade destes mantenha-se em níveis satisfatórios pelo maior tempo possível.

Independente do tamanho de uma rede de computadores, ela precisa ser gerenciada, para garantir aos usuários qualidade e disponibilidade de serviços ao um nível de desempenho aceitável. Por isso é importante para uma equipe de Tecnologia da Informação (TI) conhecer informações sobre os componentes de sua rede, como: seus equipamentos de rede (*switch*, repetidores, roteadores, entre outros), especificação de hardware e software dos seus servidores e estações, os serviços disponíveis aos seus usuários e etc.

O objetivo da Gerência de Redes é monitorar e controlar os elementos da rede (sejam eles físicos ou lógicos), assegurando um certo nível de qualidade de serviço. Para realizar esta tarefa, os gerentes de redes são geralmente auxiliados por um sistema de gerência de redes. Um sistema de gerência de rede pode ser definido como uma coleção de ferramentas integradas para a monitoração e controle da rede. Teixeira Júnior et al. (1999) diz também que essas ferramentas são controladas pela equipe de TI, e que cabe a ela escolher as melhores ferramentas e a melhor maneira de controlar esses recursos. Estes sistemas oferecem uma interface única, com informações sobre a rede e podem oferecer também um conjunto poderoso e amigável de comandos que são usados para executar quase todas as tarefas da gerência da rede (STALLINGS, 2005).

Segundo o Núcleo de Informação e Coordenação do Ponto BR (NIC-BR, 2018), através do gerenciamento de rede é possível conhecer as métricas dos dispositivos para um funcionamento correto, como uso de memória, disco, link de

dados, entre outros parâmetros. Ela cita alguns pontos como importantes no gerenciamento de redes:

- Conhecer as métricas dos dispositivos para um funcionamento correto;
- Conhecer bem a topologia da rede, ter um inventário dos ativos da rede;
- Backup automático de todas configurações da rede, de servidores, roteadores, equipamentos;
- Conhecer a *baseline* de métricas dos dispositivos para tomada de decisões;
- Utilizar ferramentas que colete métricas dos dispositivos para toma de decisões;
- Gestão de logs dos equipamentos.

Através do uso de ferramentas de gestão de redes, podemos antecipar falhas em equipamentos, antecipar incidentes de segurança, como tentativas de invasão e resolver gargalo de desempenho nos sistemas (quando algum dispositivo não consegue dar mais conta dos serviços da rede, seja porque a rede cresceu ou algum evento fora da normalidade forçou um volume maior de trabalho momentaneamente).

2.2 PROTOCOLO SNMP

O protocolo *Simple Network Managment Protocol* (SNMP, ou Protocolo de Gerência Simples de Rede) é um protocolo de gerência típica de redes *Transmission Control Protocol/Internet Protocol* (TCP/IP, ou Protocolo de Controle de Transmissão/Protocolo da Internet), da camada de aplicação que facilita o intercâmbio de informação entre os dispositivos de rede. Ele possibilita aos administradores de rede gerenciar o desempenho da rede, encontrar e resolver problemas de rede e planejar o crescimento desta (COSTA, 2008). Ele roda sobre o protocolo UDP (PAVENTHAN et al., 2013; LU et al., 2015) e sua última especificação é a RFC-1157 (CASE et al., 1990).

Em Netacad (2018), também diz que ele foi desenvolvido para permitir que os administradores gerenciem nós, como servidores, estações de trabalho, roteadores, switches e dispositivos de segurança, em uma rede IP. Permite que os administradores de rede monitorem o desempenho da rede, encontrem e resolvam os problemas da rede e planejem o crescimento da rede.

Segundo Matos (2009), o protocolo SNMP é a evolução do padrão SGMP, ele foi criado pela NYSERNet Inc. juntamente com a colaboração de várias universidades de Nova York, quando foi estabelecido a necessidade de gerenciar uma internet e não apenas monitorá-la, também da necessidade de monitorar outros equipamentos, e não somente gateways da rede (FACHINI, 2010).

2.2.1 Componentes Básicos do SNMP

Segundo Fachini (2010), uma rede gerenciada pelo protocolo SNMP é formada por três componentes chaves. São elas: 1) Dispositivos Gerenciados; 2) Agentes; e 3) Sistema Gerenciador de Redes (ou *Network Managment Systems* - NMS).

Um Dispositivo Gerenciado é um nó de rede que possui um agente SNMP instalado e se encontra em uma rede gerenciada. Estes dispositivos coletam e armazenam informações de gerenciamento e mantém essas informações disponíveis para sistemas NMS através do protocolo SNMP. Dispositivos gerenciados, também as vezes denominados de dispositivos de rede, podem ser roteadores, servidores de acesso, impressoras, computadores, servidores de rede, *switches*, dispositivos de armazenamento, dentre outros (KAKANAKOV; KOSTADINOVA; SPASOV, 2007).

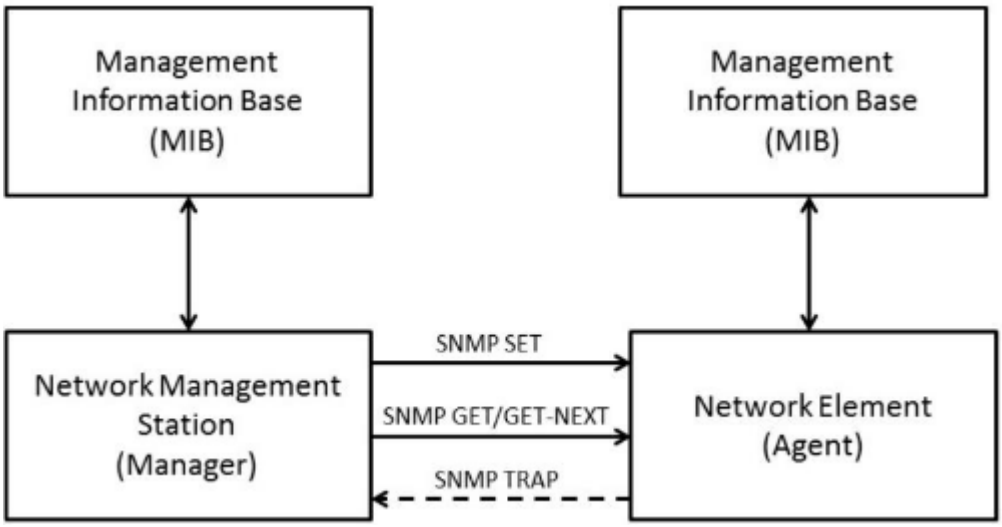
Um Agente é um módulo de software de gerenciamento de rede que fica armazenado em um Dispositivo Gerenciado. Um agente tem o conhecimento das informações de gerenciamento locais e traduz estas informações para um formato compatível com o protocolo SNMP. Mauro e Shimidt (2001), enfatizam ainda que é papel do agente fornecer informações de gerenciamento às estações de gerenciamento NMS, mantendo o controle de vários aspectos operacionais do dispositivo monitorado.

Um sistema NMS é responsável pelas aplicações que monitoram e controlam os Dispositivos Gerenciados. Normalmente é instalado em um (ou mais de um) servidor de rede dedicado a estas operações de gerenciamento, que recebe informações (pacotes SNMP) de todos os dispositivos gerenciados daquela rede (KAKANAKOV; KOSTADINOVA; SPASOV, 2007).

2.2.2 Arquitetura

O framework SNMP, apresentado na Figura 1, consiste de: a) Agentes Mestres (*Master Agents*), b) Sub-agentes (*Subagents*), e c) Estações de Gerenciamento (*Management Stations*).

Figura 1 - Arquitetura SNMP



Fonte: Lima, Fresse e Rousseau (2014).

O SNMP define como informações de gerenciamento são trocadas entre os aplicativos de gerenciamento de rede e os agentes de gerenciamento. O gerenciador de SNMP pesquisa os agentes e consulta a *Managment Information Base* (MIB, ou Base de Informação de Gerenciamento) sobre os agentes de SNMP na porta UDP 161. Os agentes de SNMP enviam as interceptações de SNMP para o gerenciador de SNMP na porta UDP 162.

- *Master Agent*:

O *Master Agent* em uma rede gerenciada é, na verdade, um software sendo executado em um dispositivo com suporte a SNMP, por exemplo, um roteador, que interage com uma estação de gerenciamento. É o equivalente a um servidor, na comunicação cliente/servidor, ou a um *daemon*, sob o ponto de vista de sistemas operacionais. Os subagentes são os responsável por passarem informações específicas para o *Master Agent* (KURYLA; SCHÖNWÄLDER, 2011).

- *Subagent*:

Os *subagents* são pequenos programas em execução no dispositivo com suporte a SNMP, responsáveis pelo monitoramento de recursos específicos naquele

dispositivo como, por exemplo, o status de um link *ethernet* em um roteador, ou a quantidade de espaço livre em um disco de um servidor.

Algumas características dos softwares subagentes são (KURYLA; SCHÖNWÄLDER, 2011):

- Coletar informações de objetos gerenciados
- Configurar parâmetros destes objetos gerenciados
- Responder a solicitações do software de gerência da rede
- Gerar alarmes ou *traps* em determinadas situações

- *Managment Station*:

O Gerente da Rede ou Estação de Gerenciamento ou ainda *Managment Station* é o componente final da arquitetura de uma solução SNMP. Funciona como um cliente em uma comunicação cliente/servidor. Realiza requisições de informações aos dispositivos gerenciados, que podem ser temporárias ou através de comandos a qualquer tempo. E ainda é o responsável por receber alarmes gerados pelos agentes e gerar saídas para estes alarmes, tais como, alterar (SET) o valor de um determinado parâmetro gerenciado no equipamento, enviar mensagem para o celular do administrador da rede, dentre outras (COSTA, 2008).

2.2.3 O SNMP e o ASN.1

O SNMP é um protocolo padrão usado para gerência de redes, que define os formatos dos pedidos que o Gerente envia para o Agente e os formatos das respostas que o agente retorna, assim como o significado exato de cada pedido e resposta. Uma mensagem SNMP é codificada com um padrão designado de ASN.1 (do inglês, *Abstract Syntax Notation.1*) (COSTA, 2008).

Para permitir a transferência de grandes inteiros, sem desperdiçar espaço em cada transferência, o ASN.1 usa uma combinação de tamanho e valor para cada objeto a ser transferido.

2.2.4 Comandos do SNMP

O SNMP não define um grande número de comandos, em lugar disso define duas operações básicas:

- *fetch*: para obter um valor de um dispositivo;
- *store*: para colocar um valor em um dispositivo.

O comando que especifica uma operação de *fetch* ou *store* deve especificar o nome do objeto, que é único.

2.2.5 Strings de Comunidade

Para que o SNMP opere, o NMS deve ter acesso à MIB, para que possa assegurar que as solicitações de acesso sejam válidas, uma forma de autenticação deve ser estabelecida.

O SNMPv1 e o SNMPv2c usam sequência de caracteres de *community* que controlam o acesso à MIB. As strings de comunidade são senhas de texto puro. As strings de comunidade SNMP autenticam o acesso a objetos MIB.

Há dois tipos de sequência de caracteres de *community*, são elas:

- Somente leitura (*Read Only* - RO): Fornece acesso às variáveis de MIB, mas não permite que essas variáveis sejam alteradas, o acesso é somente para leitura. Pelo fato da segurança ser mínima na versão 2c, muitas organizações usam o SNMPv2c no modo somente leitura.
- Leitura-gravação (*Read-Write* - RW): Fornece acesso de leitura e gravação a todos os objetos na MIB.

Para visualizar ou definir as variáveis de MIB, o usuário deve especificar a string de comunidade correto para o acesso de leitura ou gravação.

2.2.6 Nomes de Objetos e MIB

Todos os objetos acessados pelo SNMP devem ter nomes únicos definidos e atribuídos. Além disso, o Gerente e o Agente devem acordar os nomes e significados das operações *fetch* e *store*. O conjunto de todos os objetos SNMP é coletivamente conhecido como *Managment Information Base* (MIB). O padrão SNMP não define o MIB, mas apenas o formato e o tipo de codificação das mensagens. A

especificação das variáveis MIB são, assim como o significado das operações *fetch* e *store* em cada variável, especificados por um padrão próprio.

A árvore de MIB para um dispositivo específico inclui alguns ramos com variáveis comuns para muitos dispositivos de rede e alguns ramos com variáveis específicas para esse dispositivo ou fornecedor. Os RFCs definem algumas variáveis públicas comuns. A maioria dos dispositivos implementa essas variáveis de MIB. Além disso, os fornecedores de equipamentos de rede, como a Cisco, podem definir seus próprios ramos privados da árvore para acomodar as novas variáveis específicas para os dispositivos (NETACAD, 2018).

A definição dos objetos do MIB é feita com o esquema de nomes do ASN.1, o qual atribui a cada objeto um prefixo longo que garante a unicidade do nome, a cada nome é atribuído um número inteiro. Também, o SNMP não especifica um conjunto de variáveis, e que a definição de objetos é independente do protocolo de comunicação, permite criar novos conjuntos de variáveis MIB, definidos como padrões, para novos dispositivos ou novos protocolos. Por isso, foram criados muitos conjuntos de variáveis MIB, que correspondem a protocolos como UDP, IP, ARP, assim como variáveis MIB para hardware de rede como Ethernet ou FDDI, ou para dispositivos tais como *brigdes, switches* ou impressoras (COSTA, 2008).

2.2.7 SNMPv2 e SNMPv3

A versão 2 do SNMP é uma evolução do protocolo inicial. O SNMPv2 oferece uma boa quantidade de melhoramentos em relação ao SNMPv1, incluindo operações adicionais do protocolo, melhoria na performance, segurança, confidencialidade e comunicação Gerente/Gerente. A padronização de um outra versão do SNMP, no caso a versão 3, está definida nas RFC3411 – RFC3418.

Na prática, as implementações do SNMP oferecem suporte para as múltiplas versões (RFC3584), tipicamente SNMPv1, SNMPv2c e SNMPv3.

Tabela 1 - Versões existentes do protocolo SNMP

Versão	Data	RFCs	Operações Suportadas
V1	1990	1157,1155,1212 e 1215	get, get-next, set, get-response e trap
V2c	1996	1578-2580,3416-3418 e 1901	get-bulk,inform, notification e report
V3	1999	3411, 3412, 3413, 3414 e 3415	

Fonte: Frye et al. (2003, p. 4-5).

Resumindo (NETACAD, 2018):

- SNMPv1: O *Simple Network Management Protocol*, um *Full Internet Standard*, definido no RFC 1157.
- SNMPv2c: Definido dos RFCs 1901 a 1908; utiliza a estrutura administrativa baseada na string de comunidade.
- SNMPv3: Protocolo padronizado interoperável, originalmente definido nos RFCs 2273 a 2275; oferece acesso seguro autenticando e criptografando os pacotes na rede. Inclui estas funcionalidades de segurança: integridade de mensagem para assegurar que um pacote não seja alterado em trânsito; autenticação para determinar se a mensagem é de uma fonte válida e criptografia para evitar que o conteúdo de uma mensagem seja lido por uma fonte não autorizada.

Todas as versões usam gerenciadores, agentes e MIBs de SNMP. A versão 1 é uma solução antiga e já não é encontrada com frequência nas redes atuais. SNMPv1 e SNMPv2c usam uma forma baseada na comunidade de segurança. A comunidade de gerenciadores aptos a acessar a MIB do agente é definida por uma ACL e senha. O SNMPv3 oferece modelos e níveis de segurança. Um modelo de segurança é uma estratégia de autenticação estabelecida para um usuário e o grupo em que o usuário reside. Um nível de segurança é o nível de segurança permitido em um modelo de segurança. Uma combinação de nível de segurança e modelo de segurança determina qual mecanismo de segurança é usado para processar um pacote de SNMP. Os modelos de segurança disponíveis são SNMPv1, SNMPv2c e SNMPv3 (NETACAD, 2018).

2.3 ZABBIX

O Zabbix é uma ferramenta de código aberto capaz de monitorar a disponibilidade e o desempenho da infraestrutura de diversos parâmetros de uma rede (HORST; PIRES; DÉO, 2015), sendo ideal para monitorar e controlar o funcionamento dos ativos e seus serviços. Ele possui um flexível mecanismo de alarmes que permite aos usuários configurar e-mail, mensagem instantânea e SMS para receber os alertas se algum evento ocorrer com os mecanismos gerenciados, e sendo corretamente configurado pode executar comandos remotos permitindo uma fácil resolução do problema encontrado nos ativos monitorados.

A comunidade que desenvolve o produto também fez uma especificação simplificada, própria e aberta de um protocolo de gerenciamento, também chamado de Zabbix. Isso permite que um dispositivo que não suporte protocolos comuns como o SNMP possa incluir um agente Zabbix e expor seus dados.

O Zabbix é uma ferramenta moderna, de código aberto e multiplataforma, com sistema de monitoramento distribuído, capaz de monitorar a disponibilidade e o desempenho da infraestrutura de uma rede, além de aplicações (HORST; PIRES; DÉO, 2015), possuindo suporte para mecanismos *polling* e *trapping*. A comunicação por *pooling*, é aquela centralizada a partir do servidor Zabbix e executada em intervalos regulares, já o *trapping* são os processos executados localmente nos hosts monitorados e enviados ao servidor Zabbix para processamento (VLADISHEV, 2018).

Possuindo agentes de alta performance nativos, entre as principais estão GNU/Linux, Microsoft® Windows®, IBM® AIX®, HP-UX® e a família BSD, monitoramento sem um agente instalado, ou com agentes SNMP ou IPMI, autenticação segura de usuário utilizando criptografia dos dados, permissões flexíveis de usuários, flexível notificação de eventos predefinidos, execução de comandos remotos, alto nível de visualização de recursos monitorados.

Ele surgiu da iniciativa de Alexei Vladishev de criar uma ferramenta de monitoramento que, diferente do que havia no mercado, não fosse cara, não fosse de difícil manutenção, nem exigisse conhecimentos avançados para utilização. Hoje o Zabbix é uma das soluções de monitoramento mais populares de código aberto (HORST; PIRES; DÉO, 2015).

A ferramenta possui dezenas de módulos, mas as principais funcionalidades são as listadas a seguir:

- Autodescoberta de dispositivos de rede;
- Autodescoberta de recursos do host;
- Descoberta de baixo nível (ou *Low Level Discovery* - LLD);
- Possibilidade de monitoramento distribuído com administração centralizada;
- Aplicação servidor compatível com ambiente GNU/Linux, IBM AIX, HP-UX, AIX, Solaris e família BSD;
- Tradução para vários idiomas;
- Autenticação;
- Auditoria;
- Suporte nativo a SNMP;
- Monitoramento via *Intelligent Plataform Management Interface* (IPMI);
- Monitoramento de aplicações Web;
- Monitoramento de ambientes virtualizados;
- Envio de alertas via e-mail, SMS, *eXtensible Messaging and Presence Protocol* (XMPP) e scripts personalizados.

Um dos recursos mais importantes da ferramenta é o monitoramento da utilização dos recursos, como carga de processamento, quantidade de processos ativos, atividade no disco rígido, utilização da memória virtual e disponibilidade da memória física são alguns de inúmeros parâmetros de sistemas que ele é capaz de monitorar. Mas não apenas a utilização geral dos recursos do ativo, como também individualizar e monitorar cada serviço e seus recursos consumidos. Ele prove informações em tempo real sobre os recursos de um ativo. Além disso, ele pode produzir gráficos de tendências para ajudar na identificação de gargalo no desempenho do sistema (FACHINI, 2010).

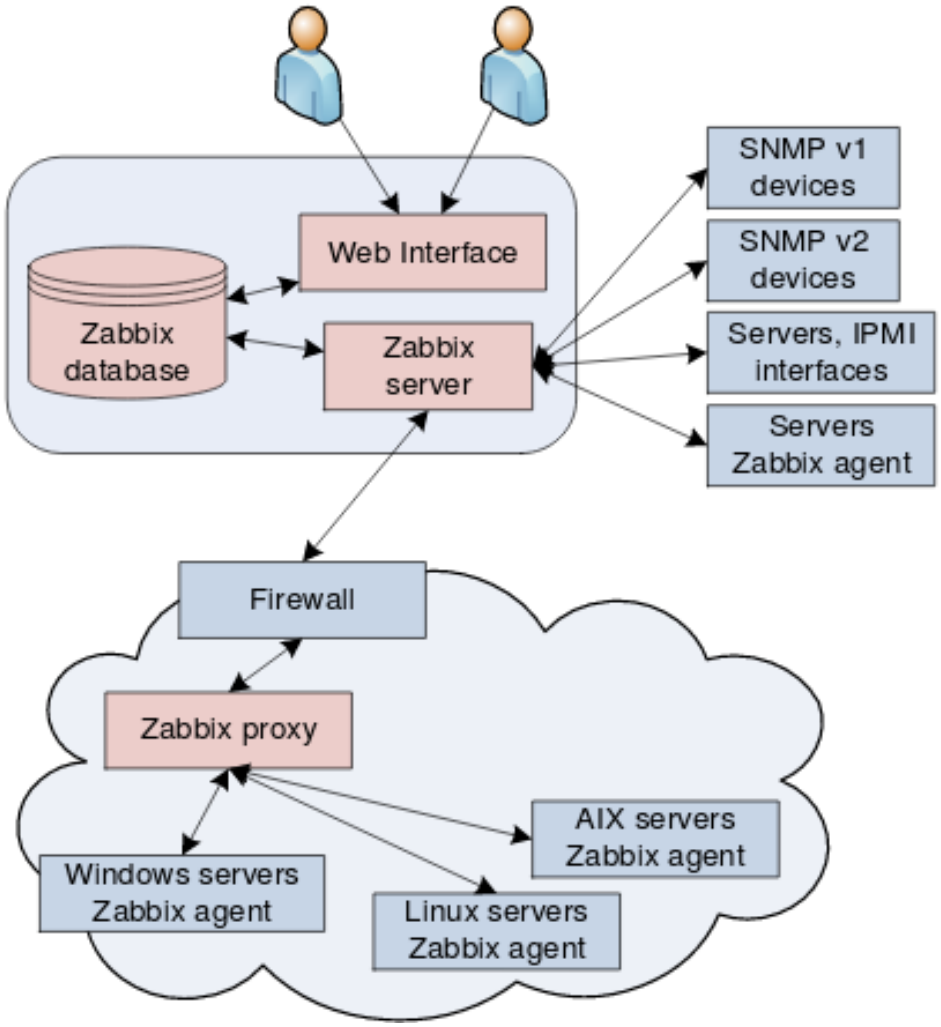
Ele possui também mecanismo de alerta e execução de comandos remotos. Com ele, um administrador pode definir uma possível condição para um gatilho, usando flexíveis expressões. Em algum momento quando essas condições forem verdadeiras (ou falsas), um alerta será enviado por e-mail para um endereço definido pelo administrador e o(s) comando(s) executado(s) no cliente. Programas externos podem ser usados para notificar o usuário como SMS (notificação por

celular) e Jabber (mensagem instantânea). Com a utilização de expressões flexíveis, para os gatilhos, o Zabbix permite que a equipe de TI seja notificada bem antes do estado do sistema alcançar um nível crítico (FACHINI, 2010).

2.3.1 Arquitetura do Zabbix

A arquitetura básica do Zabbix, apresentada na Figura 2, normalmente é instalada em uma única máquina, mas pode haver razões para separar seus elementos. Uma instalação básica do Zabbix contém ao menos o Servidor Zabbix, a Interface Web do Zabbix e o Banco de Dados Zabbix. Mas outros componentes como o Agente Zabbix e o Proxy Zabbix também fazem parte de sua arquitetura.

Figura 2 - Arquitetura e componentes do Zabbix



Fonte: Horst, Pires e Déo (2015).

Ainda na Figura 2, pode-se observar os componentes da arquitetura Zabbix, que apresentam as seguintes finalidades:

- Servidor Zabbix: é o componente central do sistema. É para ele que os agentes enviam as informações do equipamento que está sendo monitorado;
- Banco de dados Zabbix: é onde as informações de monitoramento dos dispositivos são armazenadas. É acessado pelo Servidor Zabbix e pela Interface Web;
- Interface Web Zabbix: é uma aplicação Web que permite acessar e visualizar as informações obtidas dos agentes;
- Agente Zabbix: porção de software que é executada no dispositivo gerenciado e que envia dados para o Servidor Zabbix. O agente acompanha ativamente o consumo de recursos do dispositivo gerenciado;
- Proxy Zabbix: parte opcional do Zabbix que permite distribuir a coleta dos dados de gerenciamento.

2.4 CACTI

O Cacti é uma ferramenta que recolhe e exibe informações sobre o estado de uma rede de computadores através de gráficos. Foi desenvolvido para ser flexível de modo a se adaptar facilmente a diversas necessidades, bem como ser robusto e fácil de usar. Ele monitora o estado de elementos de rede e programas, bem como a largura de banda utilizada e o uso de CPU (COSTA, 2008).

No site oficial da ferramenta, ela é descrita como uma solução gráfica completa de rede, projetada para aproveitar o poder de armazenamento de dados MySQL e funcionalidade gráfica. Cacti proporciona um rápido *poller*, criação de gráfico avançado por *template*, vários métodos de aquisição de dados, gerenciamento de usuários e de possibilidade de adicionar novos recursos. Tudo isso em uma intuitiva interface gráfica, que é fácil de usar e é recomendada para redes simples até redes complexas com centenas de dispositivos (CACTI, 2018).

Costa (2008) ainda complementa dizendo que trata-se de uma interface e uma infraestrutura para o RRDTool, que é responsável por armazenar os dados recolhidos e por gerar os gráficos. As informações são repassadas para a ferramenta através de scripts ou outros programas escolhidos pelo usuário, os quais

devem se encarregar de obter os dados. Pode-se utilizar também o protocolo SNMP para consultar informações em elementos de redes e/ou programas que suportam tal protocolo.

Sua arquitetura aberta prevê a possibilidade de expansão através de *plugins* que adicionam novas funcionalidades. Como o *plugin PHP Network Weathermap* que mostra um mapa da rede e o estado de cada elemento, o *plugin* Monitor que mostra a situação atual dos elementos da rede, com *plugins* também é possível configurar o envio de mensagens de alertas em tempo real para o melhor gerenciamento e controle das ações (CONNER, 2011).

2.5 PLUGIN PHP NETWORK WEATHERMAP

O PHP Network Weathermap surgiu em abril de 2005, e foi desenvolvido para permitir a visualização de links de dados, roteadores, switches, gráficos e demais itens que compõem uma rede de computadores. Isto é possível, com sua instalação como *plugin* de uma ferramenta de monitoramento, ou seja, como um recurso adicional.

Seu desenvolvimento, desde o início, está a cargo do britânico Howard Jones, porém, ele mesmo observa que a ideia não é originalmente sua, utilizou como ponto de partida vários outros projetos com propostas semelhantes (HOWARD, 2010). O principal deles foi o Network Weathermap criado pelo grego Panagiotis Christias. O princípio é o mesmo, ou seja, centralizar em um ou vários mapas todos os dispositivos e principais informações referentes à rede. A ferramenta criada por Christias foi desenvolvida na linguagem de programação PERL, mas o *plugin* utiliza como plataforma o PHP. Howard fez uma boa aposta, pois a linguagem PHP se desenvolveu e se popularizou. Tanto o primeiro quanto o segundo projeto são iniciativas *open source* e, portanto, permitem modificações e melhorias constantes (HOWARD, 2010).

Nesse trabalho o *plugin* é utilizado em conjunto com o Cacti, mas poderia ser qualquer outra ferramenta, desde que se utilize do RRDTool, pois este recurso permite a coleta de dados e seu armazenamento de forma otimizada, além de ser pré-requisito para este uso compartilhado. No caso do Cacti, o PHP Network Weathermap se utiliza desta base de dados, para, por meio de mapas, exibir de

forma sumarizada a situação de toda rede, e o que é melhor, em tempo real. Para o administrador de rede é possível observar, por exemplo, quais links estão ativos, qual o seu nível de utilização, quais equipamentos ou redes estão em operação. Isto possibilita uma visão global de toda a rede, oferecendo a possibilidade de identificar e agir de forma mais rápida e eficaz em relação aos possíveis problemas detectados.

Abaixo são apresentadas algumas das principais características do projeto:

- Projeto estável: É um projeto que se mantém há vários anos, sempre oferecendo atualizações e melhorias, suprindo assim, as demandas apresentadas pela comunidade de software livre;
- Integração: O PHP Network Weathermap possui total compatibilidade com o Cacti, oferecendo aproveitamento de todos os dados coletados e sumarizando em mapas de fácil leitura e configuração. Uma observação a ser feita é que em testes realizados, essa compatibilidade se dá até a versão 0.8 do Cacti, na versão 1.x o Cacti mudou a forma como utiliza os *plugins*, já não sendo possível utilizar;
- Flexibilidade: Permite diversas configurações para mapas, nós, links e demais objetos inseridos. Isto coloca à disposição do usuário diversas possibilidades para exibir e gerenciar sua rede, seja qual for a topologia de rede;
- Editor Gráfico: O *Weathermap Editor* (Web) facilita a configuração tanto de novos mapas quanto a atualização dos já existentes, podendo ser utilizado em conjunto com o editor de texto. Ele permite opções como alterar a aparência de nós e links;
- Compatibilidade: O *PHP Network Weathermap* já foi testado para os seguintes sistemas operacionais: Linux, FreeBSD, Mac OS X e Windows. Além disso, pode ser utilizado em conjunto com outros softwares como o Cacti, como é o caso do MRTG, ou qualquer outro que utilize a ferramenta RRDTool;
- Baixo Custo: Além de possuir licença de Software Livre (Licença Pública Geral GNU, Versão 2), a curva de aprendizagem é pequena, se comparada com ferramentas similares no mercado atual (HOWARD, 2010).

Howard (2010), diz ainda que o PHP Network Weathermap é uma solução atraente para o monitoramento de redes de computadores, pois consegue sintetizar em um ou mais mapas a saúde de toda a rede.

O uso da ferramenta trás várias vantagens, pois possibilita se antecipar ao usuário na localização e solução de falhas, identificar links fora de operação, medir a largura de banda dos links, detectar tráfego suspeito/malicioso, por meio da análise da largura de banda, além de oferecer uma visão geral quanto à localização e abrangência de falhas ou problemas e alertas visuais sobre falhas, através da escala de cores aplicada aos links.

3 DESENVOLVIMENTO

3.1 INSTALANDO E HABILITANDO O SERVIÇO SNMP

Para que o serviço do SNMP funcione é necessário que seus pacotes estejam instalados no servidor de monitoramento e que os dispositivos que serão monitorados estejam configurados para acesso por esse servidor e a comunidade definida nele.

Em sistemas Linux Debian a instalação é feita via o utilitário APT:

```
# apt-get -y install snmp snmpd
```

Sua configuração é feita no arquivo */etc/snmp/snmpd.conf*. Neste arquivo são definidos vários parâmetros, mas os mais importantes são o endereço IP do serviço de SNMP e a comunidade. Após alterar esse arquivo é necessário chamar o *daemon* que reinicia o serviço, através do comando */etc/init.d/snmpd restart*.

Para verificar se o serviço esta rodando, pode-se utilizar o utilitário *netstat* do Linux:

```
# netstat -l|grep snmp
udp 0 0 0.0.0.0:snmp 0.0.0.0:*
```

Os dispositivos que forem monitorados necessitam ter em suas configurações, o apontamento para o servidor SNMP, bem como as portas UDP liberadas. Nos dispositivos da Cisco que foram testados nesse trabalho, a configuração pode ser feita via interface Web do dispositivo, bastando chamar o endereço IP no navegador e indo na opção “SNMP >> Comunnities”.

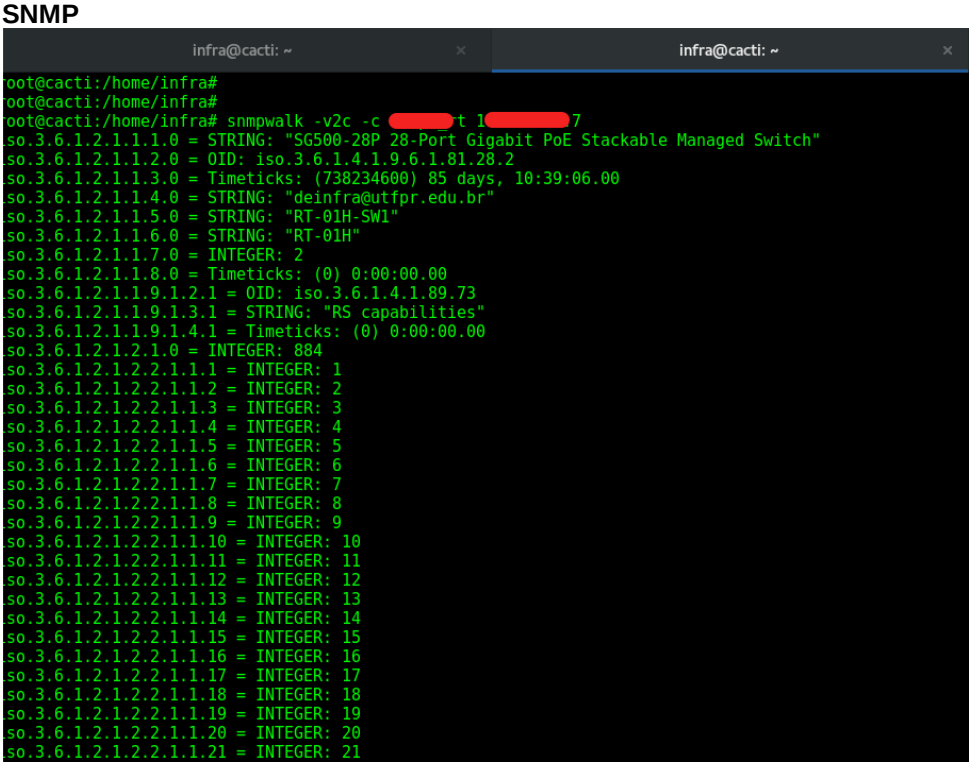
A Figura 3 mostra a configuração em um *switch* SG300 da Cisco.

Figura 3 - Configuração do SNMP em equipamentos Cisco

Fonte: Autoria própria.

Para testar o funcionamento do serviço, existe um utilitário chamado snmpwalk. Esse utilitário é chamado do servidor de monitoramento, indicando a versão do protocolo SNMP que será utilizada e o endereço IP do equipamento. Caso haja sucesso na consulta, a saída é apresentada na Figura 4, caso ocorra algum problema, como não liberação das portas UDP 162 e 163 ou falta de configuração do SNMP no dispositivo, a consulta não retornará nada.

Figura 4 - Consulta snmpwalk, utilizada para testar a comunicação utilizando o protocolo SNMP

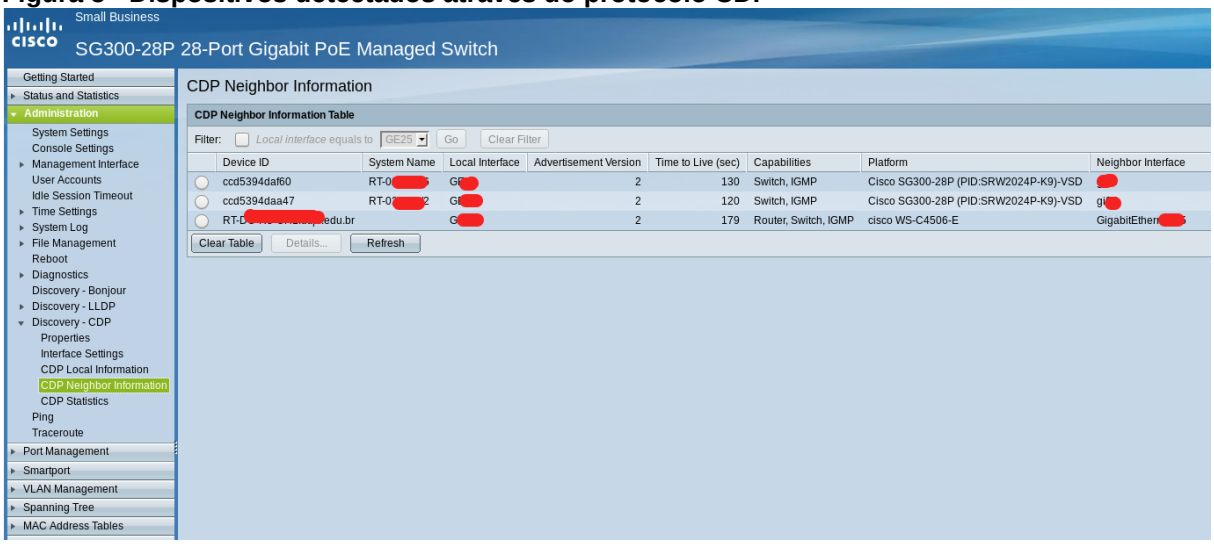


Fonte: Autoria própria.

3.2 CONFIGURAÇÃO DO CDP

A Cisco possui o protocolo proprietário chamado *Cisco Discovery Protocol* (CDP) que já vem habilitado por padrão em suas caixas. Ele auxilia na descoberta de dispositivos vizinhos, exibindo as portas que estão sendo utilizadas nas conexões. Para este trabalho foi bem útil no momento de criar os links entre os dispositivos na confecção de mapas no PHP Network Weathermap. Sua configuração pode ser feita através da CLI da Cisco ou via interface web dos dispositivos. A Figura 5, mostra os equipamentos que estão conectados no *switch* que foi verificado.

Figura 5 - Dispositivos detectados através do protocolo CDP



Fonte: Autoria própria.

3.3 INSTALAÇÃO DO ZABBIX

Os procedimentos abaixo demonstram a instalação e configuração do serviço Zabbix. A instalação foi realizada de acordo com a documentação oficial para o sistema operacional Linux Debian 9 (ZABBIX, 2018).

Foi realizada a instalação da versão 4 (ZABBIX, 2018), última versão estável, utilizando o terminal do Linux para realizar os passos a seguir:

1. Inicialmente é necessário adicionar os repositórios do Zabbix para realizar o download e atualizar o sistema operacional:

```
$ sudo wget
https://repo.zabbix.com/zabbix/3.4/debian/pool/main/zabbix-
release/zabbix-release_3.4-1+stretch_all.deb
$ sudo dpkg -i zabbix-release_3.4-1+stretch_all.deb
$ sudo apt update
```

2. Na sequência é possível instalar o Zabbix e suas dependências (agente, banco de dados, suporte a trabalhar com SNMP):

```
$ sudo apt install mysql-server apache2 build-essential snmp-
mibs-downloader snmp
$ sudo apt install zabbix-server-mysql zabbix-frontend-php
zabbix-agent zabbix-get zabbix-sender
```

3. Criar e configurar banco de dados:

- Conecta no MySQL:

  ```
  $ sudo mysql -u root -p
  ```

- Cria usuário:

  ```
  CREATE USER 'suporte'@'localhost' IDENTIFIED BY 'senha';
  ```

- Cria banco de dados:

  ```
  CREATE DATABASE zabbix CHARACTER SET utf8 COLLATE utf8_bin;
  ```

- Concede privilégios ao usuário criado anteriormente:

  ```
  GRANT ALL PRIVILEGES ON zabbix.* TO 'suporte'@'localhost'
  IDENTIFIED BY 'senha';
  ```

- Sai do MySQL:

  ```
  EXIT
  ```

4. Na sequencia é importado o esquema de tabelas e dados padrões do Zabbix:

```
# cd /usr/share/doc/zabbix-server-mysql
# zcat create.sql.gz | mysql -uroot zabbix
```

5. Edite a configuração do arquivo “zabbix_server.conf” para referenciar a instalação do MySQL que foi feita:

```
# vim /etc/zabbix/zabbix_server.conf
DBHost=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=zabbix
```

A configuração do Apache para a interface web do Zabbix está localizada em </etc/apache2/conf.d/zabbix>. Algumas das configurações do PHP já estão definidas, por exemplo:

```
php_value max_execution_time 300
php_value memory_limit 128M
php_value post_max_size 16M
php_value upload_max_filesize 2M
php_value max_input_time 300
php_value always_populate_raw_post_data -1
# php_value date.timezone America/Sao_Paulo
```

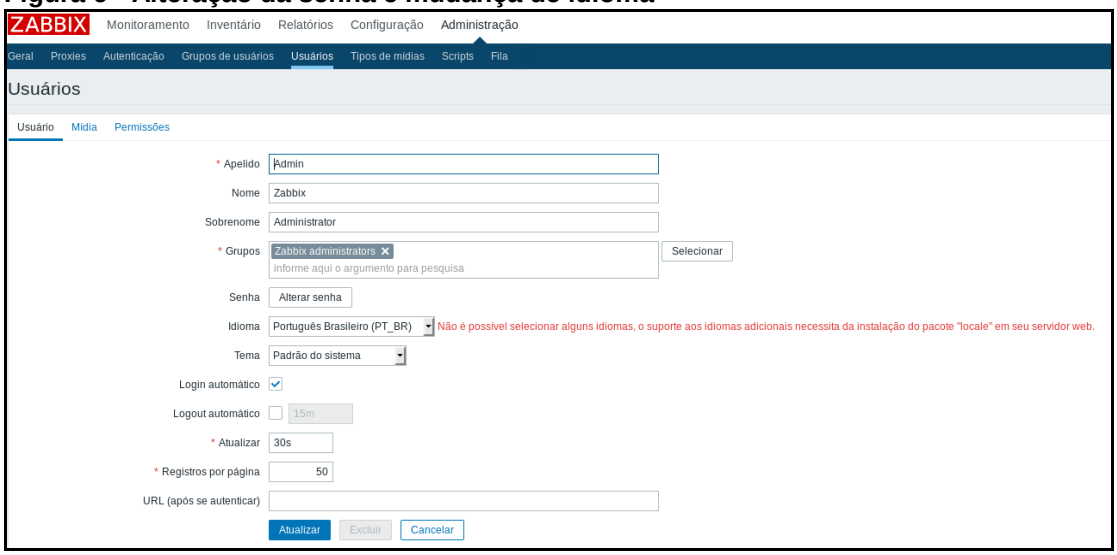
É necessário que se remova o comentário na linha do parâmetro “date.timezone” e que se defina o timezone apropriado (America/Sao_Paulo para a maioria dos estados brasileiros). Após a alteração do arquivo de configuração será necessário o reinício do processo do servidor web (Apache) e do serviço do Zabbix:

```
# service apache2 restart
# service zabbix-server start
```

3.3.1 Procedimentos Pós Instalação

Os passos seguintes são realizados pela interface WEB do Zabbix. Após a instalação a primeira mudança realizada é na mudança da senha padrão para o usuário Admin. Esse procedimento, apresentado na Figura 6, é realizado indo em Administração >> Usuários e selecionando o usuário Admin. Nesta tela é possível também alterar o idioma padrão do Zabbix, que por *default* vem o inglês e o tema utilizado.

Figura 6 - Alteração da senha e mudança de idioma



Fonte: Autoria própria.

3.3.2 Cadastro dos Grupos

Antes de realizar o cadastro dos hosts é interessante criar os grupos para melhor organização dos equipamentos. Neste trabalho foram criados quatro grupos:

1. Datacenter: cadastrar equipamentos do datacenter;
2. Rede sem Fio: cadastro dos AccessPoints da UTFPR;
3. Impressoras: cadastro das impressoras da UTFPR;
4. Switchs/Router: cadastro dos switchs e roteadores em uso.

3.3.3 Templates

Um template é um conjunto de entidades que pode ser associada de forma fácil e conveniente a vários hosts.

As entidades podem ser:

- Itens;
- *Triggers*;
- Gráficos;
- Aplicações;
- Telas (*desde o Zabbix 2.0*);
- Regras de autobusca (LLD) (*desde o Zabbix 2.0*);
- Cenários web (*desde o Zabbix 2.2*).

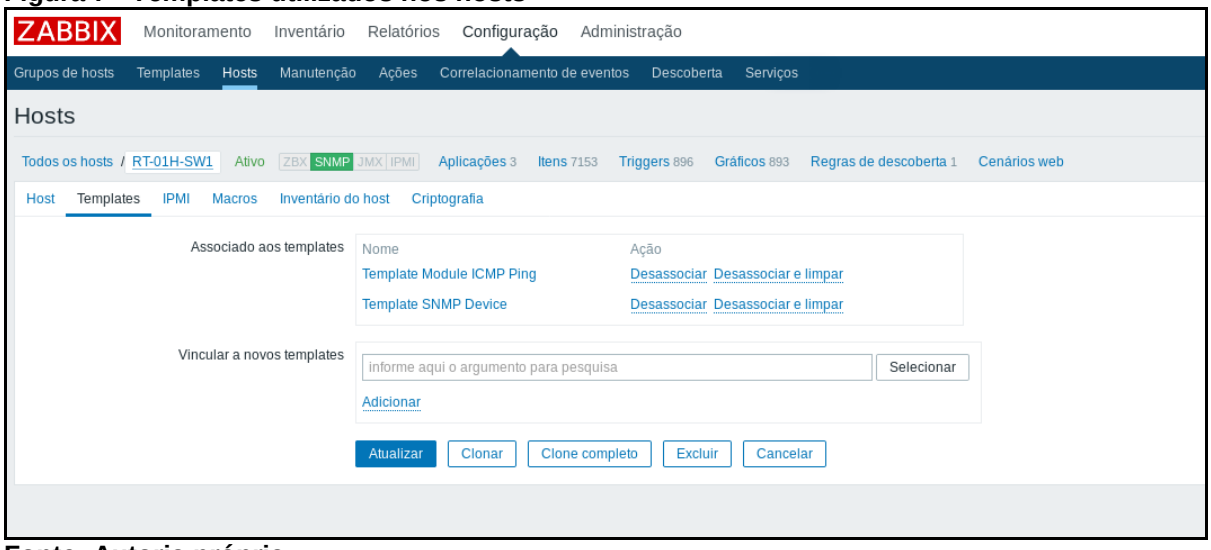
Como na vida real vários hosts são idênticos (sob a ótica de monitoração) ou muito similares, é natural que exista um conjunto de entidades (itens, triggers, gráficos, …) que você vai criar em um host, mas servirá também para vários outros. É claro que você pode copiar as entidades entre os hosts, mas isso gera grande trabalho manual. Com o uso de *templates* tal processo é simplificado ao simplesmente associar um host a um *template*, com isso o Zabbix já irá copiar todo o perfil de monitoração necessário para o host (ZABBIX, 2018).

Os *templates* também podem ser usados (e normalmente o são) para agrupar conjuntos comuns de monitoração para aplicações ou serviços específicos (tal qual o Apache, MySQL, PostgreSQL, Postfix, entre outros) e são associados de forma cumulativa nos hosts.

Outro benefício do uso de *templates* é que se for necessária a modificação de um determinado perfil de monitoração (por exemplo adicionar uma nova métrica de monitoração em todos os servidores Apache) isso poderá ser feito no nível do *template* que todos os hosts associados serão alterados em conjunto.

Nesse trabalho foram utilizados *templates* para testes de ICMP e conexão SNMP (Figura 7). Em alguns equipamentos foram utilizados também um *template* da Cisco, chamado Cisco Total, que trás diversas informações sobre equipamentos Cisco.

Figura 7 - Templates utilizados nos hosts



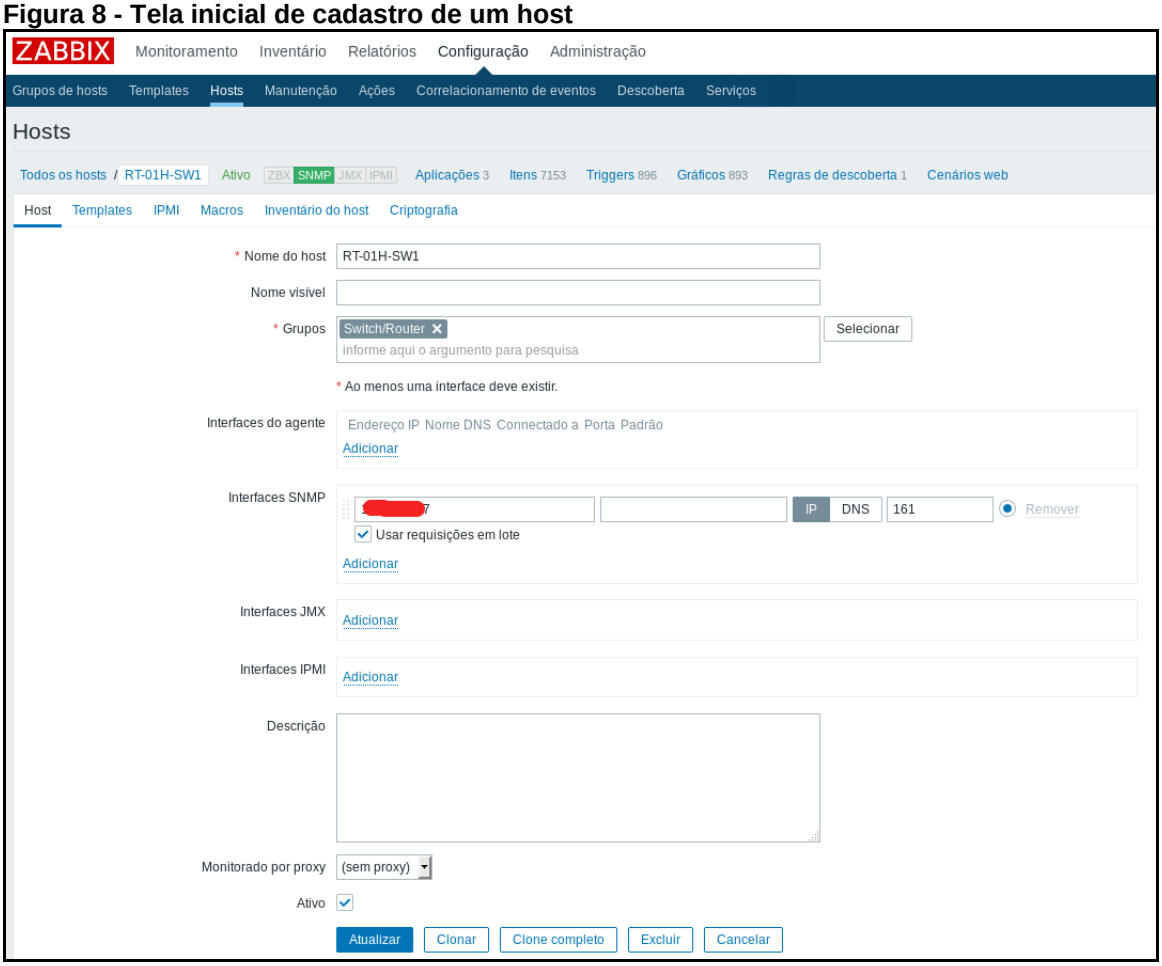
Fonte: Autoria própria.

3.3.4 Cadastro dos Hosts

Um host no Zabbix é uma entidade na rede (física ou virtual) que você deseja monitorar. A definição do que é um “host” é muito flexível. Pode ser um servidor real, um switch, uma máquina virtual ou, até mesmo, uma aplicação ou serviço. Podemos resumir a definição de host como sendo qualquer elemento de sua rede que possua um IP e tenha capacidade de comunicação com o Zabbix, seja através de coletas ativas ou passivas.

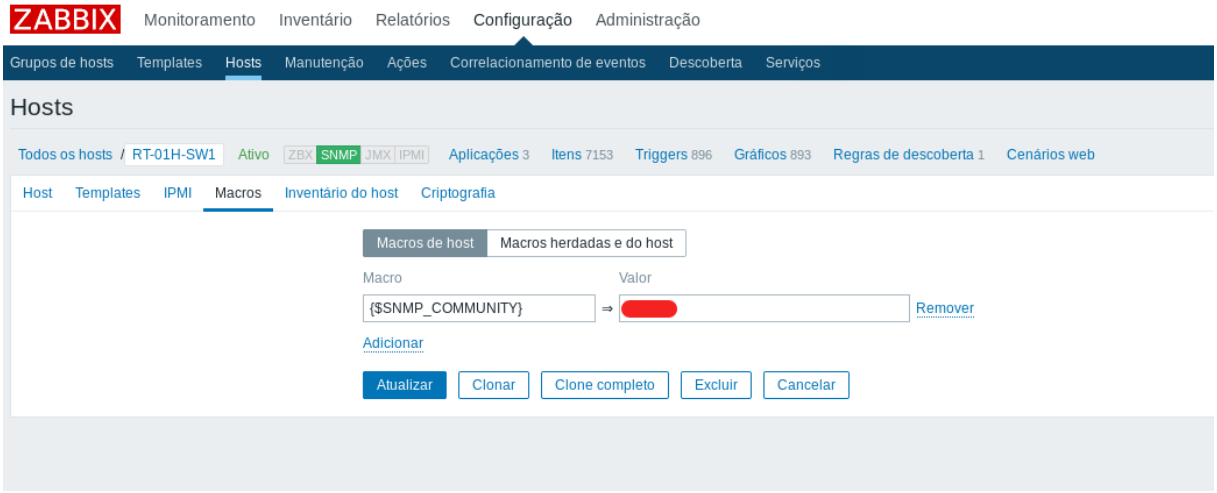
Criar os hosts é uma das primeiras tarefas de monitoração no Zabbix. Por exemplo, se você deseja monitorar alguns parâmetros em um servidor “X”, você precisa primeiramente criar um host chamado, digamos: “Servidor X” e adicionar itens a serem monitorados nele. Os hosts são organizados dentro de grupos de hosts e todo host deverá participar de, no mínimo, um grupo. As etapas de cadastro de hosts são exibidas nas Figuras 8 (Tela inicial), 9 (Cadastro) e 10 (Listagem). Os nomes e endereços IP foram ocultados por questões de segurança.

Figura 8 - Tela inicial de cadastro de um host



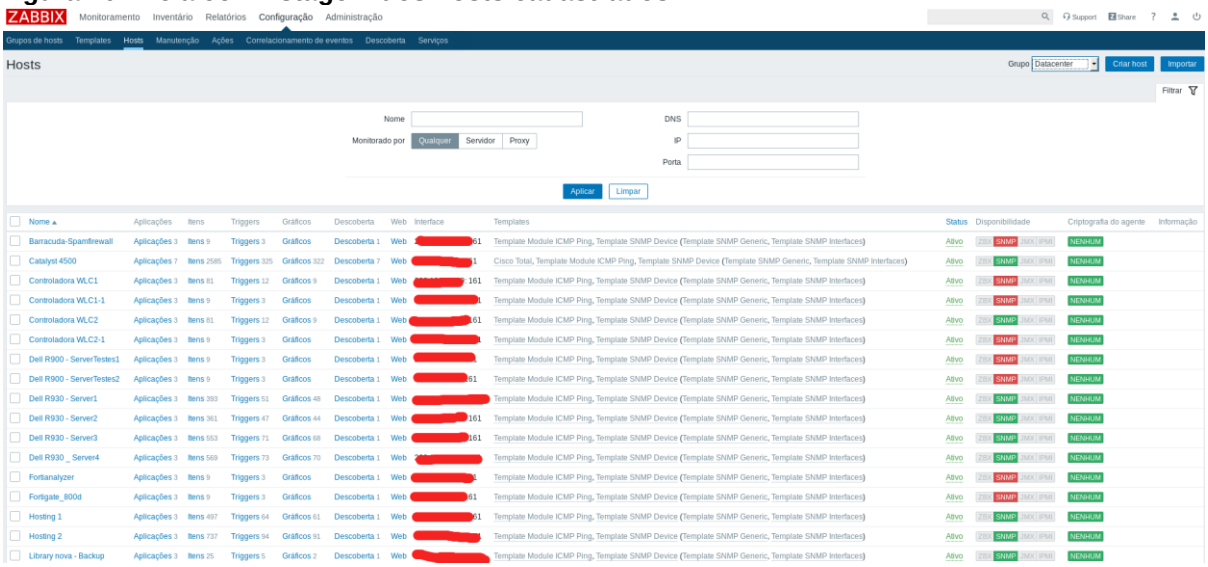
Fonte: Autoria própria.

Figura 9 - Cadastro da comunity SNMP



Fonte: Autoria própria.

Figura 10 - Tela com listagem dos hosts cadastrados



Fonte: Autoria própria.

3.4 INSTALAÇÃO DO CACTI

A versão utilizada neste trabalho foi a 0.8.8, onde o plugin PHP Network Weathermap funciona corretamente.

A primeira ação é atualizar os repositórios do Debian e instalar os pacotes necessários, os comandos necessitam ser realizar com o usuário root.

- Atualizando o repositório de pacotes:

```
# apt-get update
```

Na sequência é necessário instalar os pacotes e ferramentas necessárias para execução do Cacti, como compiladores, servidores Web e de banco de dados. No nosso caso, tanto o servidor Web Apache2 como o MySQL já estão instalados, quando foi feita a instalação do Zabbix, nesse caso, o Linux irá pular esses pacotes quando executado o comando de instalação.

- Instalando os pacotes necessários:

```
# apt-get install gcc make apache2 mysql-server
libmysqlclient-dev libperl-dev php5 php5-mysql snmp snmpd
libsnmp-dev libsnmp-base libnet-snmp-perl rrdtool openssl
```

O Cacti utiliza o MySQL como banco de dados, nesse trabalho será utilizado o mesmo banco que foi instalado o Zabbix, sendo necessário nessa etapa apenas conectar no banco, criar a base de dados e usuário para uso.

- Criando o banco de dados e usuário para o Cacti no MySQL:

```
# mysql_install_db
# mysql -u root -p
Enter password: <digite a senha de root que foi definida na
instalação do MySQL>
mysql> create database cacti character set utf8;
```

- Criar o usuário cactiuser para o banco cactiuser:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO cactiuser@localhost
IDENTIFIED BY 'senha_cactiuser' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)
mysql> quit
```

Na sequencia é necessário criar um usuário no sistema operacional para realizar as configurações do Cacti:

```
# useradd cactiuser -s /bin/false
# passwd cactiuser
Digite a nova senha UNIX:
Redigite a nova senha UNIX:
passwd: senha atualizada com sucesso
```

O próximo passo é realizar o download do Cacti e realizar sua instalação. Após instalado seus arquivos são copiados para o diretório do servidor web Apache2.

```
# wget -c "http://www.cacti.net/downloads/cacti-0.8.8f.tar.gz"
# tar -xzvf cacti-0.8.8f.tar.gz -C /var/www/html
# mv cacti-0.8.8f.tar.gz cacti
# cd /var/www/html/cacti
```

Na próxima etapa, abra o arquivo de configuração do Cacti e atualize as seguintes linhas para refletir o nome de usuário, a senha, o host e o caminho do banco de dados do MySQL, conforme ilustrado abaixo:

```
# nano /var/www/html/include/config.php
$database_type     = 'mysql';
$database_default  = 'cacti';
$database_hostname = 'localhost';
$database_username = 'cactiuser';
$database_password = 'senha_cactiuser';
$database_port     = '3306';
$database_ssl      = false;
```

Em seguida, você precisa preencher o banco de dados cacti carregando o script cacti.sql localizado no local raiz do documento do seu servidor web e verificar as tabelas do banco emitindo os comandos abaixo:

```
# mysql -u cacti_user cacti -p < /var/www/html/cacti.sql
# mysql -u cacti_user cacti -p -e 'show tables'
```

Antes de iniciar a instalação do Cacti a partir da interface web, execute os comandos abaixo para remover o arquivo “index.html” padrão instalado pelo servidor web, crie um arquivo de log para o Cacti e conceda as permissões de gravação completas para o caminho de instalação do Cacti.

```
# rm /var/www/html/index.html
# touch /var/www/html/log/cacti.log
# chown -R cactiuser:cactiuser /var/www/html/
```

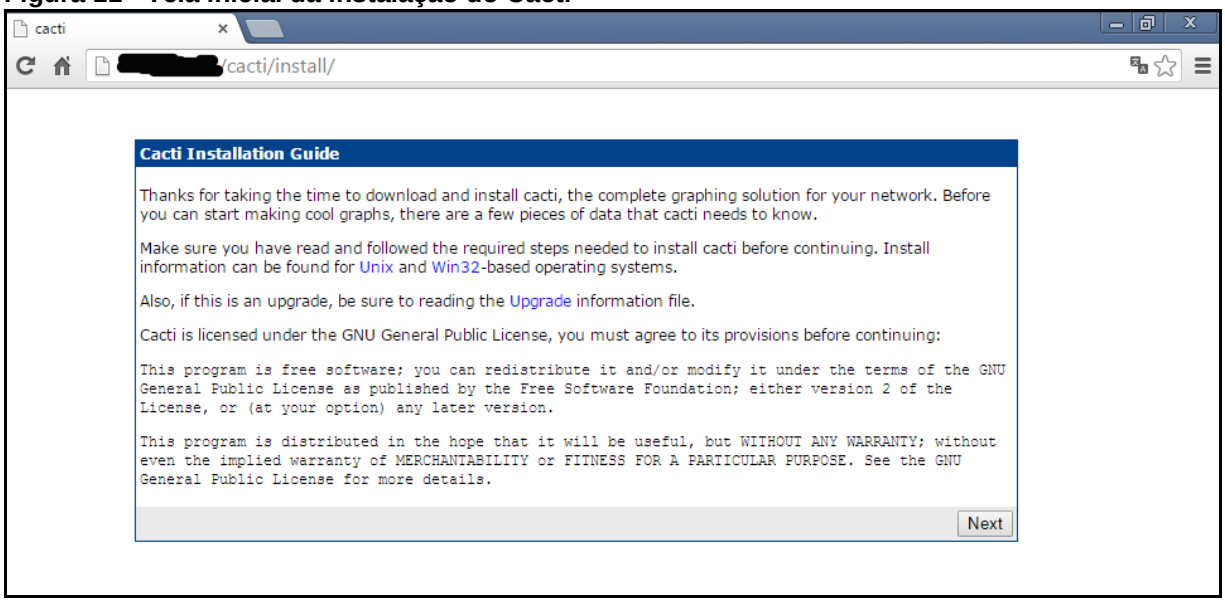
Reinicie os serviços do MySQL e Apache:

```
# /etc/init.d/mysql restart
[ ok ] Stopping MySQL database server: mysqld.
[ ok ] Starting MySQL database server: mysqld
# /etc/init.d/apache2 restart
[ ok ] Restarting web server: apache2
```

Os passos seguintes das instalação são realizados pela interface Web, acessando o endereço de instalação do servidor é possível acessar a tela inicial de instalação do Cacti.

Acessar no navegador o endereço: <http://IPSERVIDOR/cacti>. Será apresentada a tela inicial de instalação do Cacti (Figura 11). Nessa primeira tela é apresentada a licensa de uso da ferramenta.

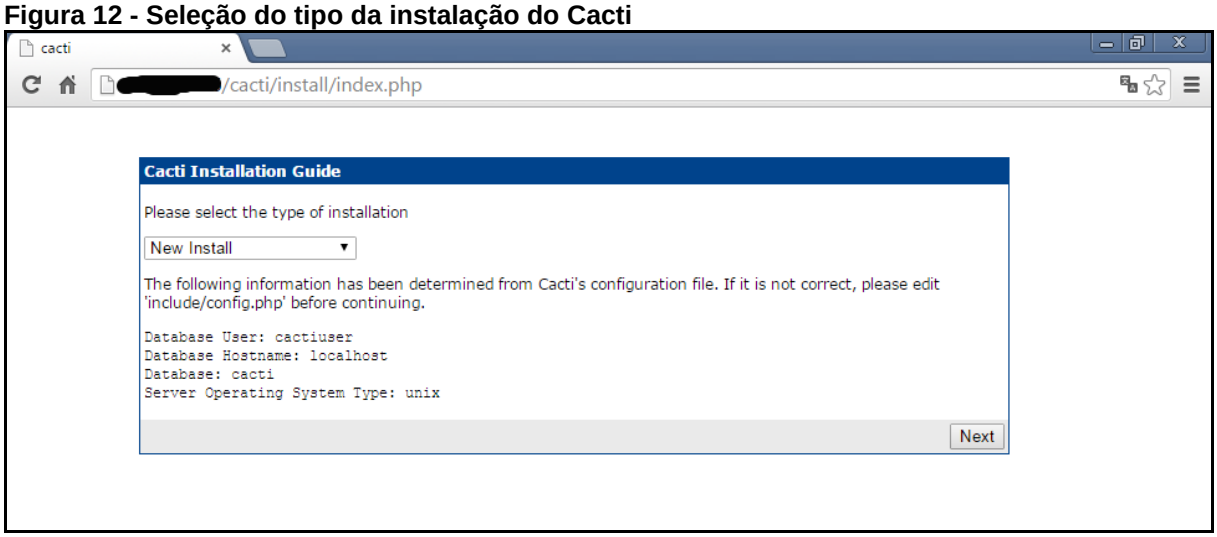
Figura 11 - Tela inicial da instalação do Cacti



Fonte: Autoria própria.

Selecionar o tipo da instalação e clicar no botão “Next” (Figura 12).

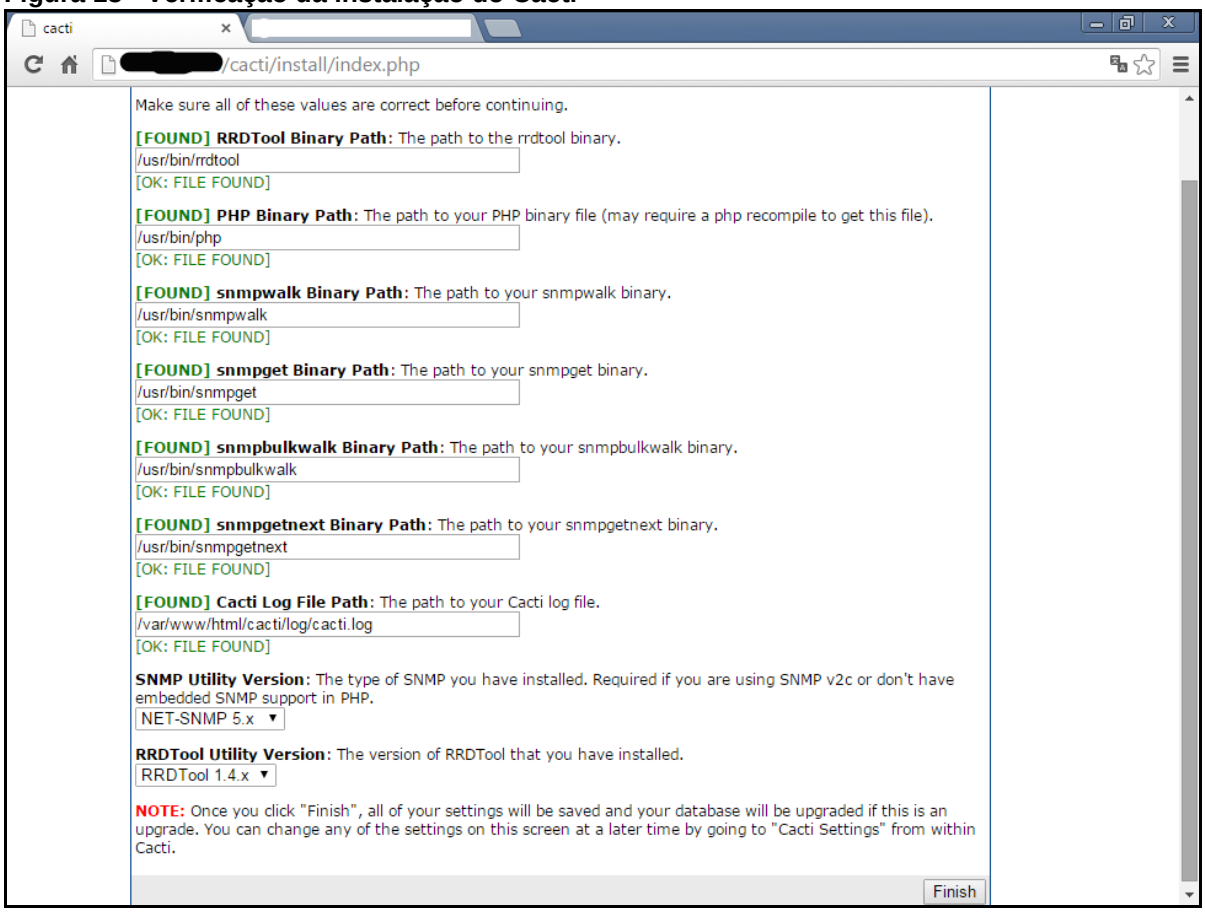
Figura 12 - Seleção do tipo da instalação do Cacti



Fonte: Autoria própria.

Na sequência é apresentada uma tela (Figura 13) indicando se foram encontrados os arquivos binários, deve estar [FOUND] na cor verde, caso esteja [NOT FOUND] na cor vermelha, provavelmente faltou instalar algum pacote pré-requisito.

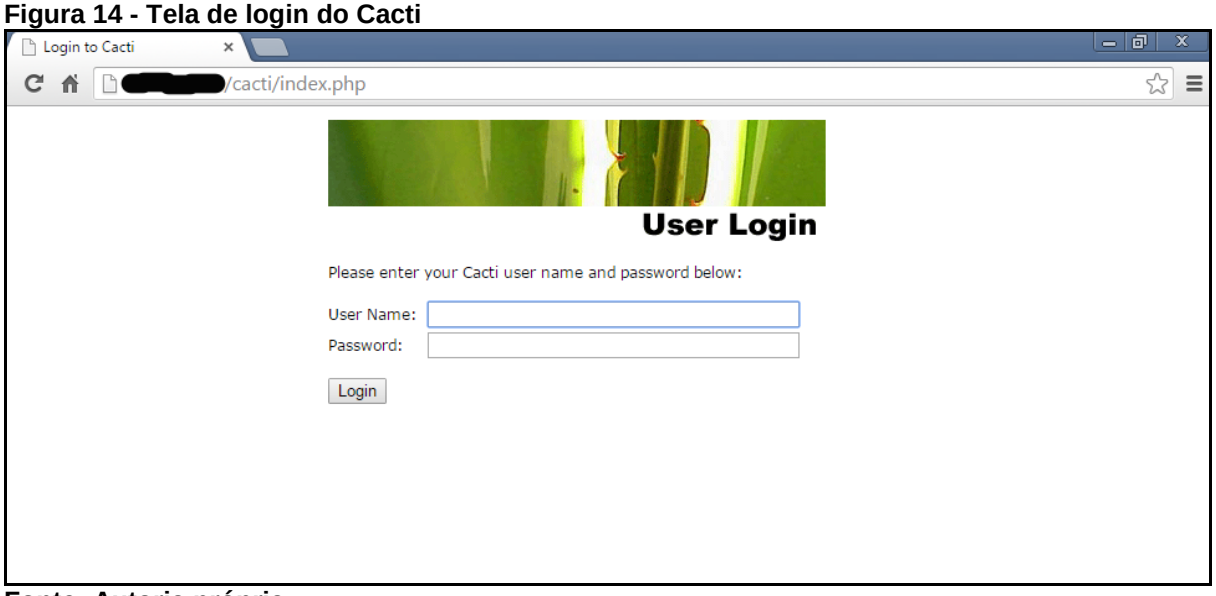
Figura 13 - Verificação da instalação do Cacti



Fonte: Autoria própria.

Estando tudo certo, clicar no botão “Finish” e na sequência será redirecionado para a tela de *login* (Figura 14), utilize o usuário *admin* e a senha *admin* e clique no botão “Login”.

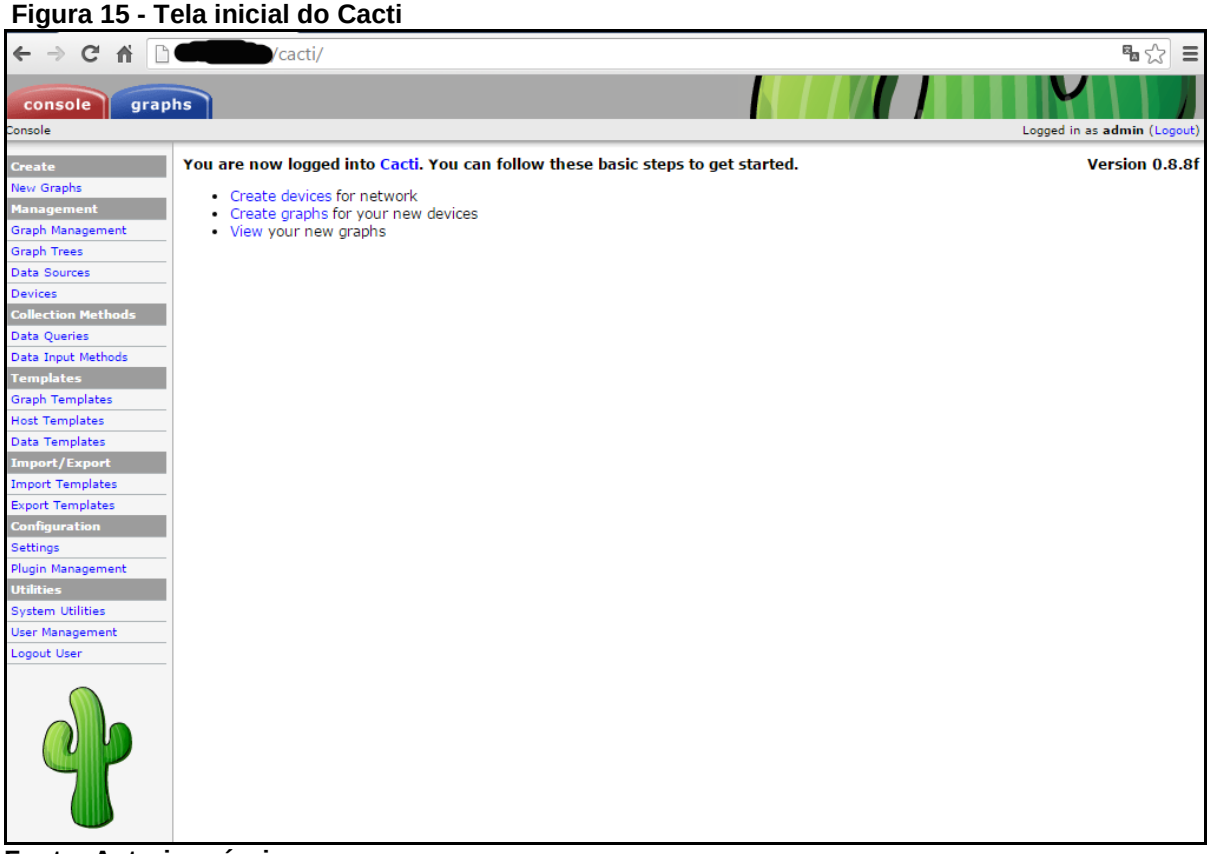
Figura 14 - Tela de login do Cacti



Fonte: Autoria própria.

Após realizar *login*, será apresentada a tela da Figura 15, onde é possível já cadastrar os hosts, gráficos e configurar a ferramenta para utilizar o plugin PHP Network Weathermap.

Figura 15 - Tela inicial do Cacti

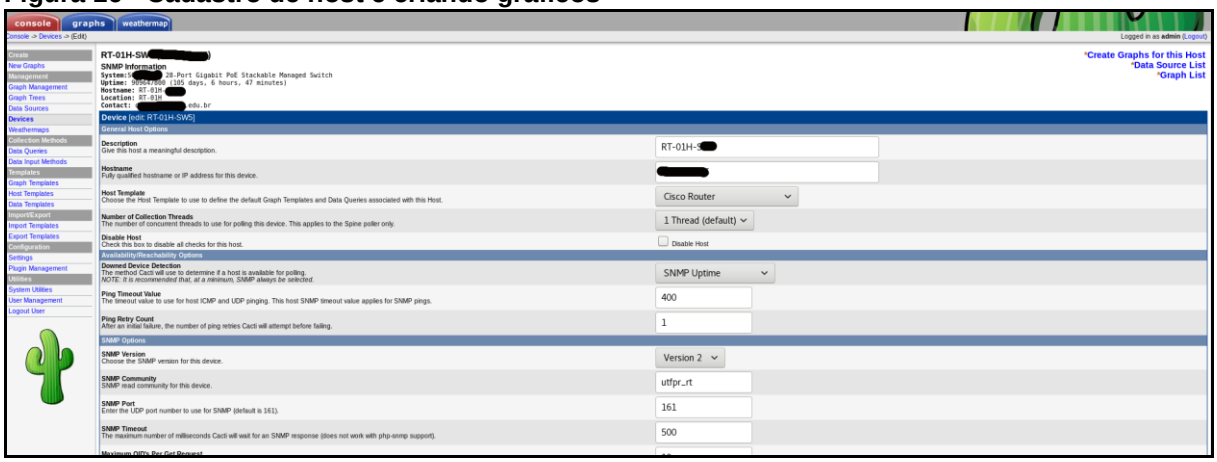


Fonte: Autoria própria.

3.4.1 Cadastrando Hosts e Gerando Mapas

Para cadastrar um novo equipamento para monitoramento, basta escolher a opção disponível em “Console > Management > Devices” e usar a opção “Add” (Figura 16). Será apresentada uma tela onde é informado o nome do host, endereço IP, versão do SNMP e comunidade. Após cadastrar o host, na parte superior da tela, vai ter uma opção para gerar os gráficos para este host. Indo nesta opção, basta indicar os tipos de gráficos que se deseja, e o Cacti irá gerar e começar a realizar a coleta dos dados por SNMP.

Figura 16 - Cadastro de host e criando gráficos



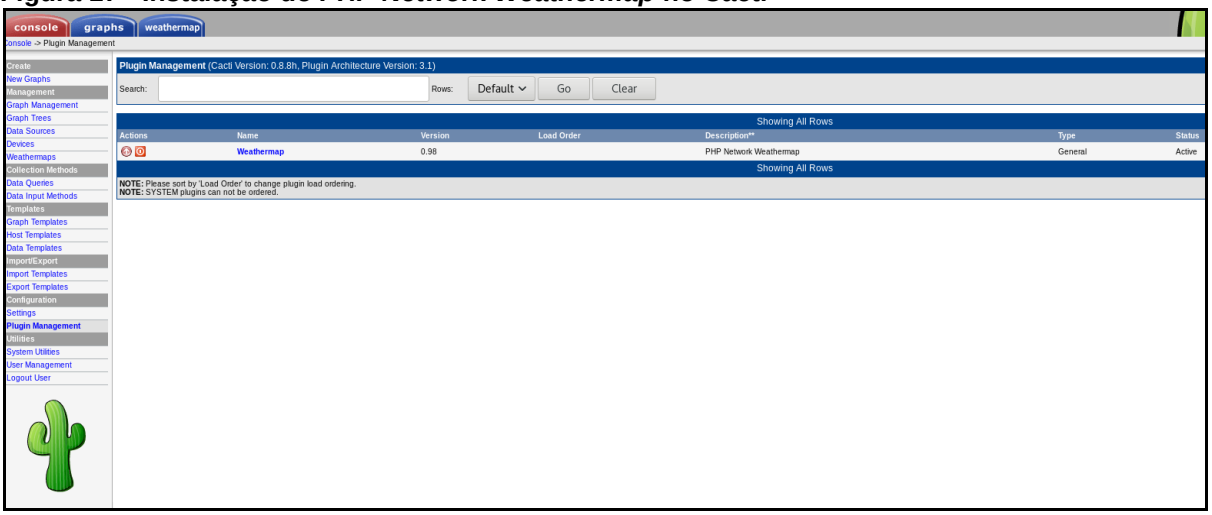
Fonte: Autoria própria.

3.5 INSTALAÇÃO DO PLUGIN PHP NETWORK WEATHERMAP

Os procedimentos de instalação e documentação estão disponível em <https://network-weathermap.com/manual/0.97b/pages/main.html#installation> (acesso em: 11 nov. 2018).

Sua instalação consiste em descompactar o arquivo *zip* do projeto e copiá-lo para o diretório de *plugins* do Cacti. Após isso, selecionando em “Console >> Configuration >> Plugin Managment” no Cacti, o plugin estará disponível para ser habilitado (Figura 17).

Figura 17 - Instalação do *PHP Network Weathermap* no Cacti



Fonte: Autoria própria.

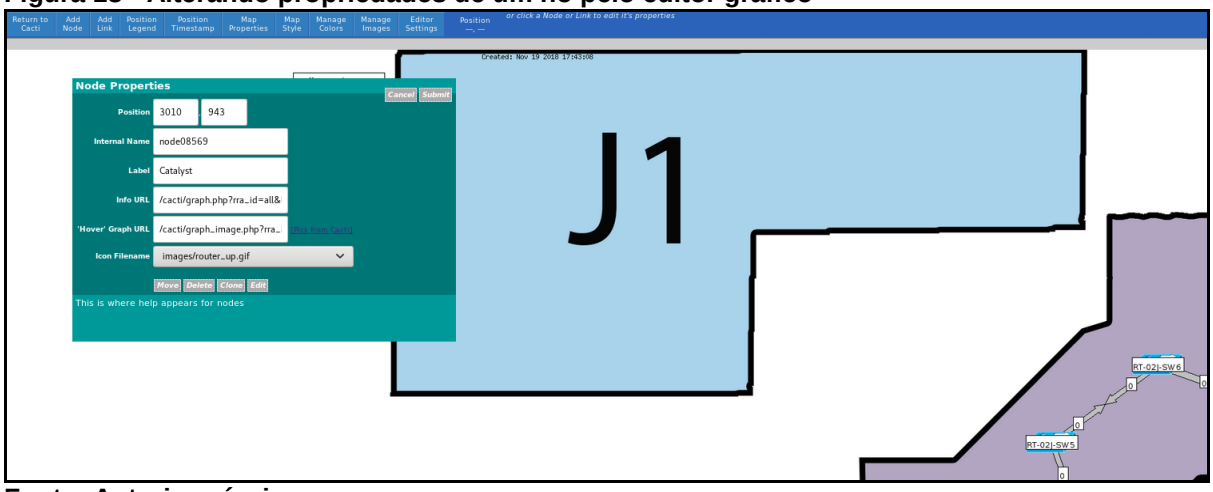
Após habilitado, irá aparecer uma aba “weathermap” no Cacti, onde é possível ver os mapas disponíveis.

3.5.1 Editor Gráfico

Por meio do editor gráfico (Figura 18) é possível criar novos mapas e personalizar os existentes. Ele permite as seguintes operações:

- Adicionar novos nós;
- Adicionar links;
- Alterar posicionamento da legenda do mapa (a legenda exibe o percentual de utilização dos links, através de escala de cores);
- Alterar a posição do *timestamp* (data e hora da última atualização);
- Alterar as propriedades do mapa. Elas englobam parâmetros como:
- Título do mapa;
- Texto da legenda;
- Texto do *timestamp*;
- Tamanho em pixels com que será desenhado o link;
- Largura de banda do link (em bps – bits por segundo);
- Dimensões da imagem de fundo do mapa;
- Campo para fazer link com a imagem escolhida para fundo do mapa.
- Estilo do mapa, tendo como opções:
  - Rótulo dos links (Permite alterar a unidade de exibição da banda do link, entre: bps, percentual de utilização ou nenhum);
  - Estilo do HTML, *overlib* (dinâmico) ou estático;
  - Estilo para as setas dos links;
  - Opções de tipos e tamanhos de fontes para os nós, para os rótulos dos links e para a legenda.

Figura 18 - Alterando propriedades de um nó pelo editor gráfico



Fonte: Autoria própria.

O *plugin* gera um arquivo *conf* para cada mapa que é criado, onde é possível alterar o valor das propriedades, editando esse arquivo. Todos os nodes, sua posição, ícone e outras propriedades aparecem nesse arquivo. A imagem apresentada na Figura 19 mostra partes deste arquivo.

Figura 19 - Alterando propriedades do mapa pelo arquivo de configuração

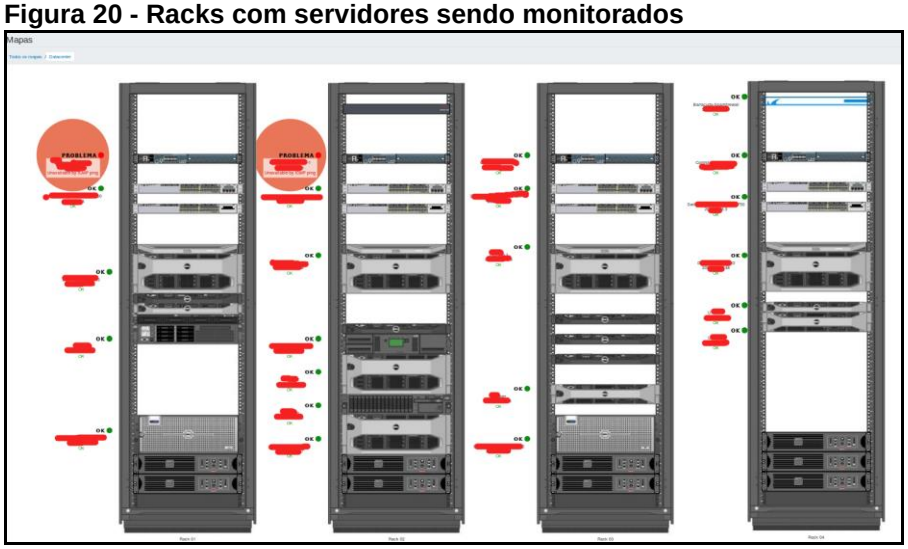
```
# TEMPLATE-only NODEs:
NODE DEFAULT
        LABELFONT 100
        MAXVALUE 100


# TEMPLATE-only LINKs:
LINK DEFAULT
        WIDTH 5
        BWFONT 100
        COMMENTFONT 100
        BWLABEL bits
        BANDWIDTH 1000M


# regular NODEs:
NODE node08569
        LABEL Catalyst
        INFOURL /cacti/graph.php?rra_id=all&local_graph_id=485
        OVERLIBGRAPH /cacti/graph_image.php?rra_id=0&graph_nolegend=true&graph_height=100&graph_width=300&local_graph_id=485
        ICON images/router_up.gif
        POSITION 3010 943

NODE RT-01J-SW1
        LABEL RT-01J-SW1
        ICON images/switch_up.gif
        POSITION 1542 1206

NODE RT-01J-SW2
        LABEL RT-01J-SW2
        ICON images/switch_up.gif
        POSITION 2285 1241

NODE RT-03J-SW1
        LABEL RT-03J-SW1
        ICON images/switch_up.gif
        POSITION 1691 1016

NODE RT-03J-SW1_copy
        LABEL RT-03J-SW1
        ICON images/switch_up.gif
        POSITION 1689 941

NODE RT-01J-SW3
        LABEL RT-01J-SW3
        ICON images/switch_up.gif
        POSITION 2276 1444
```

Fonte: Autoria própria.

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

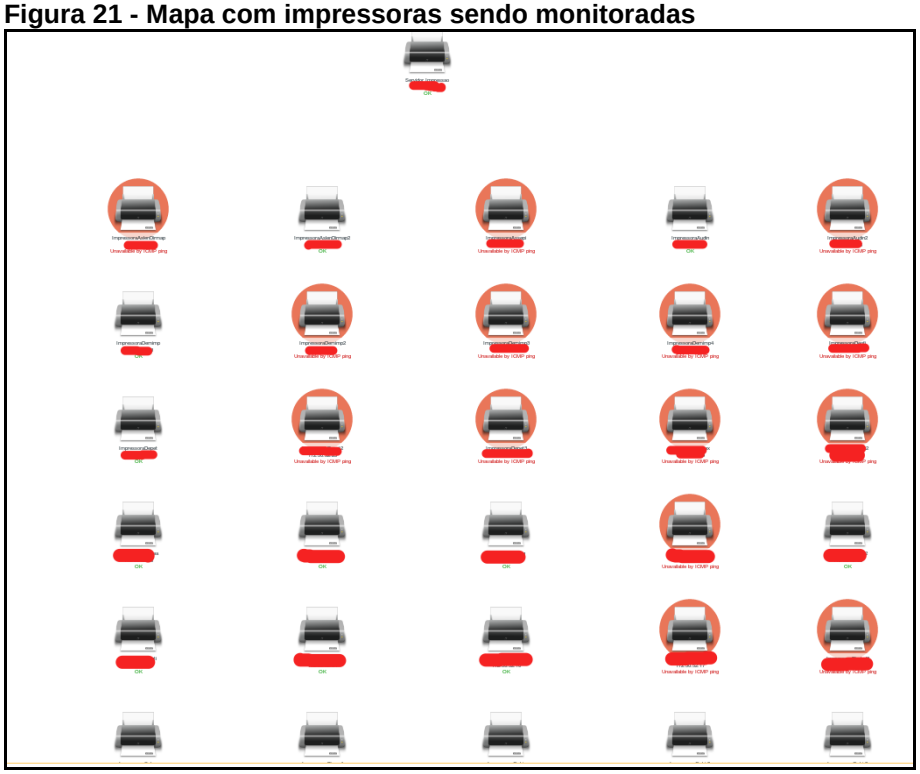
Após realização dos castrados e configuração dos hosts, foram desenvolvidos mapas para atender as necessidades de monitoramento. As Figuras 20 (Racks), 21 (Mapa Impressoras) e 22 (Monitoramento) apresentam o resultado final, com os mapas que foram desenvolvidos, contemplando os equipamentos que necessitam de monitoramento.

Figura 20 - Racks com servidores sendo monitorados



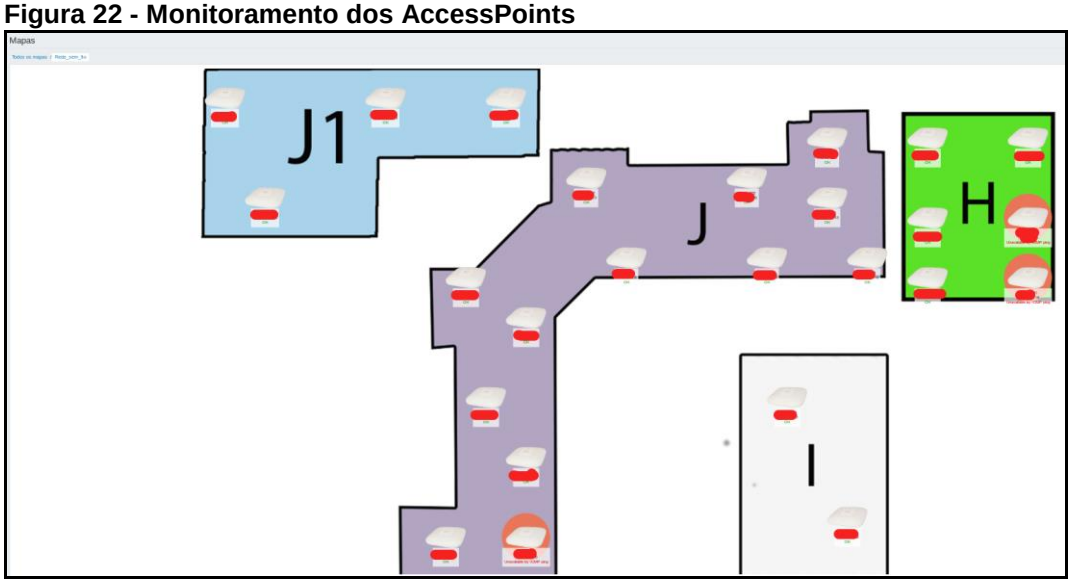
Fonte: Autoria própria.

Figura 21 - Mapa com impressoras sendo monitoradas



Fonte: Autoria própria.

Figura 22 - Monitoramento dos AccessPoints

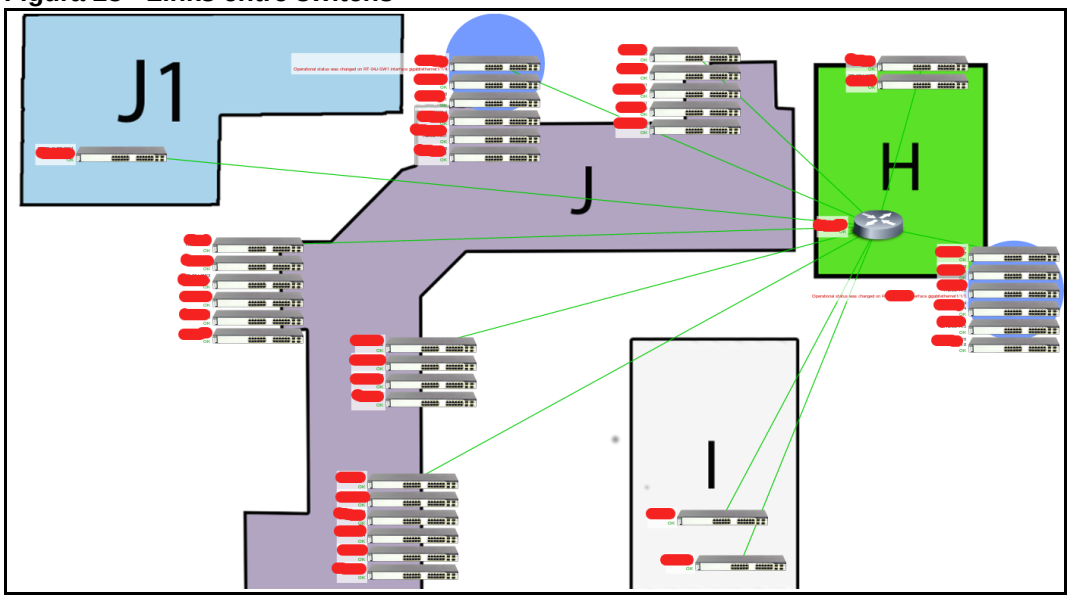


Fonte: Autoria própria.

Ao todo foram cadastrados 182 equipamentos, contemplando *AccessPoints*, *switchs*, reteadores, impressoras e servidores do *datacenter*. Por questões de segurança os nomes e endereços IP foram ocultados.

Nos mapas (Figura 23) é exibido na frente de cada host seu nome e endereço IP configurado. Caso o Zabbix não consiga comunicação com o host, irá gerar um círculo vermelho acusando a falta de comunicação.

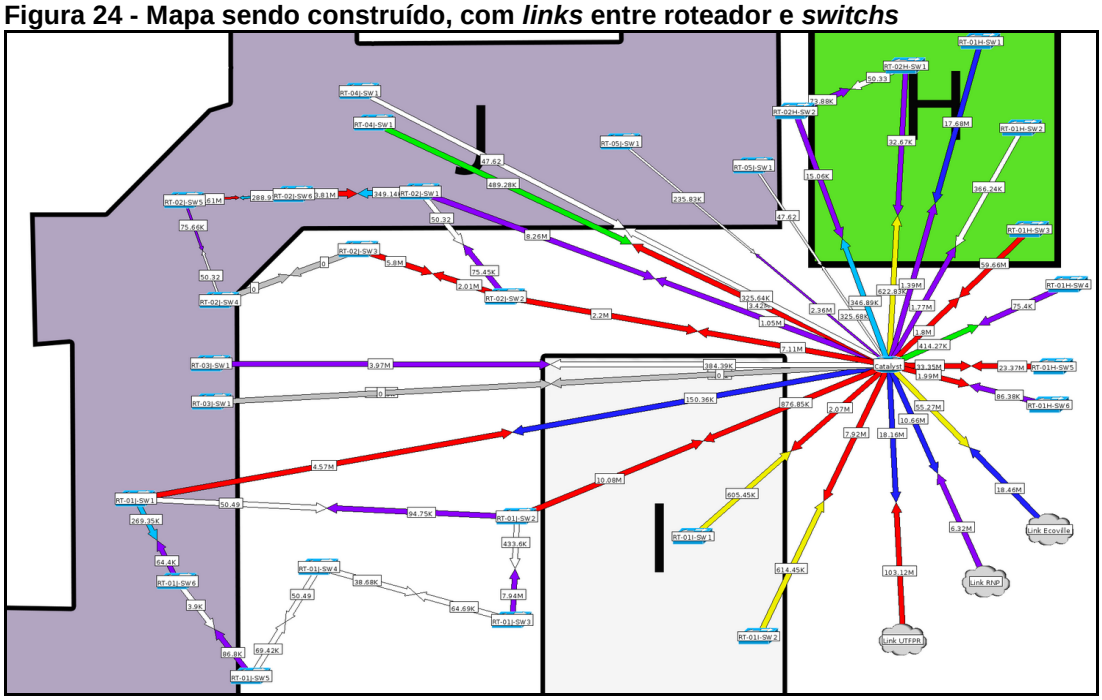
Figura 23 - Links entre switchs



Fonte: Autoria própria.

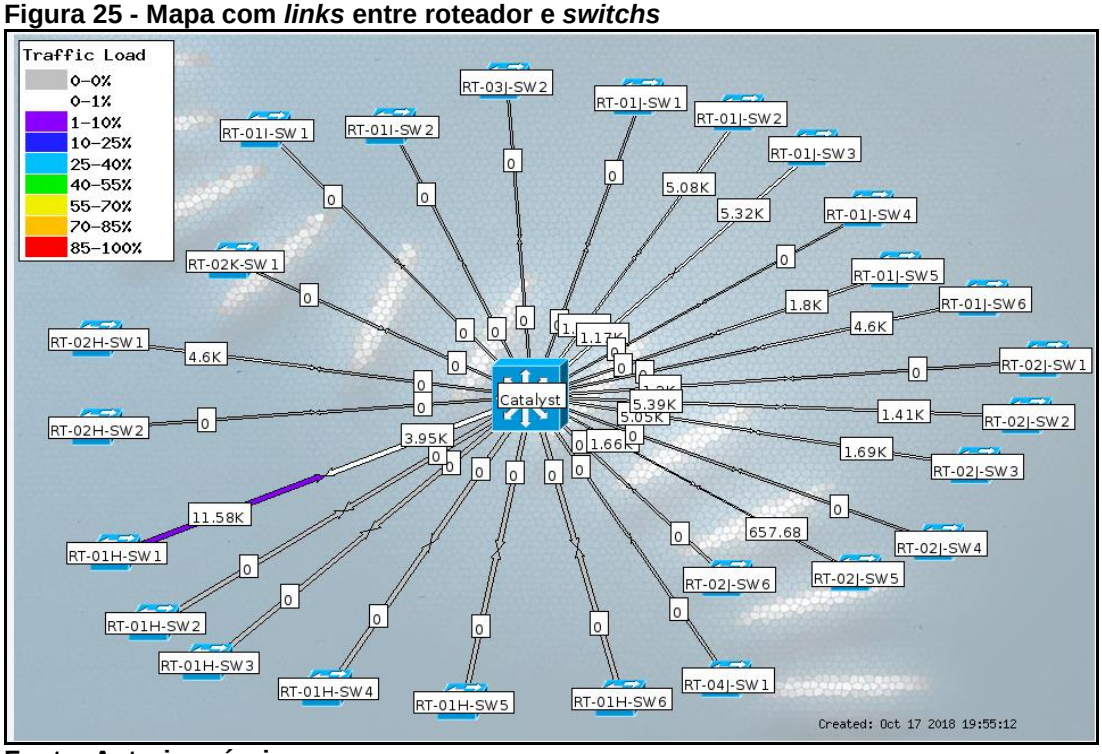
Foi desenvolvido um mapa que mostra os links entre os *switchs* nos blocos que gerenciam e o trafego que geram esses links (Figura 24 e Figura 25). O *plugin* possui uma legenda em formato de paleta de cores, onde é indicado a utilização dos links.

Figura 24 - Mapa sendo construído, com *links* entre roteador e *switchs*



Fonte: Autoria própria.

Figura 25 - Mapa com *links* entre roteador e *switchs*



Fonte: Autoria própria.

5 CONCLUSÃO

Para a melhor gestão de qualquer rede de computadores é necessário o uso de certas de ferramentas, onde seja possível verificar erros, falhas, status e qualquer tipo de ação executada nos *hosts*, que possibilite um melhor planejamento para aumentar o tempo de disponibilidade dos recursos. Neste trabalho, foram apresentadas e configuradas três ferramentas *open-source* para esta finalidade. A primeira foi o Zabbix, popular ferramenta de monitoramento, que possui diversos recursos. Com ela foram desenvolvidos vários mapas de monitoramento, como dos equipamentos do *datacenter*, *AccesPoints* e impressoras distribuídas no campus e *links* entre os *switchs* de borda. Com o Cacti, foram gerados vários gráficos para monitorar o uso dos *links* de dados e utilizado junto a ele, o *plugin PHP Network Wathermap*, que consegue trazer em tempo real o uso de *link* de dados entre os equipamentos monitorados. Tais ferramentas cumprem bem o papel a qual são definidas, trazendo uma visão mais ampla e atual da situação dos equipamentos que estão sendo monitorados.

REFERÊNCIAS

CACTI. **Software Cacti**. Copyright© 2004-2018, The Cacti Group, Inc., 2018. Disponível em: <http://www.cacti.net>. Acesso em: 15 out. 2018.

CASE, J. D. et al. **Simple network management protocol (SNMP)**. Network Working Group, mai. 1990, p.36. Disponível em: <https://www.rfc-editor.org/rfc/pdfrfc/rfc1157.txt.pdf>. Acesso em: 24 out. 2018.

CONNER, Jimmy. **CactiUser: Plugins**. Copyright© 2009-2011, The Cacti Group, 2011. Disponível em: <https://docs.cacti.net/plugins>. Acesso em: 15 nov. 2018.

COSTA. Felipe. **Ambiente de rede monitorado com Nagios e Cacti**. 1. ed. Rio de Janeiro: Ciência Moderna, 2008.

FACHINI, Thiago. **Implementação da ferramenta Zabbix para monitoramento reativo**. Universidade Luterana do Brasil ULBRA, Tec. Rede de Computação, Canoas, nov. 2010. Disponível em: <http://www.ulbra.inf.br/joomla/images/documentos/TCCs/2010_2/redes-thiago%20fachini.pdf>. Acesso em: 29 out. 2018.

FRYE, R. et al. **Coexistence between version 1, version 2, and version 3 of the internet-standard network management framework**. Network Working Group, ago. 2003. Disponível em: <https://tools.ietf.org/html/rfc3584>. Acesso em: 11 nov. 2018.

HORST, Adail Spínola; PIRES, Aécio dos Santos; DÉO, André Luis Boni. **De A a Zabbix**. 1. ed. São Paulo: Novatec, 2015.

HOWARD, Jones (2010). **Network Weathermap**. Disponível em: <http://www.network-weathermap.com>. Acesso em: 11 nov. 2018.

KAKANAKOV, Nikolay Rumenov; KOSTADINOVA, Elena Dimitrova; SPASOV, Grisha Valentinov. **Using SNMP for remote measurement and automation**. ELECTRONICS' 2007, 19-21 set., Sozopol, Bulgária, 2007. Disponível em: <https://www.researchgate.net/profile/Nikolay_Kakanakov/publication/228405146_Using_SNMP_for_Remote_Measurement_and_Automation/links/02bfe50d87648e8243000000.pdf>. Acesso em: 27 out. 2018.

KURYLA , Siarhei; SCHÖNWÄLDER, Jürgen. **Evaluation of the resource requirements of SNMP agents on constrained devices**. IFIP International Conference on Autonomous Infrastructure, Management and Security, AIMS 2011: Managing the Dynamics of Networks and Services, 13-17 jun., Nancy, França, 2011, p 100-111. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-21484-4_13>. Acesso em: 11 nov. 2018.

LESSA, Demian. **O Protocolo de gerenciamento RMON**. Rede Nacional de Ensino e Pesquisa (RNP), Boletim bimestral sobre tecnologia de redes, v. 3, n. 1, publicado em 15 jan. 1999. Disponível em: <http://www.rnp.br/newsgen/9901/rmon.html>. Acesso em: 17 out. 2018.

LIMA, Otávio Alcantara de; FRESSE, Virginie; ROUSSEAU, Frédéric. **Evaluation of SNMP-like protocol to manage a NoC emulation platform**. 2014 International Conference on Field-Programmable Technology (FPT), 10-12 dez., Shanghai, China, 2014. Disponível em: <https://ieeexplore.ieee.org/document/7082776/authors#authors>. Acesso em: 11 nov. 2018.

LU, Yung-Feng et al. **A perl-based SNMP agent of networked embedded devices for smart-living applications**. 2015 3rd International Conference on Information and Communication Technology (ICoICT), 27-29 mai. 2015, p. 342–347. Disponível em: <https://ieeexplore.ieee.org/document/7231448>. Acesso em: 21 out. 2018.

MATOS, Leonardo K. **Gerenciamento de equipamentos de rede utilizando o software CACTI**. Trabalho de Conclusão de Curso Especialização em Redes e Segurança de Sistemas. Pontifícia Universidade Católica do Paraná, 2009.

MAURO, Douglas R.; SCHMIDT, Kevin J. **Essential SNMP**. 2. ed. Sebastopol: O' Reilly, 2001.

NETACAD. **Cisco Networking Academy**. Disponível em: <https://www.netacad.com/pt-br>. Acesso em: 04 nov. 2018.

NIC-BR (2018). **Núcleo de Informação e coordenação do Ponto BR (NIC-BR)**. Disponível em: <https://www.nic.br/>. Acesso em: 11 nov. 2018.

PAVENTHAN, A. et al. **WSN monitoring for agriculture: comparing SNMP and emerging CoAP approaches**. 2013 Texas Instruments India Educators' Conference, 4-6 abr. 2013. p. 353-358. Disponível em: <https://ieeexplore.ieee.org/document/6757167?arnumber=6757167&tag=1>. Acesso em: 20 out. 2018.

STALLINGS, William. **Redes e sistemas de comunicação de dados: teoria e aplicações corporativas**. 5. ed. Rio de Janeiro: Campus-Elsevier, 2005.

TEIXEIRA JÚNIOR, José Helvécio et al. **Redes de computadores - serviços, administração e segurança**. São Paulo: Makron Books, 1999.

VLADISHEV, Alexei. **Zabbix documentation 3.0**. Copyright© 2001-2018, Zabbix SIA, 2018. Disponível em: <https://www.zabbix.com/documentation/3.0/pt/manual>. Acesso em: 12 nov. 2018.

ZABBIX. **Zabbix documentation 3.4**. Copyright© 2001-2018, Zabbix SIA, 2018. Disponível em: <https://www.zabbix.com/documentation/3.4/manual/installation/install_from_packages/debian_ubuntu>. Acesso em: 12 nov. 2018.