

# Otimização da Ferramenta de Monitoramento Zabbix no Hospital Universitário de Santa Maria

Vinícius Röpke<sup>1,2</sup>, Matheus Guarienti<sup>1,2</sup>, Tiago Antonio Rizzetti<sup>1</sup>

<sup>1</sup>Redes de Computadores – Universidade Federal de Santa Maria (UFSM)  
Caixa Postal 8052 – 97.105-900 – Santa Maria – RS – Brazil

<sup>2</sup>Hospital Universitário de Santa Maria (HUSM)  
Universidade Federal de Santa Maria (UFSM) – Santa Maria, RS – Brazil

{viniropke, matheus.guarienti, rizzetti}@redes.ufsm.br

**Abstract.** *This article describes the optimization of the Zabbix network monitoring tool at the Hospital Universitário de Santa Maria, in which the existing servers had problems caused by poor configuration. The high amount of network assets monitored by the tool caused the machine to slow down. New assets have been added to ensure item history. This makes it possible to generate statistical reports, and real-time reporting of network problems. To solve the problems, we aimed to optimize the tool, reviewing items from each host, linking templates, adapting the use of housekeeping (database records removal) function, hardware upgrade, among other processes improving server performance.*

**Resumo.** *Este artigo descreve a otimização da ferramenta de monitoramento de rede Zabbix no Hospital Universitário de Santa Maria, no qual os servidores já existentes apresentavam problemas gerados pela má configuração. A alta quantidade de ativos de rede monitorados pela ferramenta, causou a lentidão da máquina. Foram adicionados novos ativos, visando garantir o histórico dos itens. Assim é possível gerar relatórios estatísticos, e a informação em tempo real de problemas na rede. Para solução dos problemas, visou-se a otimização da ferramenta, revisando itens de cada host, vinculando templates, adequando o uso da função housekeeping (remoção de registros no banco de dados), upgrade de hardware, entre outros processos melhorando o desempenho do servidor.*

## 1. Introdução

O bom funcionamento da rede é algo muito importante para uma instituição que dela depende. Com a alta competitividade e visando maior sobrevivência no mercado, empresas com altos volumes de ativos de rede precisam ser afetadas o menos possível pela indisponibilidade de seus sistemas. Esses e outros fatores tornaram a estruturação e a implementação de redes resilientes uma tarefa desafiadora para qualquer equipe de TI. O pilar da necessidade de monitoramento se sustenta na premissa da maximização da eficiência e da produtividade da rede, através de mecanismos e ferramentas de análise de recursos.

Para implementar a gerência e o monitoramento eficiente de redes, são necessárias ferramentas específicas, como por exemplo, Zabbix, Cacti, Nagios, entre outras. Através

do gerenciamento é possível ter controle total da rede, em que seja possível evitar problemas futuros desta, além de ter serviços melhorados, então, visando obter um desempenho melhor da rede [Black 2008]. O monitoramento destes dispositivos na rede é feito através de serviços locais dessas ferramentas, normalmente utilizando um dos seguintes métodos: (a) monitoramento simples, através de agentes/gerentes ou scripts; (b) com protocolos destinados ao gerenciamento de rede, como o *Simple Network Management Protocol* (SNMP); (c) ou ainda com protocolos para acesso remoto, empregados com a finalidade de gerenciamento, como *Secure Shell* (SSH) e *Telnet*. Através da definição do tipo de monitoramento, são retornados valores que as ferramentas realizarão o tratamento e a partir deles é gerado alertas, gráficos, históricos de eventos [Horst et al. 2015].

No presente artigo é abordada a otimização (melhoramento da implementação) da ferramenta Zabbix no Hospital Universitário de Santa Maria (HUSM) pois a mesma apresentava um desempenho inadequado, ocasionando paradas de serviços, e alarmes falsos devido a quantidade de itens sendo monitorados desnecessariamente. A primeira versão instalada foi a Zabbix 2.8. Após, foram realizados upgrades para versões mais recentes. Atualmente, a versão instalada é a 4.2. Além disso, novos *hosts* e equipamentos foram adicionados visando manter histórico de itens, gerar relatórios para estatísticas e auxiliar na identificação e resolução de eventos anormais, já que a ferramenta faz o monitoramento dos equipamentos em tempo real. O presente artigo desenvolvido pelos autores, refere a maneira que a ferramenta de monitoramento Zabbix foi otimizada para um melhor desempenho no Hospital Universitário de Santa Maria, devido aos problemas de desempenho que vinha apresentando.

O Zabbix, é uma aplicação que traz diversos pontos fortes. Algumas qualidades são: *software* de rápida resposta de monitoramento, interface amigável que procura sempre facilitar seu uso, alta qualidade comercial, controle de usuários, *Open-Source* (totalmente gratuito), entre outros. Visando o gerenciamento dos nodos monitorados, foi utilizada estratégia complexa, onde cada item criado no *host* ou no próprio *template*, teve que ser avaliado para saber seu nível de importância. Toda estratégia de gerenciamento utilizada e a técnica de remoção de registros inutilizáveis (*housekeeping*), resulta em uma máquina otimizada [Horst et al. 2015] [ZABBIX SIA 2019]. Recentemente, foi realizado um *upgrade* de hardware na máquina rodando o sistema Zabbix. Nela, foi substituído o HD por um SSD e adicionados 4GB de memória RAM, passando para 16GB. Essa mudança objetiva um maior ganho de desempenho do sistema.

## 2. Zabbix

O Zabbix é uma moderna e completa ferramenta multiplataforma sob licença GPLv2 para monitoramento de redes. A ferramenta surgiu da necessidade que o administrador de sistemas, Alexei Vladishv, obteve quando precisou de uma ferramenta de monitoramento comercial financeiramente barata e que não precisasse de conhecimentos avançados para ser dominada. Seu objetivo era monitorar os servidores do Deutsch-Lettische Bank, instituição financeira onde trabalhou entre 1995 e 1996. Formado pela Latvijas Universitate, na Letônia, Alexei desenvolveu um sistema próprio de monitoramento que atendesse suas necessidades, dando origem a primeira versão (Alpha) do popular Zabbix, lançada em 2001. Com a inerente adesão, foi lançada, em 2004, a versão estável da ferramenta. No ano seguinte, foi fundada a Zabbix SIA *Company*, grupo que até hoje é responsável pelos suportes técnico e comercial, além do desenvolvimento e mantimento do software

[Prete], Jorge 2014]. O Zabbix possibilita a configuração de notificações e alertas por e-mail, Jabber e SMS para uma infinidade de acontecimentos. O sistema utiliza um Sistema de Gerenciamento de Banco de Dados (SGBD) para mantimento das informações, que são MySQL/Mariadb, PostgreSQL, SQLite, Oracle ou IBM DB2. Dessa forma, o Zabbix em si pode ser compreendido em três grandes partes: Server, Agent e Proxy [Horst et al. 2015].

## **2.1. Zabbix Server**

No Zabbix Server, o mesmo atua centralizando todas as informações coletadas através do Zabbix Agent, Zabbix Trapper (dados que são enviados ao invés de terem sido buscados), protocolo SNMP ou checagem normal. As informações geradas e coletadas são mantidas em um banco de dados que se relaciona diretamente com uma interface WEB (Dashboard), utilizada pelo administrador da rede.

## **2.2. Zabbix Agent**

O Agente do Zabbix atua, basicamente como um cliente do Zabbix Server, tendo em vista a arquitetura Cliente/Servidor básica. Existem várias soluções para diversos ambientes de instalação, como Linux, Solaris, Windows, entre outros [Ramos, Bezaliel 2016]. Além da possibilidade de instalação, o agente pode atuar passivamente ou ativamente, dependendo da necessidade. Na forma ativa, o agente estabelece uma conexão com o Zabbix Server e solicita uma lista dos itens que realizará o monitoramento do nodo ativo. Na forma passiva, o Zabbix Server estabelece uma conexão com o agente e solicita os resultados das coletas.

## **2.3. Zabbix Proxy**

O Zabbix Proxy atua de forma muito semelhante a um Zabbix Server. Entretanto, de forma subordinada ao último. A utilização é opcional, porém possui a vantagem de distribuir a carga de análise e processamento das informações do Zabbix Server, uma vez que armazena em *buffer* as informações transferidas a ele. É uma solução vantajosa para o monitoramento centralizado de localidades geograficamente separadas ou gerenciadas remotamente [ZABBIX SIA 2019].

# **3. Otimização da Ferramenta**

Com a necessidade de diminuir carga de coleta de dados de nodos ativos presentes no hospital, usou-se métricas de filtragem, temporalidade e aumento dos recursos de *hardware*. Dentre essas categorias, utilizou-se templates, peças de *hardware*, ferramenta de remoção de registro inutilizáveis do banco de dados (housekeeping) e definindo gravidade de itens e dispositivos (os mais importantes).

## **3.1. Filtragem**

A utilização da filtragem com o uso de templates foi a principal técnica para resolução dos problemas de baixo desempenho, alarmes falsos, entre outros problemas que vinham se apresentando. Templates é definido por “Ambiente estabelecido como modelo, permitindo criar conteúdo de uma forma rápida”, por exemplo, protótipos de regras de coleta, descoberta de *hosts* e itens, alertas e gráficos que são gerados conforme os dados. A

técnica foi a principal, porque em vez de configurar várias vezes o item para diversos servidores, cria-se um e nele vincula-se os itens, facilitando o gerenciamento de itens criados nele. Os templates não são apenas criados, podem ser importados de fabricantes conforme modelo de dispositivo e da *Management Information Base (MIB)* [Horst et al. 2015].

A figura 1 mostra os templates criados/importados que são utilizados pela ferramenta para realizar as coletas de determinados grupos, por exemplo, o template “Monitoramento NoBreak - RACKs - HUSM” faz a gerência dos *nobreaks* dos armários de telecomunicações presentes no hospital. Nessa filtragem, foi analisado e editado com cautela, na intenção de adicionar, editar e remover itens de coletas desnecessário para melhorar o desempenho da máquina e deixar somente informações necessárias.

Nome ▲	Aplicações	Itens	Triggers	Gráficos	Telas	Descoberta	Web
Monitoramento Impressoras MP-501 - SP-4510SF - CPC - HUSM	Aplicações 1	Itens 4	Triggers	Gráficos 3	Telas	Descoberta	Web
Monitoramento Impressoras MP-C407 - MP-C307 - CPC - HUSM	Aplicações 1	Itens 9	Triggers	Gráficos 4	Telas	Descoberta	Web
Template Monitoramento Access Point - CISCO	Aplicações	Itens 1	Triggers 1	Gráficos	Telas	Descoberta	Web
Template Monitoramento Extreme - CPD	Aplicações 7	Itens 4	Triggers 2	Gráficos 1	Telas	Descoberta 4	Web
Template Monitoramento Impressoras CPC - HUSM	Aplicações	Itens 6	Triggers 3	Gráficos 4	Telas	Descoberta	Web
Template Monitoramento NoBreak - RACKs - HUSM	Aplicações	Itens 9	Triggers 4	Gráficos 4	Telas	Descoberta	Web
Template Monitoramento Relógio Ponto - EBSEH - HUSM	Aplicações 1	Itens 1	Triggers 1	Gráficos	Telas	Descoberta	Web
Template Monitoramento SNMP PDU - HUSM	Aplicações 4	Itens 11	Triggers 7	Gráficos 3	Telas	Descoberta	Web
Template Monitoramento Switchs - HUSM - HP	Aplicações 4	Itens 11	Triggers 5	Gráficos 1	Telas	Descoberta 1	Web
Template Monitoramento Switchs - HUSM - Huawei	Aplicações 3	Itens 7	Triggers 1	Gráficos 1	Telas	Descoberta 1	Web

**Figura 1. Templates Criados/Importados no HUSM.**

A tabela 1 ilustra a comparação do antes e depois do template “Monitoramento NoBreak RACKs - HUSM”.

**Tabela 1. Itens do Template Utilizado para Monitoramento de *NoBreaks*.**

Antes			Depois		
Nome do Item	Intervalo	Estatística	Nome do Item	Intervalo	Estatística
Load %	30 segs.	365 dias	Capacid. de Saída	2 minutos	30 dias
Frequência	30 segs.	365 dias	Frequência	10 minutos	30 dias
Uptime	1 minuto	365 dias	Incoming eth0	10 minutos	7 dias
ICMP Ping	1 minuto	365 dias	Outgoing eth0	10 minutos	7 dias
Input Voltage	30 segs.	365 dias	Nível da Bateria	5 minutos	90 dias
Oulput Voltage	30 segs.	365 dias	ICMP Ping	1 minuto	30 dias
Temperatura (C°)	30 segs.	365 dias	Temperatura (C°)	10 minutos	30 dias
Capacidade	30 segs.	365 dias	Tensão de Entrada	2 minutos	90 dias
Protótipo de Interfaces	1 hora	365 dias	Tensão de Saída	5 minutos	30 dias
Description	1 minuto	TEXTO	-	-	-
Location	1 minuto	TEXTO	-	-	-
Name	1 minuto	TEXTO	-	-	-
Contact	1 minuto	TEXTO	-	-	-
ICMP Response	1 minuto	365 dias	-	-	-
ICMP Loss	1 minuto	365 dias	-	-	-

Pode-se observar a otimização realizada no mesmo template, além de reduzir a quantidade de itens, foi realizado o ajuste dos tempos de intervalos (tempo de busca) e

históricos de estatísticas (gráficos) mais regulares de acordo com o grupo de *nobreaks* a ser monitorado.

É importante relatar sobre um item em específico sobre a tabela abaixo. O “Protótipo de Interfaces” é um item referente a descoberta de interface, entrando em execução a cada 1 hora, descobrindo todas as interfaces presentes no *host*, então, adicionando às interfaces conforme os itens do protótipo. Por exemplo, o citado possui descoberta para interfaces de Administração, Tráfego, Erros, Descrição e Operação.

### 3.2. Temporalidade

Outra técnica de otimização usada, mas não menos importante é o uso do *housekeeping* (serviço de limpeza). Na ferramenta esse serviço atua de forma a remover os registros que ultrapassam o período de “validade” de armazenamento em banco de dados. Todos os dados coletados pelos Zabbix Server dos *hosts*, possuem ou não um tempo de retenção mínima, por exemplo, pode-se armazenar dados durante um dia ou um ano.

Para configuração do serviço de limpeza, existem setores dentro da página *web* do Zabbix que é possível alterá-la a qualquer momento. A imagens abaixo ilustra as configurações realizadas sobre a retenção dos dados no Hospital Universitário, com valores mínimos com o intuito de ter históricos suficientes para soluções de problemas que necessitam de dados passados e valores máximos para não deixar a máquina sobrecarregada com quantidade de dados desnecessários.

A figura 2 mostra o tempo configurado para de terminadas ações internas da máquina, com um período global de 60 dias para retenção de dados de serviços, eventos, alertas, auditoria, usuários.

The image shows a screenshot of the Zabbix configuration interface, divided into several sections with checkboxes and input fields for retention periods:

- Eventos e alertas:**
  - Ativar a limpeza de dados interna:
  - \* Período de retenção dados de triggers: 60d
  - \* Período de retenção interna de dados: 60d
  - \* Período inválido para armazenamento dos dados de descoberta de rede: 60d
  - \* Período de armazenamento dos dados de autorregistro: 60d
- Sessões de usuários:**
  - Ativar a limpeza de dados interna:
  - \* Período de retenção de dados: 60d
- Histórico:**
  - Ativar a limpeza de dados interna:
  - Substituir o período de histórico do item:
  - \* Período de retenção de dados: 90d
- Serviços:**
  - Ativar a limpeza de dados interna:
  - \* Período de retenção de dados: 60d
- Auditoria:**
  - Ativar a limpeza de dados interna:
  - \* Período de retenção de dados: 60d
- Estatísticas:**
  - Ativar a limpeza de dados interna:
  - Substituir o período de retenção das estatísticas (médias):
  - \* Período de retenção de dados: 365d

**Figura 2. Configurações de tempos de remoção de registros globalmente.**

Históricos e Estatísticas podem ser configurados de duas maneiras, item por item em um template ou *host* ou globalmente atribuindo uma valor padrão para todos, como apresentado na figura 2. A configuração usada na ferramenta para remoção de registros foi configurada em itens, por possuir diversos tipos de nodos ativos (impressoras, *nobreaks*, servidores) com critérios críticos superiores a outros.

Na figura 3, é retratado a configuração de itens de um template de *nobreaks* utilizado no HUSM, monitorando temperatura, entrada e saída de tensão e dados, entre outros. Nestes itens, cada um internamente pode ser configurado com tempo de retenção dos dados conforme analisados pela equipe de TI do hospital, por exemplo o item “Temperatura (C°)”, tem um histórico (dados) de 15 dias e estatísticas (gráficos) de 30 dias.

Nome	Triggers	Chave	Intervalo	Histórico	Estatísticas
Temperatura (C°)	Triggers	degrees	10m	15d	30d
Frequência		hertz	10m	1w	30d
PING - ICMP	Triggers	icmpping[...]	1m	30d	30d
Incoming traffic on interface eth0		ifInOctets["eth0"]	10m	1w	1w
Outgoing traffic on interface eth0		ifOutOctets["eth0"]	10m	1w	1w
Nível da Bateria	Triggers	percent	5m	30d	90d
Capacidade de Saída		percent1	2m	2w	30d
Tensão de Entrada	Triggers	volts	2m	2w	90d
Tensão de Saída		Volts1	5m	2w	30d

Figura 3. Configurações de tempos de remoção de registros por item.

Com os períodos de tempos definidos, houve necessidade de configurar o processo de *housekeeping* presente em linhas do arquivo “zabbix\_server.conf”. Neste arquivo aborda-se todas as configurações da ferramenta, como nome e senha de banco de dados, limites mínimos e máximos da ferramenta, entre outros.

No problema inicial, a ferramenta estava com configurações padronizadas de instalação, onde o valor de “MaxHousekeeperDelete” está atribuído a 5000 valores, então, a máquina iria deletar somente 5000 valores da imensa quantidade de dados que a máquina coletava, com um tempo de intervalo (“HousekeepingFrequency”) de uma hora. Para aprimorar essa limpeza a figura 4 exibe as configurações que foram realizadas no arquivo com o objetivo de agrupar está técnica com as apresentadas anteriormente, atribuindo 50000 valores para realizar a remoção dos dados “vencidos”, em um mesmo período de frequência.

```
### Option: MaxHousekeeperDelete
# Mandatory: no
# Range: 0-1000000
# Default:
MaxHousekeeperDelete=50000

### Option: HousekeepingFrequency
# Mandatory: no
# Range: 0-24
# Default:
HousekeepingFrequency=1
```

Figura 4. Configurações do processo de *housekeeping*.

### 3.3. Upgrade de Hardware

Recentemente implementado, o *upgrade de hardware* foi a metodologia mais poderosa para solução dos problemas. O avanço para o aumento do processamento da máquina foi feito de 2 formas, com um aumento de memória *Random Access Memory* (RAM) e

migrando o *Hard Disk Drive* (HD) para *Solid State Drive* (SSD), aprimorando o sistema de leitura, gravação e execução no sistema de armazenamento.

De início, o servidor Zabbix atuava em cima de 12 *Gigabytes*(GB) de memória randômica. Após estudos, encontrou-se uma maneira de reservar memória somente para atuação do banco de dados, simplificando o armazenamento e busca de valores em nível de *data base*. Logo, teve a inserção de mais 4 GB de memória, totalizando 16 GB.

A segunda progressão de *hardware* recentemente implementada, foi a troca do HD por um SSD. Com essa atitude, o desempenho da máquina mudou-se completamente, em relação a uso de CPU, Memória, e Processos de *housekeeping*, cache, entre outros. Nas figuras 5 e 6 é exibido dados respectivos das técnicas de upgrade em relação a quantidade de recurso usada pela aplicação do Zabbix com média avaliada em um período de 30 dias.

		último	min	méd	max
Zabbix busy timer processes, in %	[méd]	0.028 %	0 %	0.0514 %	100 %
Zabbix busy escalator processes, in %	[méd]	13.9192 %	1.5082 %	15.9725 %	100 %
Zabbix busy housekeeper processes, in %	[méd]	100 %	0 %	99.3283 %	100 %
Zabbix busy alerter processes, in %	[méd]	0.0438 %	0 %	1.0194 %	100 %
Zabbix busy configuration syncer processes, in %	[méd]	0.6183 %	0 %	0.7026 %	69.9712 %
Zabbix busy db watchdog processes, in %	[não há dados]				
Zabbix busy history syncer processes, in %	[méd]	23.74 %	0.6625 %	25.1152 %	100 %
Zabbix busy self-monitoring processes, in %	[méd]	0.0023 %	0 %	0.001968 %	0.0339 %

Figura 5. Porcentagem Zabbix *Internal Process Busy* antes das Otimizações.

		último	min	méd	max
Zabbix busy timer processes, in %	[méd]	0.0014 %	0 %	0.13 %	100 %
Zabbix busy escalator processes, in %	[méd]	0.21 %	0 %	0.28 %	100 %
Zabbix busy housekeeper processes, in %	[méd]	1.85 %	0 %	1.65 %	100 %
Zabbix busy alerter processes, in %	[méd]	0.1 %	0 %	0.48 %	66.6 %
Zabbix busy configuration syncer processes, in %	[méd]	1.83 %	0 %	1.85 %	18.3 %
Zabbix busy db watchdog processes, in %	[não há dados]				
Zabbix busy history syncer processes, in %	[méd]	0.24 %	0 %	0.41 %	100 %
Zabbix busy self-monitoring processes, in %	[méd]	0.004 %	0 %	0.004555 %	25.08 %

Figura 6. Porcentagem Zabbix *Internal Process Busy* depois das Otimizações.

Com os métodos de otimizações das seções 3.1, 3.2 e 3.3 implementados, abaixo é descrito o antes e depois da atuação da ferramenta, quantidade de valores obtidos por segundo em determinado período de otimização.

Antes de toda configuração da ferramenta, eram obtidos em torno 250 valores por segundo. Com essa alta quantidade de valores, afetou a técnica de *housekeeping*, ficando sempre em execução, consumindo todo recurso de *hardware* da máquina, para remoção de registros impossibilitando o uso para os demais recursos da máquina, conforme mostrado na figura 5.

Ao se estudar os mecanismos para otimização e implementá-los, houve uma queda destes valores. Dos templates refeitos e avaliados, esses dados coletados passaram de 250 para aproximadamente 150 valores por segundo, conforme as figuras 7 e 8. O *housekeeping* foi configurado conforme a figura 4, para atuar a cada 1 hora, apagando no máximo 50000 mil valores em seu tempo de atividade. Diante disso, a ferramenta obteve estabilidade, corrigindo problema de baixo desempenho.

	méd
Values processed by Zabbix server per second	244.88
Zabbix queue	20.06

Figura 7. Antes da otimização - Valores coletados por segundo.

■ Values processed by Zabbix server per second	méd 127.3
■ Zabbix queue	4.76

Figura 8. Depois da otimização - Valores coletados por segundo.

#### 4. Monitoramento de Itens e *Hosts*

Nesta sessão, é abordado como é realizado a inserção na ferramenta de novos *hosts* e itens para melhoria do serviço oferecido pelo hospital. Para um monitoramento mais amplo e com maior desempenho, foram analisados em parte grupos de *hosts* que possuíam um grande volume de dados providos de itens desnecessários. Para realização deste processo, foi empregada uma técnica manual de filtragem. Esse método analisa itens que cada *host* e template abrangiam. Após essa análise dos itens, usa-se a técnica abordada no tópico 3.1, para organização e agrupamentos de itens para determinado grupo, evitando que itens fiquem espalhados e distintos, facilitando o gerenciamento futuramente.

Com problemas solucionados, a equipe de TI do Hospital Universitário decidiu abranger uma maior área monitorada, adicionados dispositivos possíveis de monitoramento, para monitoramento de sala de *Data Center*, *Access Points*, novos servidores, entre outros. A finalidade desse monitoramento é em vista para automatizar o processo de solução de problemas por técnicos da TI. Todo o novo *host* acionado na ferramenta de monitoramento é enquadrado em determinado grupo de dispositivos, com template adequado.

#### 5. Considerações Finais

O problema de abundância de dados coletados devido grande quantidade de *hosts* foi sanado através de um estudo sobre a ferramenta Zabbix, implementando técnicas para organização e remoção de registros no banco de dados. Além disso, trouxe outros ganhos para a ferramenta suportar um monitoramento mais abrangente na rede do Hospital Universitário de Santa Maria.

Hoje se estuda novas técnicas de melhoria de desempenho presentes na nova versão 4.2 do Zabbix, com finalidade de deixar a ferramenta mais otimizada possível, possibilitando uma implementação adequada e segura. Outro projeto é a integração da ferramenta Zabbix com sistema de gráficos e *Help Desk*, como Grafana (visualizar métricas por meios de gráficos) e OTRS (sistema de gestão de serviços).

#### Referências

- Black, T. L. (2008). Comparação de ferramentas de gerenciamento de redes. *Universidade Federal do Rio Grande do Sul - Instituto de Informática*.
- Horst, A. S., dos Santos Pires, A., and Déo, A. L. B. (2015). *De A a Zabbix: Aprenda a monitorar e gerenciar aplicações e equipamentos de redes com o Zabbix*. Novatec Editora.
- Pretel, Jorge (2014). Conheça o zabbix, história e arquitetura. Acesso em 19 jul. 2019.
- Ramos, Bezaluel (2016). Monitoração de ativos de rede com o zabbix. Acesso em 19 jul. 2019.
- ZABBIX SIA (2019). Zabbix. Acesso em 15 jul. 2019.