PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

GERENCIAMENTO E MONITORAÇÃO DE REDES DE COMPUTADORES COM ÊNFASE EM DISPONIBILIDADE DE SERVIDOR *WEB* COM FERRAMENTA ZABBIX

JOÃO XAVIER DA SILVA NETO

GOIÂNIA
2021/1

JOÃO XAVIER DA SILVA NETO

GERENCIAMENTO E MONITORAÇÃO DE REDES DE COMPUTADORES COM ÊNFASE EM DISPONIBILIDADE DE SERVIDOR *WEB* COM FERRAMENTA ZABBIX

Trabalho de Conclusão de Curso apresentado à Escola de Ciências Extas e da Computação, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Orientadora: Prof.ª Ma. Angélica da Silva Nunes

GOIÂNIA
2021/1

JOÃO XAVIER DA SILVA NETO

GERENCIAMENTO E MONITORAÇÃO DE REDES DE COMPUTADORES COM ÊNFASE EM DISPONIBILIDADE DE SERVIDOR *WEB* COM FERRAMENTA ZABBIX

Este Trabalho de Conclusão de Curso julgado adequado para obtenção do título de Bacharel em Engenharia de Computação, e aprovado em sua forma final pela Escola de Ciências Exatase da Computação, da Pontifícia Universidade Católica de Goiás, em_____/_____/_____.

_____
Prof.ª Ma. Ludmilla Reis Pinheiro dos Santos
Coordenadora de Trabalho de Conclusão de Curso

Banca Examinadora:

_____
Orientadora: Prof.ª Ma. Angélica da Silva Nunes

_____
Prof. Dra. Solange da Silva

_____
Prof. Me. Wilmar Oliveira de Queiroz

GOIÂNIA
2021/1

DEDICATÓRIA

Primeiramente a Deus que me fortaleceu nas horas mais necessárias. Aos meus pais por sempre me apoiarem no meu sonho de ter uma formação acadêmica, pelo carinho e por ter me dado essa oportunidade de esta aqui.

AGRADECIMENTOS

A professora Ma. Angélica da Silva Nunes, orientadora acadêmica, a qual agradeço pelo apoio, paciência e empenho no apoio deste trabalho académico.

Aos meus colegas que me incentivaram a não desistir e me auxiliaram em diversos momentos durante esses 6 anos.

“O segredo da criatividade é: Saber como esconder suas fontes”

Albert Einstein

RESUMO

Este trabalho tem como objetivo mostrar como o monitoramento de aplicações pode ser realizado pelo Zabbix, com ênfase em sua disponibilidade de serviços e recursos. O objetivo é demonstrar como os alertas de incidentes de uma aplicação de alta disponibilidade podem auxiliar no *troubleshooting* mais eficiente para equipes responsáveis pelas aplicações ou recursos utilizados. Neste trabalho foi criada a estrutura de rede a qual será monitorada, composta por: um servidor de gerenciamento, no qual o servidor Zabbix está instalado e configurado, um servidor *Web* Apache para testar a disponibilidade dos serviços e recursos constituintes e computador que servira como estação de monitoramento. Como resultado, foi possível verificar como a ferramenta Zabbix pode lidar com a identificação, distribuição de alertas e da execução de pré-comandos remotos para restabelecimento de serviços. O uso do Zabbix pode auxiliar o administrador ou sua equipe a identificar, criar métricas e expressões para o ambiente monitorado o que auxilia na prevenção e correção destes problemas.

Palavras-Chave: Aplicações, Disponibilidade, Monitoramento de Redes, Zabbix.

ABSTRACT

This work aims to show how application monitoring can be performed by Zabbix, with emphasis on its availability of services and resources. The objective is to demonstrate how incident alerts from a high availability application can help in more efficient troubleshooting for teams responsible for the applications or resources used. In this work, the network structure that will be monitored was created, consisting of: a management server, in which the Zabbix server is installed and configured, an Apache Web server to test the availability of services and constituent resources, and a computer that will serve as a station of monitoring. As a result, it was possible to see how the Zabbix tool can handle the identification, distribution of alerts and the execution of remote pre-commands to restore services. The use of Zabbix can help the administrator or his team to identify, create metrics and expressions for the monitored environment, which helps to prevent and correct these problems.

Keywords: Applications, Availability, Network Monitoring, Zabbix.

LISTA DE FIGURAS

Figura 1 - Principais Componentes de uma arquitetura de rede 15
Figura 2 - Árvore de identificadores de objetos ASN.1 17
Figura 3 - Objetos gerenciados no grupo system da MIB-2 18
Figura 4 - Tipos de dados Básicos da SMI 19
Figura 5 - Modo de funcionamento do GetRequest 20
Figura 6 - Modo de funcionamento do SetRequest 21
Figura 7 - Modo de funcionamento do GetNextRequest 22
Figura 8 - Modo de funcionamento do GetBulkRequest 22
Figura 9 - Modo de funcionamento do Trap 23
Figura 10 - Arquitetura de NMS Centralizada 24
Figura 11 - Arquitetura de NMS Distribuída 25
Figura 12 - Arquitetura de NMS Híbrida 26
Figura 13 - Topologia de rede 32
Figura 14 - Itens Template_Teste_Disp_Porta 33
Figura 15 - Triggers de conexão negada do servidor Web 34
Figura 16 - Criação das Triggers de conexão negada do servidor Web 35
Figura 17 - Inclusão de domínio para Apache 36
Figura 18 -Criação do certificado 37
Figura 19 - Script de verificação de certificado 38
Figura 20 - Criação item responsável pela execução do script 38
Figura 21 - Dados do certificado exibido pelo Zabbix 39
Figura 22 - Cenário monitoração web 40
Figura 23 - Notificações enviadas ao Discord 41
Figura 24 -Notificações enviadas ao Telegram 42
Figura 25 - Configuração específica de envio 44
Figura 26 - Notificação enviada ao Gmail 43

LISTA DE TABELAS

Tabela 1 - Tabela de pré-requisitos para instalação do Zabbix 28
Tabela 2 - Plataformas suportadas 28
Tabela 3 - Bancos de dados suportados 29
Tabela 4 - Requisitos de *software* 29

LISTA DE SIGLAS

API	Interface de Programação de Aplicação - *Application Programming Interface*
ASN.1	Sintaxe Abstrata Notação Um- *Syntax Notation One*
CPU	Unidade Central de Processamento - *Central Process Unit*
DNS	Sistema de Nomes de Domínios - *Domain Name System*
FTP	Protocolo de Transferência de Arquivos - *File Transfer Protocol*
GPL	Licença Pública Geral - *General Public License*
HTTP	Protocolo de Transferência de Hipertexto - *HyperText Transfer Protocol*
HTTPS	Protocolo de Transferência de Hipertexto Seguro - *Hyper Text Transfer Protocol Secure*
IMAP	Protocolo de acesso a mensagem da *internet* - *Internet Message Access Protocol*
IP	Protocolo de *Internet* - *Internet Protocol*
ISO	Organização Internacional para Padronização - *International Organization for Standardization*
JSON	Notação de Objetos JavaScript - *JavaScript Object Notation*
LDAP	Protocolo de Acesso a Diretório Leve - Lightweight Directory Access Protocol
MI	*Megabyte*
MIB	Base de Informações de Gerenciamento - *Management Information Base*
NNTP	Protocolo de Transferência de Notícias da Rede - *Network News Transfer Protocol*
NMS	Estação de Gerenciamento - *Network Management Station*
NOC	Centro De Operações De Rede - *Network Operations Center*
OID	Identificador Único - *Object Identifier*

OTRS Sistema de Solicitação de Tíquetes de Código Aberto - *Open-source Ticket Request System*

P2P Ponto a Ponto – *Point to Point*

POP Procedimento Operacional Padrão - *Standard Operational Procedure*

RAM Memória de Acesso Aleatório - *Random Access Memory*

RFC Pedido de Comentários - *Request for Comments*

RPC Chamada Remota de Procedimento - *Remote Procedure Call*

SGMP Protocolo de Monitoramento do Gateway Simples - *Simple Gateway Monitoring Protocol*

SMI Estrutura de Informações de Gerenciamento - *Structure of Management Information*

SMS Serviço de Mensagens Curtas - *Short Message Service*

SNMP Protocolo Simples de Gerenciamento de Redes - *Simple Network Managent Protocol*

SSH Protocolo de Rede Criptográfico - *Secure Socket Shell*

SSL Camada de Soquete Seguro - *Secure Socket Layer*

TCP Protocolo de Controle de Transmissão - *Transmission Control Protocol*

TCP/IP Protocolo de Controle de Transmissão/Protocolo de Internet - *Transmission Control Protocol/Internet Protocol*

TELNET Protocolo de Terminal Virtual - *Virtual Terminal Protocol*

TI Tecnologia da Informação

TLS Segurança de Camada de Transporte - *Transport Layer Security*

URL Localizador Padrão de Recursos - *Uniform Resource Locator*

VM Máquina Virtual - *Virtual Machines*

Sumário

1 INTRODUÇÃO 11
1.1 Objetivo Geral 12
1.2 Objetivo Específicos 12
1.3 Procedimentos Metodológicos 13
1.4 Estrutura da Monografia 13
2 GERENCIAMENTO DE REDES 14
2.1 O que é gerenciamento de rede? 14
2.2 Modelo ISO de gerenciamento de redes 14
2.3 Arquitetura do Gerenciamento 15
2.4 ESTRUTURA DE GERENCIAMENTO PADRÃO *INTERNET* 16
2.4.1 Base de Informações de Gerenciamento – MIB 16
2.4.2 Estrutura de Informações de Gerenciamento – SMI 18
2.4.3 Protocolo SNMP 19
2.5 Modos Operacionais SNMP 20
2.5.1 *GetRequest* 20
2.5.2 *SetRequest* 20
2.5.3 *GetNextResponse* 21
2.5.4 *GetBulk* 22
2.5.5 *Trap* 23
2.6 ARQUITETURAS DE NMS 23
2.6.1 Arquitetura Centralizada 23
2.6.2 Arquitetura Distribuída 25
2.6.3 Arquitetura Híbrida 26
3 FERRAMENTA DE GERENCIAMENTO 27
3.1 Pré-requisitos de sistema 27
3.1.1 Requisitos de *hardware* 27
3.1.2 Requisitos de *software* 28
3.2 Conceitos Zabbix 29
3.2.1 Definições zabbix 29
4 USO DA GERÊNCIA DE REDE NA IDENTIFICAÇÃO DE INCIDENTES EM UM SERVIDOR *WEB* 31
4.1 Apresentação do Ambiente 31
4.2 Elaboração do monitoramento 32
4.3 Disponibilidade de conexão 32
4.4 Certificado SSL 36

4.5 Monitoração *Web* 39
4.6 Integração com mídias 40
5 CONCLUSÃO 45
5.1 Sugestões de Trabalhos futuros 46
ANEXO A – INSTALAÇÃO DO ZABBIX 5.0.9 51
ANEXO B – CONFIGURAÇÃO MÍDIA DO TELEGRAM 59
ANEXO C – CONFIGURAÇÃO MÍDIA DO DISCORD 62
ANEXO D - CONFIGURAÇÃO MÍDIA DO GMAIL 65
ANEXO E – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA 67

1 INTRODUÇÃO

Com o crescimento da exigência da *Internet*, o gerenciamento de redes ficou indispensável para o dia a dia do mercado de Tecnologia da Informação (TI). Para que se tenha um bom gerenciamento é preciso alcançar um bom fluxo no tráfego das informações, que os recursos sejam corretamente utilizados e não sobrecarregados e que os dados sejam transportados com confiabilidade e segurança.

O gerenciamento de redes habitualmente é feito por uma equipe que inclui, no mínimo, um administrador de redes e uma equipe de TI. O administrador e sua equipe são os responsáveis:

- Pela qualidade dos ativos e serviço de infraestrutura;
- Avaliar e projetar a implementação de *upgrades* de ativos, estruturas e *links* de dados de acordo com a necessidade da empresa;
- Ter o conhecimento da estrutura física para ingressar os equipamentos conectados na rede.

O ato de gerenciar a rede de uma empresa tende a trazer muitos benefícios para os administradores do ambiente de infraestrutura, pois tem como foco monitorar e controlar todos os serviços do ambiente de TI da empresa

O monitoramento de redes, ou, monitoramento de rede é o processo responsável por medir em tempo real, 24 horas por dia, os recursos das redes de computadores, desta forma, é o responsável por detectar anomalias de performance e disponibilidade. Para isso é implementada uma ferramenta de monitoramento de rede que aumentará a disponibilidade do ambiente, pois, a capacidade de prever os incidentes traz vantagem para a área de TI. Atuando assim preventivamente na infraestrutura de TI, aplicações e demais componentes (UNIREDE, 2020).

A gerência de redes é o monitoramento de qualquer estrutura física e/ou lógica de uma rede. É importante esse gerenciamento para que se alcance um bom fluxo no tráfego das informações, que os recursos sejam corretamente utilizados e não sobrecarregados, e que os dados sejam transportados com confiabilidade e segurança.

O gerenciamento de rede envolve a implementação, a integração e a coordenação de elementos de *hardware*, *software* e recursos humanos para realizar o monitoramento,

testes e configuração com o objetivo de controlar os recursos da rede, para satisfazer as exigências de desempenho e melhorar a qualidade de serviços (KUROSE, 2013).

Com as ferramentas corretas para o gerenciamento e monitoramento da rede de uma corporação, são gerados inúmeros cenários onde o administrador se favorece para gerar a melhor combinação e soluções de problemas.

O Zabbix é uma ferramenta de monitoramento de redes, servidores e serviços, pensada para monitorar a disponibilidade, experiência de usuário e qualidade de serviços. A arquitetura do Zabbix e a flexibilidade dos módulos permitem que a ferramenta seja utilizada para o monitoramento convencional (vivo/morto *on/off*), acompanhamento de desempenho de aplicações, análise de experiência de usuário e análise de causa raiz em ambientes complexos, através do servidor Zabbix e as regras de correlacionamento**.**

A ferramenta de monitoramento de redes Zabbix oferece uma *interface* 100% *Web* para administração e exibição de dados. Os alertas do sistema de monitoramento Zabbix podem ser configurados para utilizar vários métodos de comunicação, como SMS, *e-mail* e abertura de chamados em sistemas de *helpdesk*. O sistema permite ainda que ações automáticas como, por exemplo, *restart* de serviços sejam executados a partir de eventos (ZABBIX SIA, 2020).

O Zabbix permite monitoramento *agentless* (sem agentes) para diversos protocolos e conta com funções de *auto-discovery* (descoberta automática de itens) e *low level discovery* (descoberta de métricas em itens monitorados).

Esta ferramenta é uma solução de monitoração integrada, que provê recursos de monitoração em um único pacote, capaz de coletar dados como, verificação de disponibilidadee desempenho, suporta Protocolo Simples de Gerenciamento de Redes - *Simple Network Managent Protocol* (SNMP), possui alertas altamente confiáveis, gráficos sob demanda, histórico e armazenamento de dados etc(ZABBIX SIA, 2020).

1.1 Objetivo Geral

Demonstrar o funcionamento do monitoramento da disponibilidade de portas e serviços em um Servidor *Web*.

1.2 Objetivo Específicos

- Aprofundar o conhecimento na ferramenta Zabbix;
- Aplicar a ferramenta na identificação de falhas;
- Utilizar a ferramenta Zabbix na identificação e tratativa de incidentes com menor impacto possível.

1.3 Procedimentos Metodológicos

Este trabalho tem sua natureza um resumo de assunto, pois agrega, uma análise e discute sobre conhecimentos me informações já publicadas. O resumo de assunto busca apenas sistematizar uma área de conhecimento, usualmente indicando sua evolução histórica e estado da arte, e, portanto, adequado aos cursos de graduação (WAZLAWICK, 2014).

1.4 Estrutura da Monografia

No Capítulo 2, é realizada o respaldo sobre a teoria do Gerenciamento de Redes, com seus principais conceitos e sua importância.

No Capítulo 3, aborda sobre a ferramenta de monitoramento Zabbix.

No Capítulo 4, demonstra como o Gerenciamento de Rede auxilia na identificação de incidentes em um Servidor *Web*.

No Capítulo 5, conclusão sobre a importância Gerenciamento de Rede utilizando a ferramenta Zabbix.

2 GERENCIAMENTO DE REDES

2.1 O que é gerenciamento de rede?

O Gerenciamento de Redes é o processo de controle de uma rede de computadores visando maximizar sua eficiência e produtividade das atividades e recursos, estes recursos podem ser roteadores, *modems* etc. Aos que utilizam de protocolos partilhados na rede para garantir a segurança e confiabilidade das informações.

2.2 Modelo ISO de gerenciamento de redes

Conforme KURUSE e ROSS, a Organização Internacional para Padronização - *International Organization for Standardization* (ISO) o gerenciamento de rede é definido em cinco áreas:

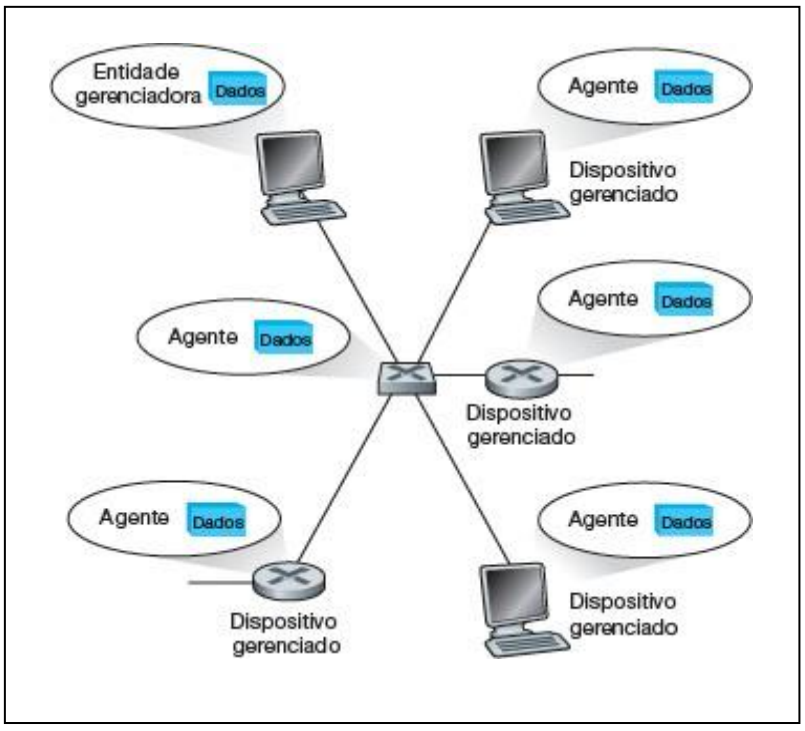
- Gerenciamento de desempenho: visa monitorar, classificar e manter o administrador informado sobre o desempenho de diversos componentes conectados à rede;
- Gerenciamento de falhas: tem como objetivo detectar falhas em serviços ou dispositivos e reagir de acordo com necessidade, além da possibilidade de reação antecipada a possíveis falhas no futuro;
- Gerenciamento de configuração: sustenta o administrador de informações devidamente em relação os dispositivos conectados à rede, e quais recursos estão consumindo, além de suas configurações de *hardware* e de *software*;
- Gerenciamento de contabilização: possibilita ao administrador manter o controle de acesso de usuários a determinados dispositivos e serviços, e mantem também o controle de quotas e utilização de serviços;
- Gerenciamento de segurança: tem como objetivo manter o controle de acesso utilizando políticas de rede bem definidas, além do uso de *firewalls* para obter controle de acesso externo.

2.3 Arquitetura do Gerenciamento

A Figura 1 apresenta os componentes do modelo utilizado para o gerenciamento de redes o Protocolo de Controle de Transmissão/Protocolo de *Internet* - *Transmission Control Protocol/Internet Protocol* (TCP/IP), que é formado pelos seguintes elementos:

- Entidade de Gerenciamento – *Network Management Station* (NMS);
- Dispositivo Gerenciado;
- Agente de Gerenciamento;
- Base de Informações Gerenciamento – *Management Information Base* (MIB);
- Protocolo Simples de Gerenciamento de Rede - *Simple Gateway Monitoring Protocol* (SNMP).

Figura 1 - Principais Componentes de uma arquitetura de rede



Fonte: KUROSE, 2013

A entidade gerenciadora é uma aplicação que em geral tem um ser humano no circuito e que é executada em uma estação central de gerenciamento de rede no Centro de operações de rede - *Network Operation Center* (NOC). Ela é o centro da atividade; ela controla a coleta, o processamento, a análise e/ou a apresentação de informações de gerenciamento de rede.

O dispositivo gerenciado é um equipamento de rede (incluindo seu *software*) que se encontra em um ambiente gerenciado. Para um dispositivo gerenciado pode-se ter um objeto gerenciado. Estes são, na verdade, as peças de *hardware* propriamente ditas que estão juntamente integradas ao dispositivo gerenciado, por exemplo, uma placa de *interface* de rede.

O agente de gerenciamento, é um *Software* instalado no dispositivo a ser gerenciado, o agente responde às solicitações de informações e de ações da estação de gerenciamento, que fornecendo assincronamente informações importantes que não foram solicitadas por esta estação.

Cada recurso a ser gerenciado são representados como objetos com uma classe específica, e a coleção objetos é referenciada como MIB.

A forma de comunicação entre a estação de gerenciamento e o agente é definido pelo protocolo de gerenciamento de rede, o SNMP.

2.4 ESTRUTURA DE GERENCIAMENTO PADRÃO *INTERNET*

A estrutura de gerenciamento é constituída das seguintes partes (KUROSE e ROSS, 2013):

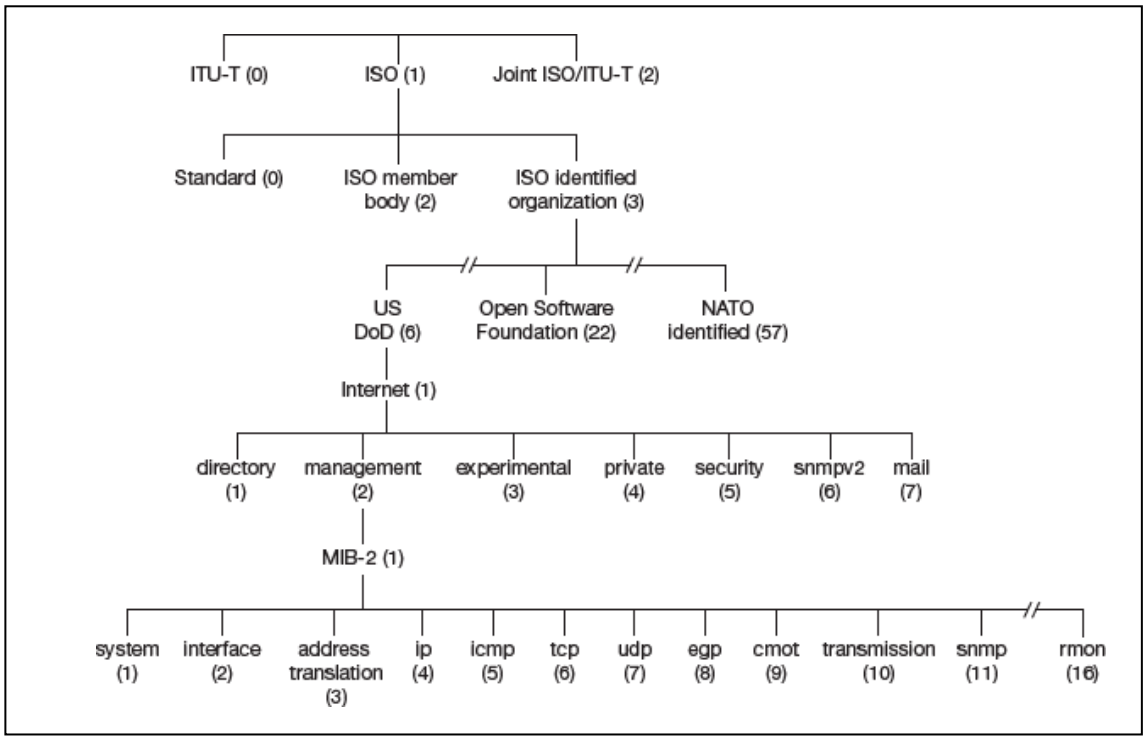
- MIB;
- Estrutura de Informações de Gerenciamento - *Structure of Management Information* (SMI);
- Protocolo SNMP;
- Capacidades de segurança e de administração

2.4.1 Base de Informações de Gerenciamento – MIB

A MIB pode ser imaginada como um banco virtual de informações que guarda objetos gerenciados cujos valores, coletivamente, refletem o “estado” atual da rede. Esses valores podem ser consultados e/ou definidos por uma entidade gerenciadora por meio do envio de mensagens SNMP ao agente que está rodando em um dispositivo gerenciado em nome da entidade gerenciadora. Para se comunicar com objeto, utiliza-se o padrão Organização Internacional para Padronização - *International Organization for Standardization*(ISO) no formato de árvore, em que cada uma de suas ramificações possuem um número e nomenclatura, e os objetos aos quais são as folhas recebem um

Identificador Único - *Object Identifier* (OID), como mostra a Figura 2 (MAURO e SCHMIDT, 2005).

Figura 2 - Árvore de identificadores de objetos ASN.1



Fonte: KUROSE, 2013

A Figura 3 apresenta uma das trilhas de um dos objetos MIB do grupo *system*. Onde cada objeto recebe sua identificação (OID), seguido pelo nome, tipo de dado da SMI e sua descrição detalhada pelo Pedido de Comentários - *Request for Comments* (RFC) 1213.

Figura 3 - Objetos gerenciados no grupo system da MIB-2

Identificador de objeto	Nome	Tipo	Descrição (segundo RFC 1213)
1.3.6.1.2.1.1.1	sysDescr	OCTET STRING	"Nome completo e identificação da versão do tipo de hardware do sistema, do sistema operacional do software e do software de rede."
1.3.6.1.2.1.1.2	sysObjectID	OBJECT IDENTIFIER	ID atribuído pelo fabricante do objeto fornece um meio fácil e não ambíguo para determinar que tipo de objeto está sendo gerenciado.
1.3.6.1.2.1.1.3	sysUpTime	TimeTicks	"O tempo (em centésimos de segundo) desde que a parte de gerenciamento de rede do sistema foi reinicializada pela última vez."
1.3.6.1.2.1.1.4	sysContact	OCTET STRING	"A pessoa de contato para esse nó gerenciado, juntamente com a informação sobre como contatá-la."
1.3.6.1.2.1.1.5	sysName	OCTET STRING	"Um nome atribuído administrativamente para esse nó gerenciado. Por convenção, esse é o nome de domínio totalmente qualificado do nó."
1.3.6.1.2.1.1.6	sysLocation	OCTET STRING	"A localização física do nó."
1.3.6.1.2.1.1.7	sysServices	Integer32	Um valor codificado que indica o conjunto de serviços disponível no nó: aplicações físicas (por exemplo, um repetidor), de enlace/sub-rede (por exemplo, ponte), de Internet (por exemplo, gateway IP), fim a fim (por exemplo, hospedeiro).

Fonte: KUROSE, 2013

2.4.2 Estrutura de Informações de Gerenciamento – SMI

A SMI é a linguagem usada para definir as informações de gerenciamento que residem em uma entidade gerenciada de rede. Essa linguagem de definição e necessária para assegurar que a sintaxe e a semântica dos dados de gerenciamento de rede sejam bem definidas e não apresentem ambiguidade. A Chamada Remota de Procedimento - Remote Procedure Call(RFC) 2578 especifica os tipos de dados básicos da linguagem SMI de definição de módulos MIB. Embora a SMI seja baseada na linguagem de definição de objetos notação de sintaxe abstrata 1- *Abstract Syntax Notation One* (ASN.1), como mostra a Figura 2 (KUROSE e ROSS, 2013).

A Figura 4 organiza os dados básicos do SMI de maneira que a primeira coluna informa o tipo do dado utilizado e a segunda coluna informa a descrição dele.

Figura 4 - Tipos de dados Básicos da SMI

Tipo de dado	Descrição
INTEGER	Número inteiro de 32 bits, como definido em ASN.1, com valor entre -2^{31} e $2^{31}-1$, inclusive, ou um valor de uma lista de valores constantes possíveis, nomeados.
Integer32	Número inteiro de 32 bits, com valor entre -2^{31} e $2^{31}-1$, inclusive.
Unsigned32	Número inteiro de 32 bits sem sinal na faixa de 0 a $2^{32}-1$, inclusive.
OCTET STRING	Cadeia de bytes de formato ASN.1 que representa dados binários arbitrários ou de texto de até 65.535 bytes de comprimento.
OBJECT IDENTIFIER	Formato ASN.1 atribuído administrativamente (nome estruturado); veja a Seção 9.3.2.
IPaddress	Endereço Internet de 32 bits, na ordem de bytes da rede.
Counter32	Contador de 32 bits que cresce de 0 a $2^{32}-1$ e volta a 0.
Counter64	Contador de 64 bits.
Gauge32	Número inteiro de 32 bits que não faz contagens além de $2^{32}-1$ nem diminui para menos do que 0.
TimeTicks	Tempo, medido em centésimos de segundo, transcorrido a partir de algum evento.
Opaque	Cadeia ASN.1 não interpretada, necessária por compatibilidade com versões anteriores.

Fonte: KUROSE, 2013

2.4.3 Protocolo SNMP

O protocolo tem a sua ideia inicial a flexibilidade e clareza em sua implementação, sua especificação é definida pela RFC 1157.

Pelo contrário do que o nome do protocolo SNMP sugere, o gerenciamento de rede na *Internet*, é muito mais do que um protocolo que transporta os dados do gerenciamento entre a entidade gerenciadora e seus agentes, onde SNMP passou a ser mais complexo do que seu nome vem a sugerir. O protocolo que tem sua base formada pelo Protocolo de Monitoramento do Gateway Simples - *Simple Gateway Monitoring Protocol* (SGMP). O SGMP foi traçado por uma equipe de pesquisadores, usuários e administradores universitários de rede, de que maneira a experiência com o protocolo proporcionou que eles idealizassem e oferecessem o SNMP em poucos meses (KUROSE e ROSS, 2013).

Desde então, o SNMP vem progredindo do SNMPv1 para o SNMPv2 até chegar na sua versão mais atual SNMPv3. Onde na sua última versão foi criado para suprir uma necessidade padronização que se fez necessária com as várias variações do SNMPv2

quem tentavam criar soluções de segurança para o protocolo. As melhorias implementadas autenticação, criptografia, controle de acesso e proteção contra-ataques.

2.5 Modos Operacionais SNMP

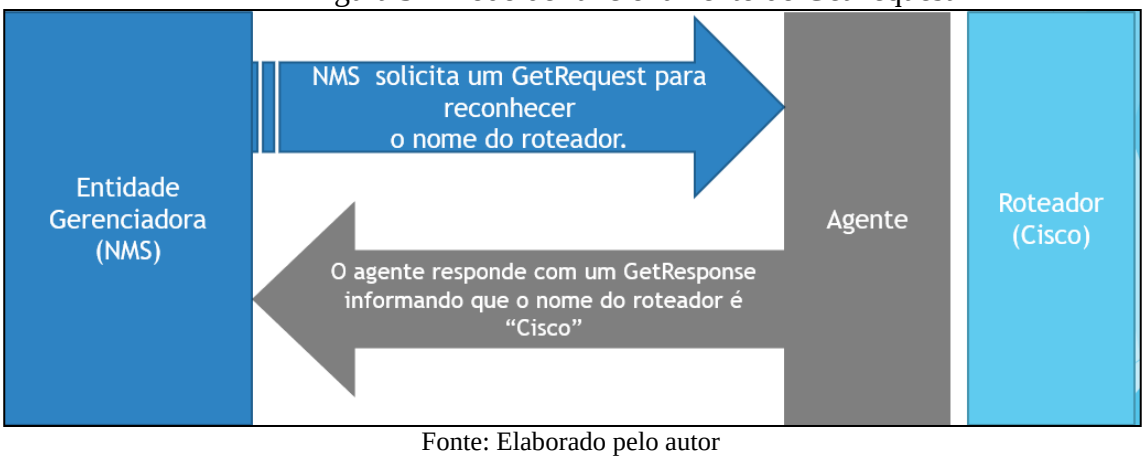
O SNMP desfruta das seguintes operações que serão caraterizadas ao decorrer deste trabalho:

- *GetRequest;*
- *SetRequest;*
- *GetNextResponse;*
- *GetBulk;*
- *Trap.*

2.5.1 GetRequest

O comando *GetRequest* é uma requisição do gerente ao agente para que o valor de uma variável ou uma lista de variáveis seja retornado. As variáveis-alvo são especificadas no campo Enlace de variáveis. O agente então retorna a requisição com uma mensagem do tipo *Response*, com os valores correntes das variáveis requisitadas. A Figura 5 mostra uma solicitação dessa operação.

Figura 5 - Modo de funcionamento do GetRequest

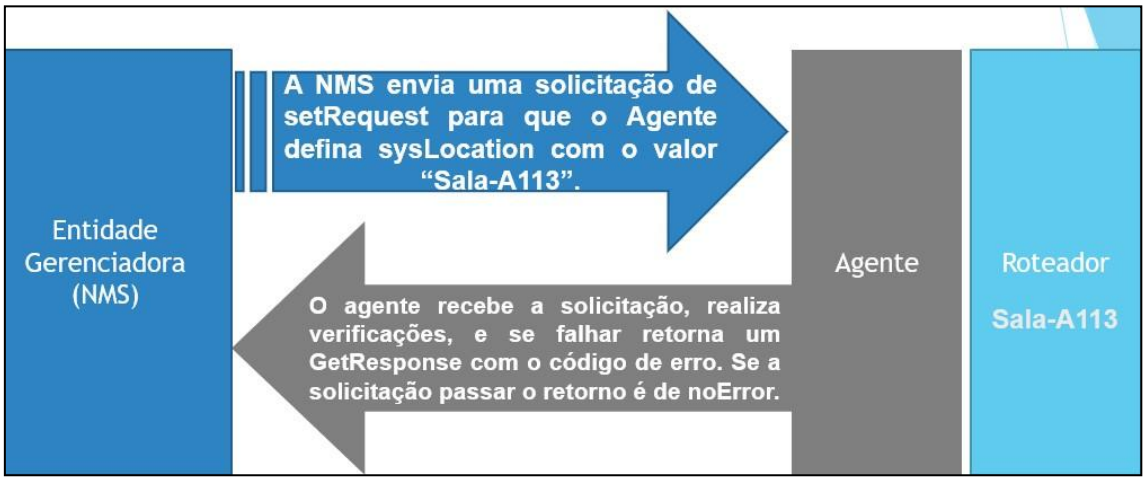


Fonte: Elaborado pelo autor

2.5.2 SetRequest

São objetos aos quais tem permissão de *read-write* ou *read-only*, (em português: leitura e escrita ou apenas leitura), o *SetRequest* é utilizado para gravar valores, mensagens em objetos ou adicionar uma nova linha em determinada tabela. Portanto, esse comando define um novo valor de uma variável ou de uma lista de variáveis. No exemplo da Figura 6, a NMS envia uma solicitação de *SetRequest* para gravar no objeto *sysLocation*, que é um objeto que armazena a localização física do dispositivo. O agente analisa se o objeto é *read-write*, leitura e escrita, e se a NMS possui permissões para gravar informações no roteador, caso ocorra tudo corretamente, o agente grava o texto "Atlanta, GA", e então retorna para a NMS um comando *GetResponse* com o valor "*noError*", indicando que o valor do objeto foi alterado com sucesso. A Figura ¨6 demonstra uma solicitação dessa operação.

Figura 6 - Modo de funcionamento do SetRequest



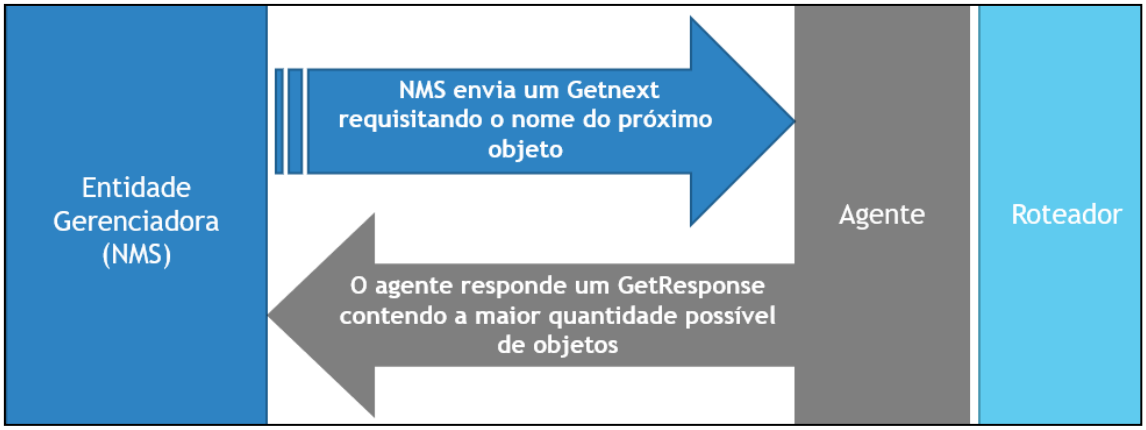
Fonte: Elaborado pelo autor.

2.5.3 GetNextResponse

Esta operação permite que uma sequência de comandos seja emitida para que seja recuperado um grupo de valores de um MIB específico. Para cada objeto MIB a ser recuperado, são geradas solicitações separadamente de *get-next* e *get-response*. O comando *get-next* atravessa uma sub-árvore em ordem lexicográfica. O agente requisita a primeira solicitação de informações do objeto filho na sua posição zero da ramificação, após receber o *Get-Response* o agente envia uma requisição com um valor correspondente ao próximo objeto e assim sucessivamente até que o agente retorne uma mensagem de erro informando que aquele ramo não possui mais objetos filhos. Após receber essa

mensagem o gerente para de solicitar requisições. A Figura 7 demonstra uma solicitação dessa operação.

Figura 7 - Modo de funcionamento do GetNextRequest

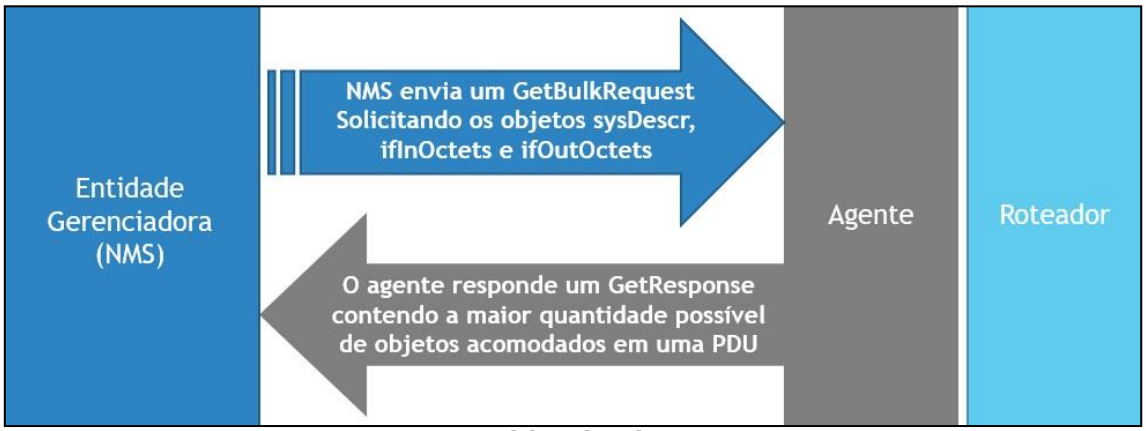


Fonte: Elaborado pelo autor.

2.5.4 GetBulk

O *GetBulk* é uma operação no qual a NMS envia uma mensagem para o agente solicitando vários objetos instruindo para o agente que envie a maior quantidade de objetos solicitados. No exemplo da Figura 8 foram enviadas as solicitações para os objetos *sysDescr*, *ifInOctets* e *ifOutOctets*.

Figura 8 - Modo de funcionamento do *GetBulkRequest*

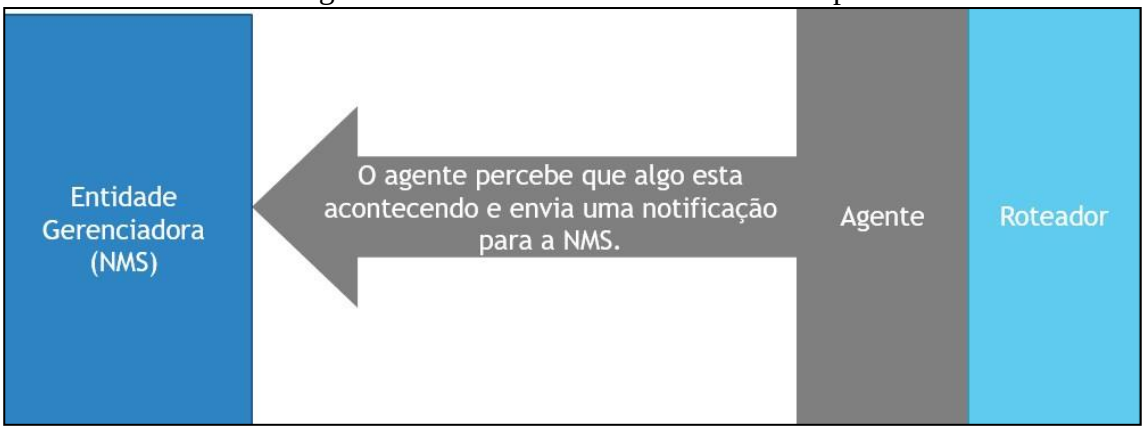


Fonte: Elaborado pelo autor.

2.5.5 Trap

Uma *trap* não necessita de resposta por parte do gerente, pois ela pode ser considerada uma via de mão única. No exemplo da Figura 9 o agente de um roteador envia uma mensagem de *Trap* para a NMS, que executa uma validação do dispositivo que enviou e emite um alerta para o gerente da rede tomar medidas cabíveis diante do ocorrido. A Figura 9 demonstra o envio da mensagem desta operação.

Figura 9 - Modo de funcionamento do Trap



Fonte: Elaborado pelo autor

2.6 ARQUITETURAS DE NMS

A arquitetura de NMS é a forma por uma estrutura de gerenciamento da rede a qual será construída, ou seja, a forma em que os dados de gerenciamento serão processados para coleta. Para a sustentação da rede devidamente estruturada o gerenciador deve elaborar uma arquitetura harmônica com a necessidade do ambiente que será implementado. Em uma circunstância na qual se possui uma um Matriz e uma filial deve-se coletar informações de gerenciamento, sendo assim necessário a aplicação de uma NMS central na matriz, mesmo que não possua filiais em sua infraestrutura. Em circunstâncias onde a arquitetura poderá ser distribuída, possuindo mais de um NMS, ou seja, uma central sendo a Matriz e outras em filiais. E para melhorar a segurança e aliviar o tráfego, ainda pode-se implantar arquiteturas do tipo híbridas que utilizam uma rede própria para troca de informação.

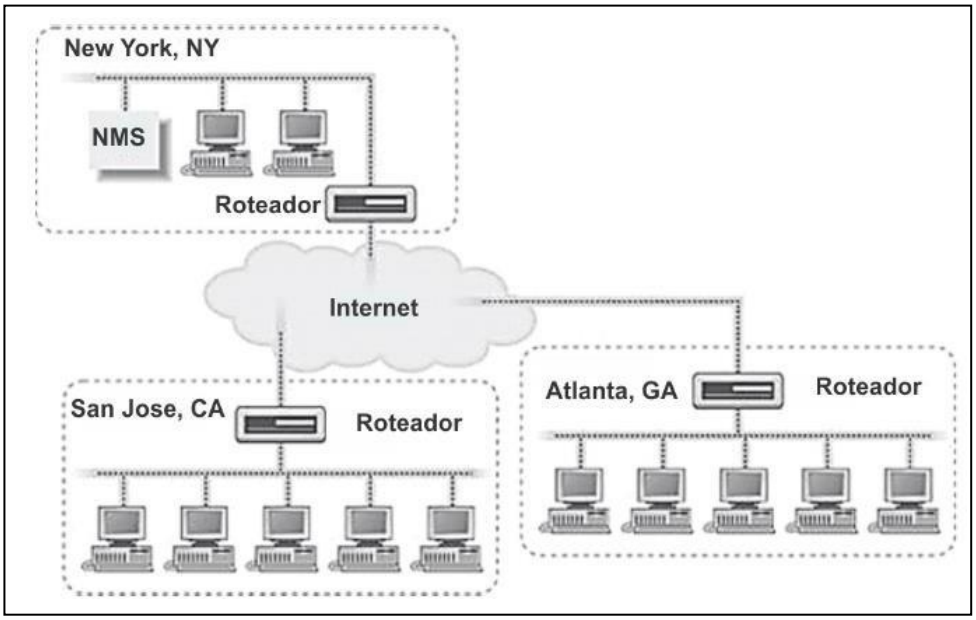
2.6.1 Arquitetura Centralizada

A arquitetura Centralizada utiliza somente uma única NMS que gerencia toda a rede mesmo que parte da rede esteja distante fisicamente, como no caso em que a empresa possui uma matriz e diversas filiais, é necessita receber dados de todos os equipamentos gerenciados.

A Figura 10 exemplifica esse modelo de arquitetura, o gerenciamento iniciasse na matriz de New York e se dispersa nas filiais localizadas em Atlanta e São José, a matriz dispõe de uma entidade gerenciadora, dispositivos gerenciados, porém em suas filiais não possuem de uma entidade gerenciadora somente os dispositivos gerenciados. Todas as informações sobre os objetos aos quais são gerenciados são requeridos pela matriz em New York para suas filiais em Atlanta e São José, essa solicitação pode gerar um consumo de alto impacto no tráfego de rede entre matriz e filiais. Esse tipo de comunicação expõe vulnerabilidade por utilizar-se da *internet* onde as informações transmitidas de filial a matriz por uma rede aberta, possibilitando a oportunidade de sofrer ataques de captura de pacotes onde o atacante pode capturar informações de alguns dispositivos.

Por outro lado, em empresas que possuem poucos dispositivos e filiais próximas fisicamente, a arquitetura centralizada pode ser considero vantagem, pois com apenas uma NMS os gatos de implementação são menores e o deslocamento é menor, se necessário, é feito em menor distância, mas não elimina a vulnerabilidade na transmissão de dados de gerência usando a *Internet* (MAURO & SCHMID, 2001).

Figura 10 - Arquitetura de NMS Centralizada



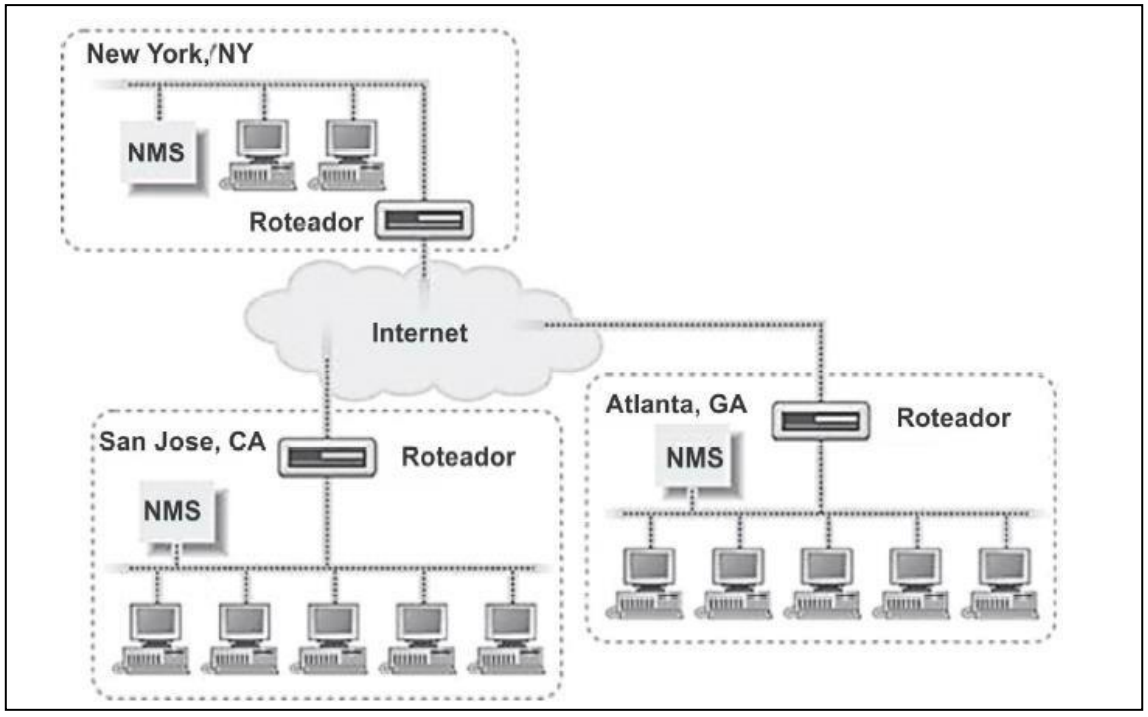
Fonte: MAURO, 2005

2.6.2 Arquitetura Distribuída

A arquitetura distribuída tem como objetivo reduzir o tráfego de itens gerenciados, pois diferentemente da arquitetura centralizada que possui somente uma NMS, esta possui uma entidade gerenciadora para cada filial, e na matriz uma NMS central, que pode receber apenas relatórios periódicos, pelo motivo que a descentralização da coleta de dados possibilita que uma equipe secundária seja alocada em cada filial. Essa arquitetura não desconsidera a utilização da *Internet* para transferência de dados para a gerência, porém eles são transmitidos em menor frequência, porque a entidade localizada na filial envia somente relatórios periódicos. Isso potencializa o tempo de resposta, pelo motivo de cada filial possuir uma equipe responsável pelo gerenciamento da rede de forma local, ou seja, não existe a dependência do gerente da matriz para tomar todas as decisões.

A Figura 11 é a forma como a arquitetura distribuída pode ser implantada na empresa descrita no exemplo mostrado em que tanto as filiais quanto a matriz possuem uma NMS. Assim, quando ocorrer algum problema em San José ou Atlanta não será necessário deslocar uma pessoa ou equipe para resolver tal eventualidade, pois cada filial possui um responsável pela NMS.

Figura 11 - Arquitetura de NMS Distribuída

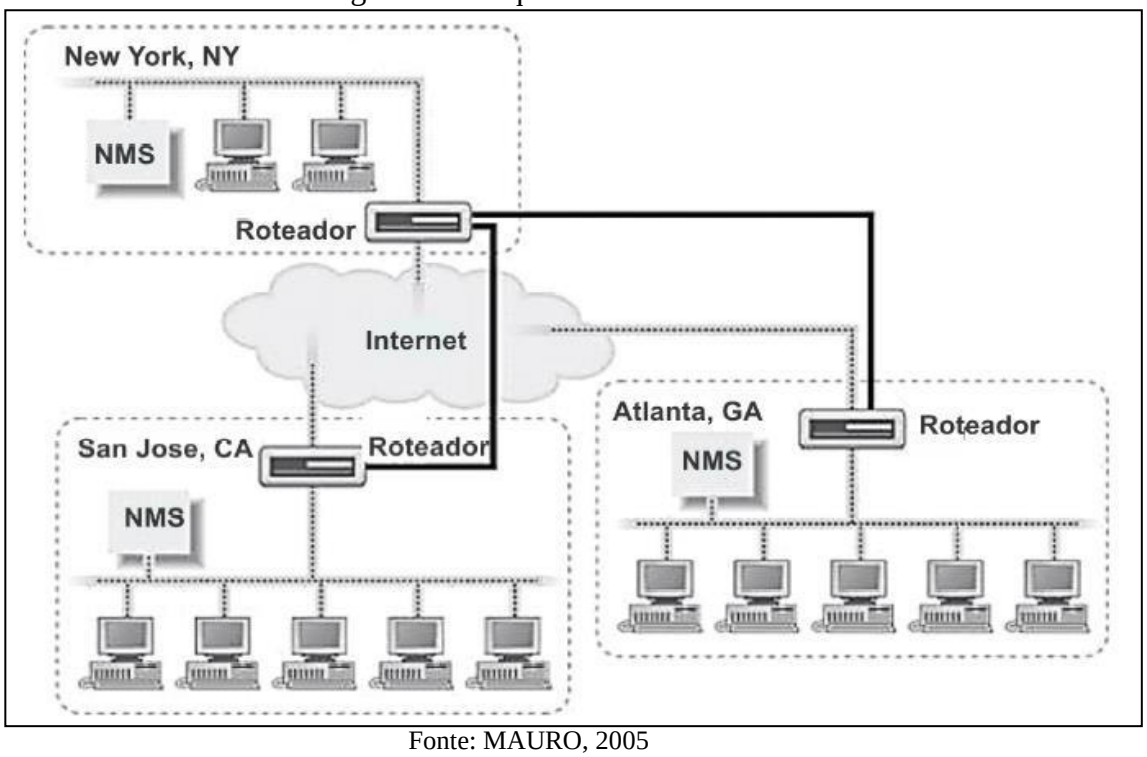


Fonte: MAURO, 2005

2.6.3 Arquitetura Híbrida

A NMS Hibrida dispõe de um *link* de dados dedicado para interligar a matriz e suas filiais. Essa arquitetura pode gerar um gasto maior dependendo da forma que foi implementada, pois será necessário contratar um serviço secundário de dados, que as operadoras de telecomunicações fornecem por um valor mais alto do que os *links* de *Internet*, este serviço é nomeado como ponto-a-ponto - *peer-to-peer* (P2P). A Figura 12 demonstra o funcionamento da arquitetura híbrida, que apresenta a interligação da matriz com suas filiais.

Figura 12 - Arquitetura de NMS Híbrida



Fonte: MAURO, 2005

3 FERRAMENTA DE GERENCIAMENTO

O Zabbix é um *software*, distribuído sob a Licença Publica Geral – *General Public License* (GPL) criado por Alexei Vladishev, é uma ferramenta de monitoramento de redes, servidores e serviços, Maquinas Virtuais (VM) e serviços em nuvem, pensada para monitorar a disponibilidade, experiência de usuário e qualidade de serviços. A ferramenta de monitoramento de redes Zabbix oferece uma *interface* 100% *Web* para administração e exibição de dados.

O Zabbix usa um flexível mecanismo de mídia que são os canais de distribuição utilizados para enviar notificações e alertas no Zabbix. Pode-se dizer que são através delas que o Zabbix se comunica com o mundo externo.

As mídias podem ser de vários tipos:

- *E-mail*;
- Serviço de Mensagens Curtas - Short Message Service (SMS);
- Telegram;
- Slack;
- Sistema de Solicitação de Tíquetes de Código Aberto - Open-source Ticket Request System (OTRS);
- *Scripts* customizados;

3.1 Pré-requisitos de sistema

3.1.1 Requisitos de *hardware*

Para a implementação do Zabbix S*erver* é recomendado pela ZABBIX SIA *hardware* com no mínimo 256 Megabyte(MB) de espaço disponível e 128 MB de Memória de Acesso Aleatório - Random Access Memory(RAM), porém esses requisitos iram depender da quantidade de *hosts* e de parâmetros monitorados, em especial o espaço que deve ser planejado para armazenamento das informações coletadas e armazenadas no servidor de banco de dados. As Tabelas 1 e 2 dmonstram os requisitos de *hardware* e plataformas suportadas (ZABBIX SIA, 2020).

Tabela 1 - Tabela de pré-requisitos para instalação do Zabbix

Nome	**Plataforma**	**CPU/Memória**	**SGDB**	***Hosts* Monitorados**
Pequeno	CentOS	Virtual Appliance	MySQL InnoDB	100
Médio	CentOS	2 CPU cores/2GB	MySQL InnoDB	500
Grande	RedHat Enterprise Linux	4 CPU cores/8GB	RAID10 MySQL InnoDB ou PostgreSQL	> 500
Muito grande	RedHat Enterprise Linux	8 CPU cores/16GB	RAID10 rápido MySQL InnoDB ou PostgreSQL	>1 0000

Fonte: Zabbix SIA, s.d.

Tabela 2 - Plataformas suportadas

Plataforma	Zabbix Server	Zabbix Agent
Linux	X	X
IBM AIX	X	X
FreeBSD	X	X
NetBSD	X	X
OpenBSD	X	X
HP-UX	X	X
Mac OS X	X	X
Solaris	X	X
Windows: 2000, Server 2003, XP, Vista, Server 2008, 7, 8, 10, Server 2012, 2016, 2019	-	X

Fonte: Zabbix SIA, s.d.

O Zabbix pode rodar em várias outras variantes do UNIX.

3.1.2 Requisitos de *software*

Os requisitos de *software* podem ter distintas configurações de acordo com a versão a ser implementada do Zabbix, os dados a serem apresentados são referentes a versão implementada para realização deste trabalho.

A Tabela 3 ilustra quais bancos de dados e suas versões podem ser utilizados na instalação.

Tabela 3 - Bancos de dados suportados

Software	**Versão**
MySQL	5.0.3 ou superior
Oracle	10g ou superior
PostgreSQL	8.1 ou superior
SQLite	3.3.5 ou superior
IBM DB2	9.7 ou superior

Fonte: Zabbix SIA, s.d.

A Tabela 4 ilustra os demais requisitos de *software*.

Tabela 4 - Requisitos de *software*

Software	**Versão**
Apache	1.3.12 ou superior
PHP	5.4.0 ou superior

Fonte: Zabbix SIA, s.d.

Referente a navegadores *Web o* suporte a *Cookies* e Java *Script* deve estar habilitado. As últimas versões do Google Chrome, Mozilla Firefox, Microsoft *Internet Explorer* e Opera são suportadas. Outros navegadores (Apple Safari, Konqueror) também podem funcionar.

3.2 Conceitos Zabbix

3.2.1 Definições zabbix

O Zabbix é uma solução de monitoração integrada, que possui termos que definem sua composição:

- *Host*: dispositivo ao qual precisa ser monitorado, através de Protocolo de Internet - *Internet Protocol* (IP)/ Sistema de Nomes de Domínios - *Domain Name System* (DNS)*;*
- Grupo de *hosts*: agrupamento logico de *hosts,* utilizado para especificar *hosts* ou *templates* pertencentes ao grupo específico;
- Item: um dado específico ao qual se deseja receber de um *host* monitorado, uma métrica de dados;
- *Trigger*: expressão lógica que determina o início da mudança de estado de algum item, normalmente utilizado para reportar incidentes;

- Evento: ocorrência de algum incidente ao qual tenha uma relevância ao gerenciador. Esses incidentes podem ser gerados por como uma mudança de estado em uma *trigger* ou a descoberta ou o auto registro de um *host*;
- Ação: forma estabelecida para reação à um evento específico;
- Escalonamento: um cenário preparado para a execução de alguma tratativa como, envio de notificações ou execução de comandos remotos;
- Mídia: forma de notificar algo ocorrido, um canal de distribuição;
- Notificação: mensagem sobre algum evento ocorrido no ambiente, essas mensagens são enviadas através das mídias;
- Comando remoto: um comando pré-definido que será automaticamente executado quando determinada condição ocorrer em um *host* monitorado
- *Template*: um conjunto de entidades itens, *triggers*, gráficos, telas, aplicações, regras, cenários *Web*, aos quais serão aplicados a um ou a um conjunto de *host*;
- Aplicação: conjunto de itens;
- Cenário *Web:* uma ou mais requisições do Protocolo de Transferência de Hipertexto (*Hypertext Transfer Protocol* - HPPT) ou Protocolo de Transferência de Hipertexto (*HyperText Transfer Protocol Secure* – HTTPS), utilizados para verificar a disponibilidade de um site ou página *Web* específica;
- *Front-end: interface Web* utilizada para as configurações e acesso ao ambiente de monitoração do Zabbix;
- API Zabbix: a API Zabbix fornece *interface* programável para atualizações em massa, integração com ferramentas de terceiros e outros recursos como, uma Chamada Remota de Procedimento (*Remote Procedure Call* – RPC), Notação de Objetos JavaScript (*JavaScript Object Notation* – JSON;
- Servidor Zabbix: Componente central do Zabbix, responsável por realizar o monitoramento, interage com os proxies e agentes, calcula as mudanças de estado nas *triggers*, envia notificações, controla o repositório central de dados;
- Agente Zabbix: componente instalado nos *hosts* a serem monitorados, usado para monitorar ativamente seus recursos e aplicações. Monitora ativamente recursos do dispositivo como, Unidade Central de Processamento - *Central Process Unit* (CPU), memória, discos, partições de disco, serviços em execução e *interfaces* de rede, emitindo alertas em caso de falha de algum

recurso. Os alertas podem ser criados de acordo com a necessidade do administrador da rede;

- *Proxy* Zabbix: Componente com capacidade de realizar a coleta de dados no lugar do Servidor Zabbix, distribuindo a carga de processamento. (ZABBIX SIA, 2020).

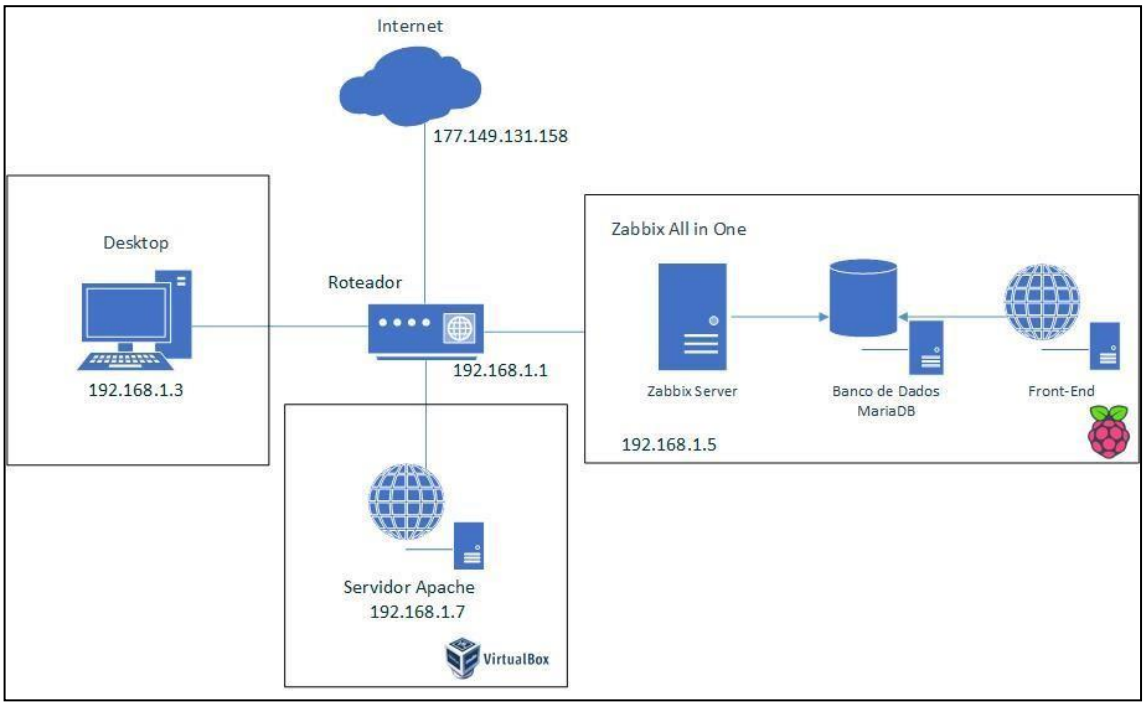
4 USO DA GERÊNCIA DE REDE NA IDENTIFICAÇÃO DE INCIDENTES EM UM SERVIDOR *WEB*

4.1 Apresentação do Ambiente

O ambiente ao qual foi criado será monitorado no intuído de demonstrar a incidência de disponibilidade de serviço e o restabelecimento deles. O ambiente proposto consiste em um servidor *Web* o qual está sendo utilizado o *Software VirtualBox* para realizar a virtualização, onde está instalado o servidor Apache e recebe o nome de "*Web*server". Há também um servidor Zabbix *All in One* o qual é composto por um banco de dados e uma *interface Web* que está implementado em *Raspberry Pi 3*, um roteador o é um aparelho usado em redes de computadores para o encaminhamento das informações acondicionadas em pacotes de dados, proporcionando conectividade entre os dispositivos e uma máquina *desktop* para demonstrar onde será feita a monitoração utilizando a *interface Web* da ferramenta Zabbix.

A topologia utilizada para elaboração deste ambiente foi do tipo estrela, por possuir uma manutenção simplificada, permitindo adicionar novos dispositivos sem grandes planejamentos. A Figura 13 esboça o desenho do ambiente de rede.

Figura 13 - Topologia de rede



Fonte: Elaborado pelo autor

4.2 Elaboração do monitoramento

Com o ambiente Zabbix predefinido, foi utilizado da descoberta automática de *hosts* que é uma funcionabilidade do Zabbix que varre um *range* de IP definido na regra de descoberta de rede, ao identificar algum *host* o Zabbix gera algumas ações as quais podem ser aplicadas dependendo da configuração realizada pelo administrador, como:

- Envio de notificações;
- Adicionar/Remover *hosts*;
- Ativar/Desativar *hosts*;
- Adicionar/Remover *hosts* a um grupo;
- Associar/Desassociar um *host* a um *template*;
- Executar comandos remotos.

4.3 Disponibilidade de conexão

Com o a crescente as empresas vêm dependendo da *Internet* e dos recursos de TI para operar seus serviços. Por este motivo, qualquer que seja a indisponibilidade pode ser

muito prejudicial aos negócios. Além do prejuízo por negociações possivelmente perdidas, ainda ocorrem problemas de baixa produtividade, já que sem os sistemas, os colaboradores não podem trabalhar até que eles sejam restabelecidos. Portanto, as falhas e indisponibilidades são nocivas ao negócio (EVEO, 2019).

Com a identificação do *host* realizada ele e automaticamente inserido em um grupo de *templates* as quais são compostas por aplicações, itens, *triggers*, gráficos, telas, regras de descoberta e cenários *Web*. Para a criação da *trigger* corretamente foi feito teste locais na rede indisponibilizando o acesso do servidor *Web* aos protocolos HTTP, HTTPS e protocolo de rede criptográfico Protocolo de Rede Criptográfico - *Secure Socket Shell*(SSH), além da sobrecarga em alguns recursos como memória e CPU.

Para as *Triggers* dos protocolos HTTP, HTTPS e SSH foi criada a Template_Teste_Disp_Porta, onde é realizada uma monitoração do tipo simples. Para estes casos, o Zabbix possibilita do uso de um *Simple Check*, checagens que não necessitam de SNMP e nem de uma agente no servidor remoto, essa checagem retorna uma resposta do tipo "sim" ou "não, 0 ou 1. Foi utilizada a expressão (net.tcp.service.perf[service,<ip>,<port>]) a qual verifica a performance de um serviço do Protocolo de Controle de Transmissão -*Transmission Control Protocol*(TCP). A Figura 14 apresenta os itens da *template.*

Figura 14 - Itens *Template*_Teste_Disp_Porta

Name	Triggers	Key	Interval ▲	History	Trends	Type	Applications
Porta HTTP	Triggers 1	net.tcp.service[http,,80]	10s	90d		Simple check	HTTP
Porta HTTPS	Triggers 1	net.tcp.service[https,,443]	10s	90d		Simple check	HTTPS
Porta SSH	Triggers 1	net.tcp.service[ssh,,22]	10s	90d		Simple check	SSH

Fonte: Elaborado pelo autor

Os parâmetros a serem inseridos entre as chaves são:

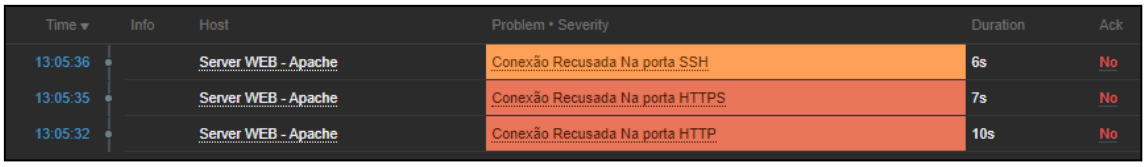
- ***service*** - valores possíveis: SSH, Protocolo de Acesso a Diretório Leve - *Lightweight Directory Access Protocol*(LDAP), Protocolo de Transferência de Correio Simples - *Simple Mail Transfer Protocol*(SMTP), Protocolo de Transferência de Arquivos - *File Transfer Protocol*(FTP), HTTP, Procedimento Operacional Padrão - *Standard Operational Procedure*(POP), Protocolo de Transferência de Notícias da Rede - *Network News Transfer Protocol*(NNTP), Protocolo de acesso a

mensagem da internet - Internet Message Access Protocol (IMAP), TCP, HTTPS, Protocolo de Terminal Virtual - Virtual Terminal Protocol(TELNET);

- ***ip*** - endereço IP ou nome de DNS (por padrão, o IP/DNS do host será utilizado);
- ***port*** - número da porta (se ausente, a porta padrão do serviço será utilizada).

Para a criação das *Triggers* da Template_Teste_Disp_Porta foram configuradas as expressões de acordo com cada protocolo. Os valores definidos para acionamento das *Triggers* foram "0" para quando ocorra a indisponibilidade ao serviço e "1" para quando serviço for normalizado, se caso ocorra algum problema a mesma é disparada alertando o administrador. A Figura 15 demonstra as *Triggers* acionadas.

Figura 15 - *Triggers* de conexão negada do servidor *Web*

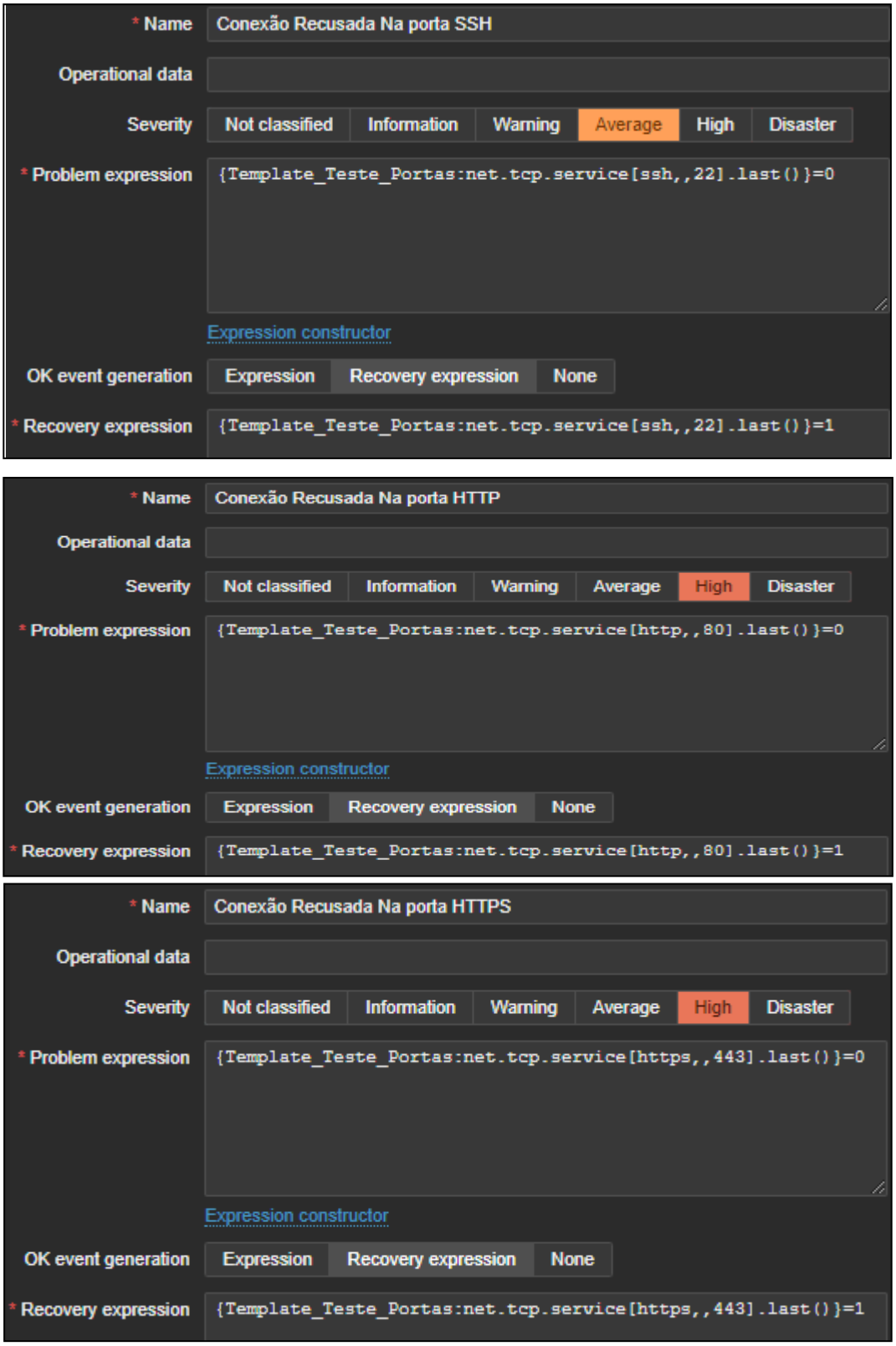


Fonte: Elaborado pelo autor

A Figura 16 mostra como foram definidas as *Triggers* que indicam se a conexão foi recusada à alguma das portas de cada protocolo, e a severidade de cada uma delas. A expressão destas *Triggers* identifica a parti de um binário identifica se existe conexão a porta especifica a qual deseja comunicação onde:

- **0** – Serviço indisponível;
- **1** – Serviço está em execução.

Figura 16 - Criação das *Triggers* de conexão negada do servidor *Web*



Fonte: Elaborado pelo autor

Este procedimento tem como o objetivo validar o reconhecimento de algumas portas de comunicação do servidor *Web*, esse procedimento auxilia o administrador a identificar e criar a tratativa correta para o incidente a ser informado. Tendo as informações em mãos o administrador pode verificar o que pode estar ocasionando a interrupção das conexões.

4.4 Certificado SSL

Foi implementado um *script* para capturar informações sobre o Camada de Soquete Seguro - *Secure Socket Layer* (SSL), que assinala a data de expiração do certificado. Para esta configuração foi feita a configuração dentro de um ambiente do Servidor Apache, a Figura 17 mostra a inclusão do domínio.

Figura 17 - Inclusão de domínio para Apache

```
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin webmaster@localhost
                ServerName www.joaozbx.com
                ServerAlias www.joaozbx.com
                DocumentRoot /var/www/joaozbx
```

Fonte: Adaptado pelo autor

Para gerar o certificado foi utilizado do *OpenSSL*, que nada mais é uma implementação de código aberto dos protocolos SSL e Segurança de Camada de Transporte - *Transport Layer Security* (TLS), possui um conjunto de ferramentas robustas, de nível comercial (OPENSSL.ORG, 2020).

Na sua implementação o certificado *OpenSSL* é assinado pelo seu criador, que poderá ser utilizado em ambientes de desenvolvimento ou qualquer outra coisa que requer o SSL, porém não se indica o uso em ambientes de produção (GENIAR, 2015).

A Figura 18 mostra a execução do comando de gerar o certificado SSL auto assinado, esse tipo de configuração livra de incômodos futuros de quais quer tipo de solicitação e informações futuras.

Figura 18 -Criação do certificado

```
root@webserver:/etc/apache2# openssl req -x509 -sha256 -days 365 -newkey rsa
:2048 -keyout apache.key -out apache.crt -nodes

Generating a 2048 bit RSA private key
.............................+++
.............+++
writing new private key to 'apache.key '
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:BR
State or Province Name (full name) [Some-State]:GO
Locality Name (eg, city) []:Goiania
Organization Name (eg, company) [Internet Widgits Pty Ltd]:PUC GOIAS
Organizational Unit Name (eg, section) []:TI
Common Name (e.g. server FQDN or YOUR name) []:www.joaozbx.com.br
Email Address []:zbxtcc2021@gmail.com

root@webserver:/etc/apache2#
```

Fonte: GENIAR, 2015. Adaptado pelo autor

A Figura 19 exibe o *script* que recolhe as informações do certificado do *site* ao qual se deseja, a qual é recolhida a informação de validade de seu certificado através de do comando *openssl* que é executado é reconhecido por um conjunto de variáveis. Para a busca da informação desejada do *openssl* foi utilizado o *sed,* que obtém somente as linhas que contêm a *string Not After* e a informação de interesse que se constituem em um bloco de texto cercado por "*BEGIN CERTIFICATE*" e "*END CERTIFICATE*". Este bloco contém o texto ao qual o *openssl* busca na segunda chamada de comando (via *pipe).* É como se estivesse realizando o salvamento do arquivo do certificado em um local temporário para depois solicitar sua chamada quando necessário (PISSARRA, 2014).

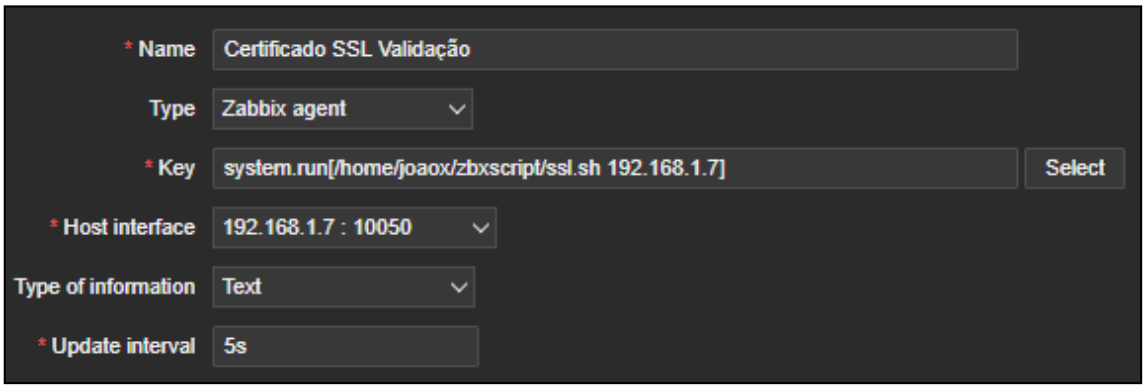
Figura 19 - Script de verificação de certificado

```
#!/bin/bash
    # Parametro primario URL
    [ -z "$1" ] && (

        echo "Execução: `basename "$0"` <site>";
        exit 1;
)
    #Recebe os dados do certificado enviados pelo site
    dados-"`openssl s_client -conect ${1}:443 < /dev/null 2>&1 | \
        openssl x509 -noout -text | \
        sed -n 's/^[ \t]\+Not After : \(.\+ GMT\) .*$/\1/p'`"

    #Informa data/hora UTC e BR.
    echo "\"$1\" expires at $DT ($(date --date="$dados"))"
```

Fonte: PISSARA, 2014. Adaptado pelo autor

Com a configuração feita no servidor Apache, é feita a configuração de um item dentro da ferramenta Zabbix. A Figura 20 mostra esta configuração para execução do item.

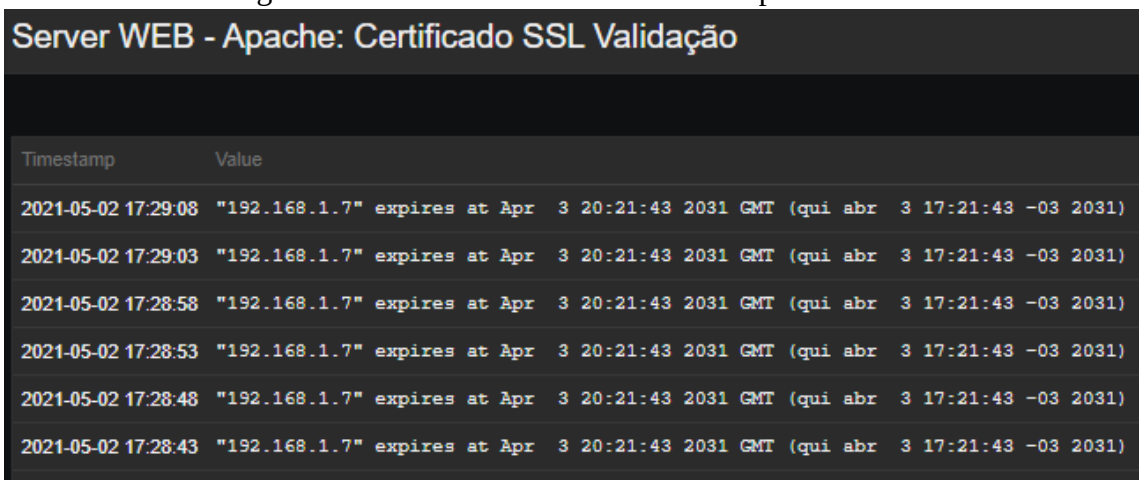
Figura 20 - Criação item responsável pela execução do script



Fonte: Painel Zabbix

A Figura 21 mostra o *script* após sua execução coletando os dados do certificado no qual foi configurado.

Figura 21 - Dados do certificado exibido pelo Zabbix



Fonte: Painel Zabbix

Este procedimento teve como o objetivo demonstrar a utilização de *scripst* em conjunto com a ferramenta Zabbix para a captura de informações sobre a validade do certificado, mostrando se a aplicação a qual se deseja acessar está segura.

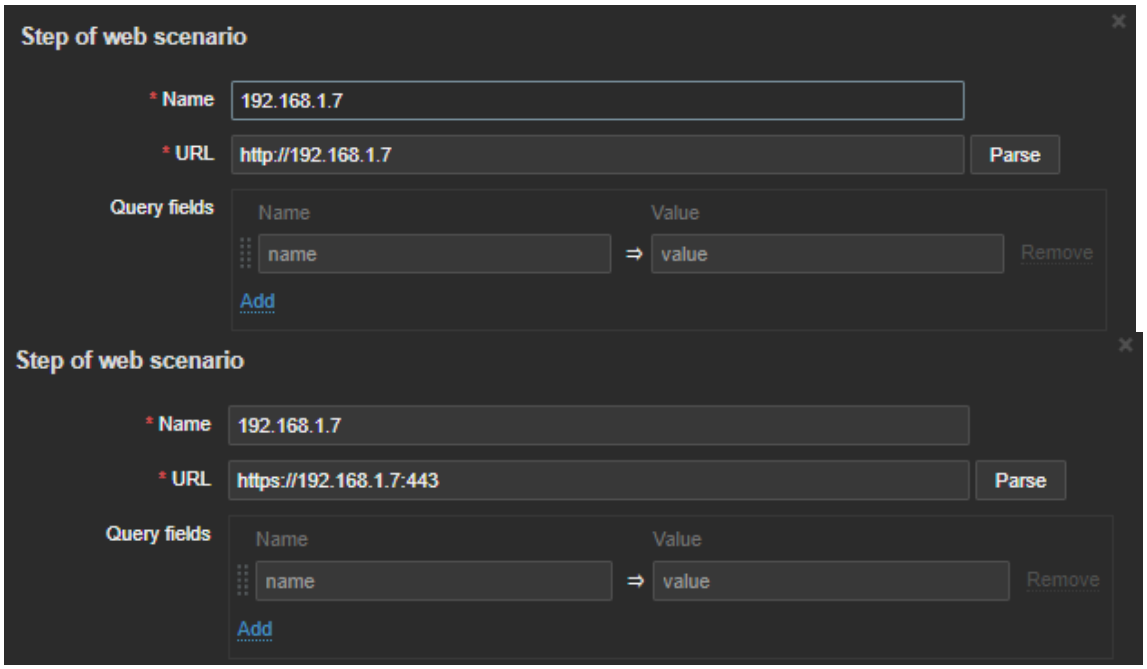
Esse tipo de certificado garante a impossibilidade de leitura de quaisquer dados transferidos entre usuários e sites ou entre dois sistemas. Ele usa criptografia de algoritmos para "embaralhar" dados em trânsito, o que impede a leitura por parte dos *hackers* durante a conexão.

4.5 Monitoração *Web*

Foi definido um cenário *Web* para a verificação da disponibilidade do HTTPS, cenário *Web* que é nada mais que uma ou mais requisições HTTP/HTTPS que são executadas pelo Zabbix *Server* em uma ordem pré-definida, em que este cenário é associado ao *host* Monitoramento de *links Web* do Servidor *Web*.

A Figura 22 mostra a configuração do cenário Web para que seja possível realizar o monitoramento via Localizador Padrão de Recursos - *Uniform Resource Locator*(URL).

Figura 22 - Cenário monitoração *web*



Fonte: Painel Zabbix

4.6 Integração com mídias

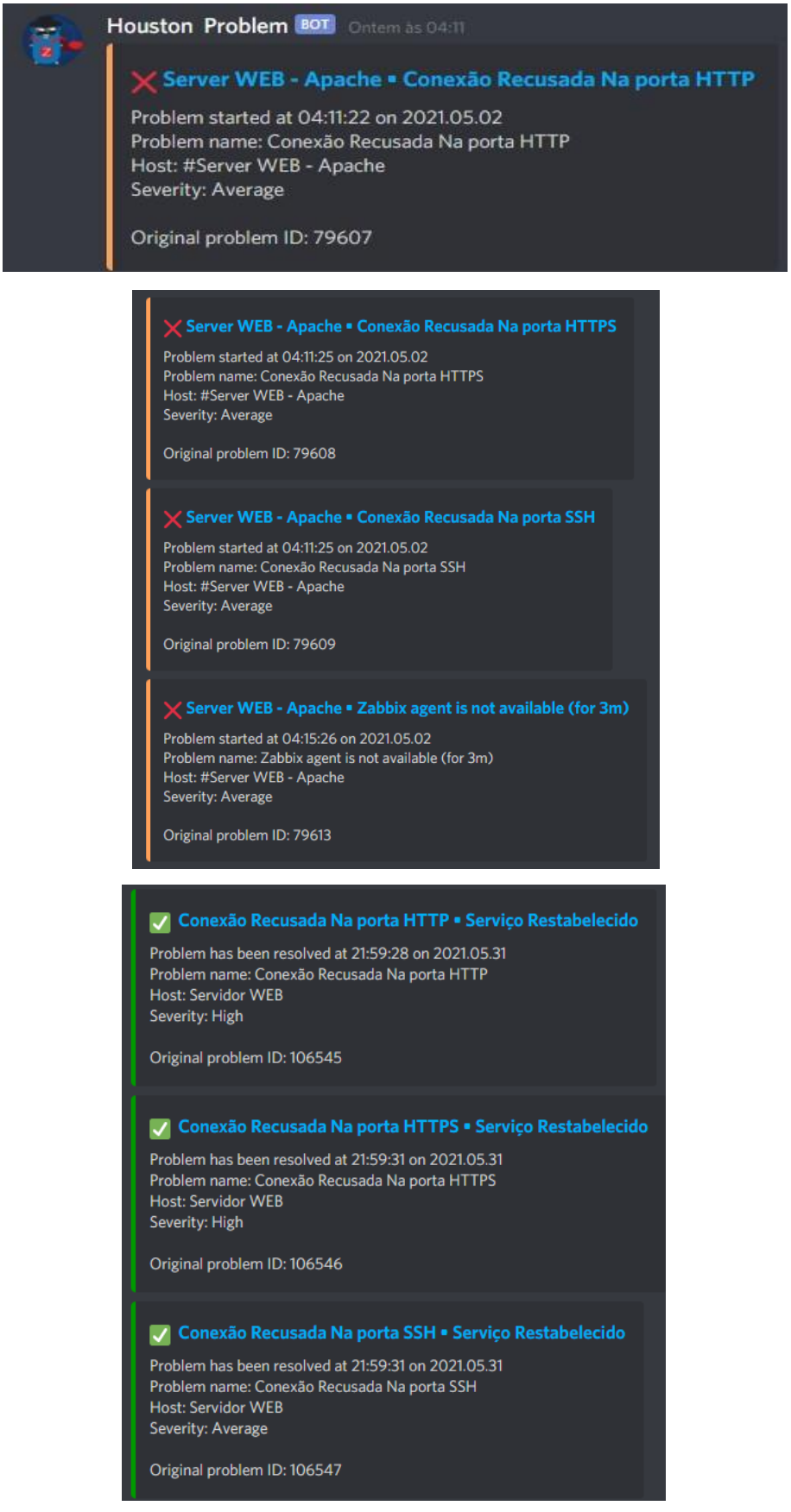
As notificações de *Triggers* que o Zabbix gera foram divididas em 3 plataformas de comunicação:

- Gmail;
- Discord;
- Telegram;

Das 6 notificações que são enviadas sobre o Servidor *Web* foram divididas em criticidade por plataforma de notificação. Pelo Discord e Telegram são enviadas as notificações de disponibilidade de portas e *status* agente Zabbix, enquanto via Gmail é enviada somente as notificações sobre monitoração *Web*.

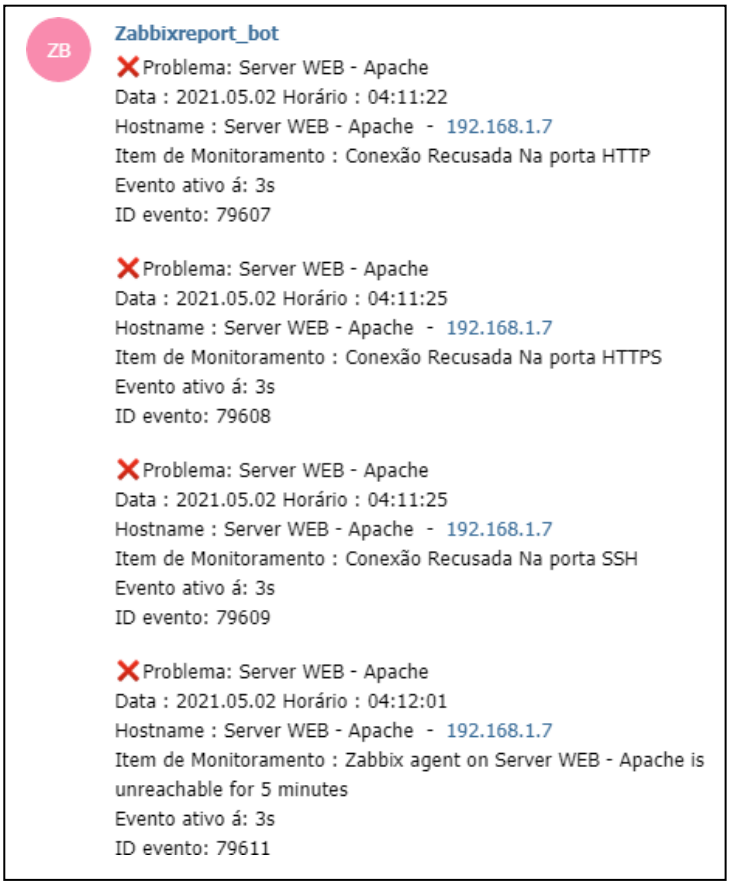
A Figura 23 e Figura 24 exibem as mensagens de alertas enviadas para Discord e Telegram.

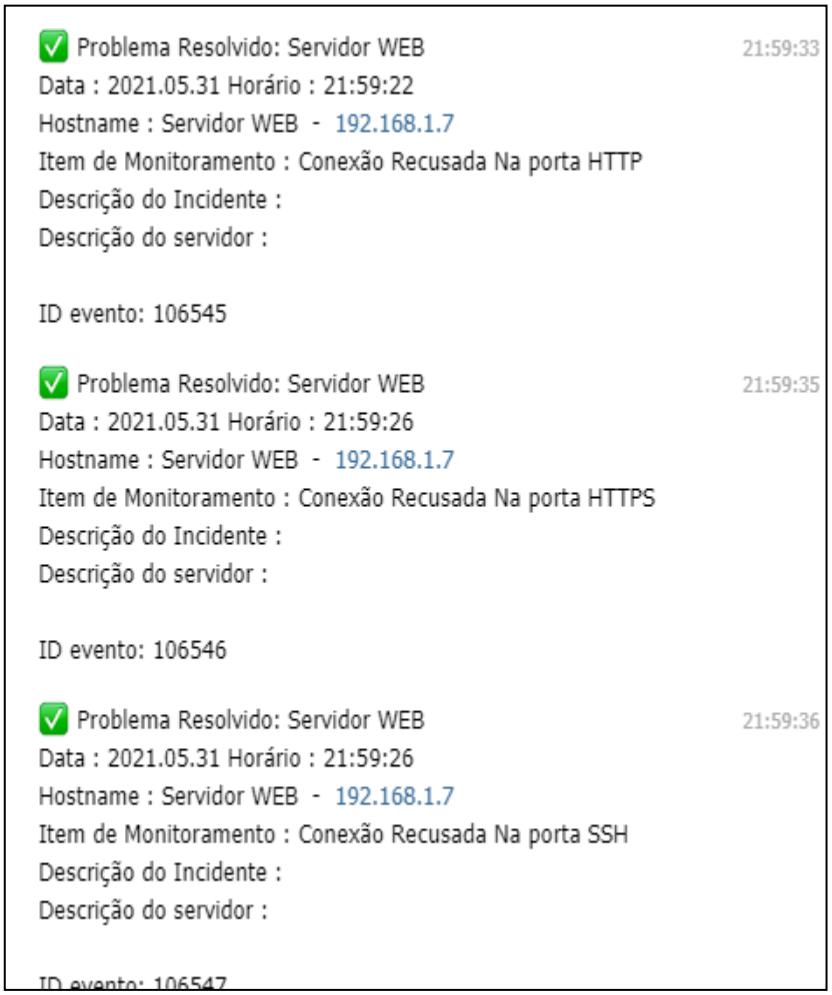
Figura 23 - Notificações enviadas ao Discord



Fonte: Discord, adaptado pelo autor

Figura 24 -Notificações enviadas ao Telegram





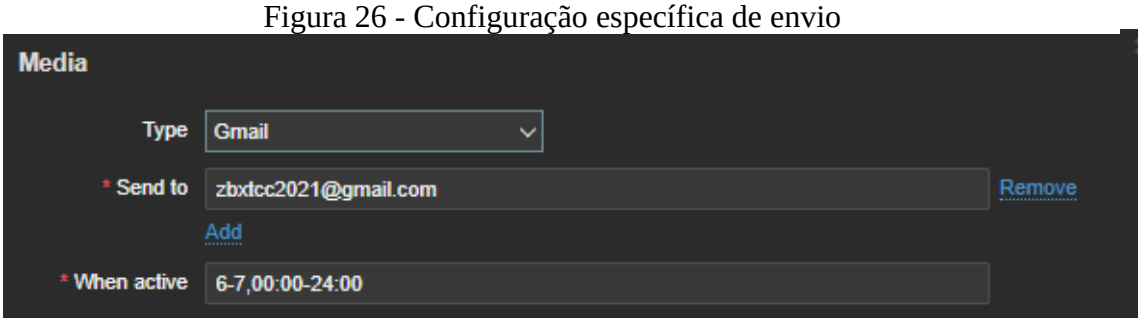
Fonte: Telegram, adaptado pelo autor.

As Figuras 25 e 26 exibem a notificação enviada ao Gmail, a qual e configura para ser acionada somente aos fins de semana como a Figura 26 demonstra na configuração.

Figura 25 - Notificação enviada ao Gmail

Fonte: Gmail, adaptado pelo autor.

Figura 26 - Configuração específica de envio



Fonte: Painel Zabbix

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento e a documentação para a o monitoramento do ambiente de um Servidor *Web*, apresentando a sua justificativa, descrição, ferramentas utilizadas para o sua implementação e seus resultados.

Para desenvolvimento do ambiente foi realizada a configuração de servidores em ambientes físicos e virtualizados. Estes serviram para configurar e determinar as características a serem utilizadas para implementação do monitoramento do ambiente de um servidor *Web*.

Com a utilização do *Software VirtualBox*, possibilitou a instalação do Debian um sistema operacional composto inteiramente de *software* livre e o Apache que é o servidor *Web* livre, a utilização do *Raspberry pi 3* que é uma série de computadores de placa única do tamanho reduzido que hospedou o servidor Zabbix e MariaDb um tipo de banco de dados, assim como o MySQL.

A escolha da ferramenta de monitoramento Zabbix foi feita por possuir uma grande possibilidade de configurações, coletas de informações, facilidade de manipulação dos objetos além de possuir um Interface de Programação de Aplicação - *Application Programming Interface*(API) que permite a integração com inúmeras outras ferramentas de exibição de painéis, notificações e análise de *logs*.

Foi observado na implantação do ambiente de monitoramento que é possível integrar várias outras ferramentas para a visualização das informações coletadas e os alertas emitidos dando uma visão clara e direta que possibilita que o administrador da rede e toda sua equipe monitorem o ambiente sem surpresas.

O objetivo deste trabalho foi propor uma solução de identificação de incidentes que fosse capaz de auxiliar na gerência de redes, o qual teve a abordagem nas 5 áreas do modelo ISO de gerenciamento de redes, onde com auxílio do Zabbix foi possível propor uma monitoração a partir do modelo ISO.

No gerenciamento de falhas, foi possível monitorar os recursos de *hardware* e *software* dos *hosts* monitorados. Para que que as falhas ocorressem foram realizados teste de *stress* de alguns recursos e a parada de alguns serviços. Com auxílio das *Triggers* e os alertas do Zabbix foram possíveis as notificações dos incidentes.

No gerenciamento de configuração, foi possível monitorar e determinar os *hardwares* dos *hosts* monitorados como: Memória, CPU e espaço em disco. Também foi possível monitorar serviços e aplicações como Apache e banco de dados.

No gerenciamento de contabilização, foi possível verificar a eficiência da rede, assistindo a troca de procedimentos e uso de recursos necessários para o servidor *Web*. E assim documentando as mudanças de configuração.

No gerenciamento desempenho foi realizado monitoramento da utilização de disco rígido, memória, CPU e serviços em um determinado período. Esse tipo de análise pode auxiliar na identificar gargalhos e/ou picos nas operações.

No gerenciamento de segurança foi possível realizar a verificação do certificado SSL do servidor *Web* a parti do uso de um *script*, que demonstrou que o Zabbix pode auxiliar na obtenção de informações integrado juntamente com outros meios ou ferramentas.

Para o envio de todas as notificações que o Zabbix gerou a partir das *Triggers*, foi realizada a integração com Gmail, Discord e Telegram, em que a mensagem de alerta é composta por: data e hora do incidente, descrição da *Trigger*, nome do *host* e severidade do problema.

Por fim, o objetivo desde trabalho foi cumprido ao detectar a ocorrência de disponibilidades em um servidor *Web*, possibilitando a notificação e solução em alguns casos. O monitoramento é algo importante para a área de TI e precisa ser tratado com cautela, o Zabbix se apresentou como uma ferramenta compacta a qual pode auxiliar empresas no acompanhamento de seus ambientes de trabalho, pode ser modelada a qualquer tipo de cenário, desde o mais simples até o mais complexo e dinâmico dos ambientes.

5.1 Sugestões de Trabalhos futuros

- Integração Zabbix com Grafana que é uma plataforma para visualizar e analisar métricas por meio de gráficos.
- Configurar descoberta automática de *host*;
- Criar um alerta específico, caso algum serviço seja parado por um usuário autorizado;

- Utilização do Zabbix Proxy;
- Identificar portas de acesso ao *host* automaticamente;
- Autenticação do HTTPS.

REFERÊNCIAS

EVEO. **Alta disponibilidade: como montar uma infraestrutura com o máximo de uptime**. 2019. Disponível em: <https://www.eveo.com.br/blog/alta-disponibilidade/.> Acesso em: 23 abr. 2021.

IETF.ORG. **Protocolo de transferência de hipertexto - HTTP**, 2020. Disponivel em: <https://tools.ietf.org/html/rfc2616>. Acesso em: 14 março 2021.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet:** Uma Abordagem Top-Down. 6. ed. São Paulo: Pearson Education, 2013.

MAURO, D.; SCHMIDT, K. **Essencial SNMP**. 2. ed. Sebastopol: O'Reilly, 2005.

NIC.BR. **Introdução ao Gerenciamento de Redes - parte 2**, 2014. Disponivel em: <https://www.nic.br/videos/ver/introducao-ao-gerenciamento-de-redes-parte-1/>. Acesso em: 20 fev. 2021.

PISSARRA, F. L. **Obtendo informações de certificados em sites com OpenSSL**, 2014. Disponível em: <https://bitismyth.wordpress.com/2014/04/24/obtendo-informacoes-de-certificados-em-sites-com-openssl/>. Acesso em: 18 abr. 2021.

PENSSL.ORG. **Bem-Vindo ao OpenSSL**, 2020. Disponivel em: <https://www.openssl.org/ >. Acesso em: 01 abr. 2021.

PRETEL, J. **A importância do monitoramento no ambiente de TI**, 2015. Disponível em:<https://www.profissionaisti.com.br/2015/09/a-importancia-do-monitoramento-no-ambiente- de-ti/>. Acesso em: 27 mar. 2021.

PRETEL, J. **Conheça o Zabbix:** *software* para monitoramento de ambientes de TI, 2015.Disponível em: <https://www.profissionaisti.com.br/2015/10/conheca-o-zabbix-*software*- para-monitoramento-de-ambientes-de-ti/>. Acesso em: 04 mar. 2021.

PYTHON.ORG. **SimpleHTTPServer:** Manipulador de solicitação HTTP simples, 2019. Disponível em: <https://docs.python.org/2/library/simplehttpserver.html>. Acesso em: 15 mar. 2021.

RFC 1156. **Processo SNMP,** 1990. Disponível em: <https://tools.ietf.org/html/rfc1157> Acesso em: 05 mar. 2021

RFC 1157. **A Simple Network Management Protocol (SNMP),** 1990. Disponível em: <https://tools.ietf.org/html/rfc1157> Acesso em: 05 mar. 2021

RFC 3411, **Arquitetura SNMP**. Disponível em: <https://www.ietf.org/rfc/rfc3411.txt>. Acesso em: 26 fev. 2021.

SILVA, R. S. S. **Simple Network Management Protocol (SNMP)**, 2020. Disponivel em: <https://www.gta.ufrj.br/grad/04_1/snmp/arquitetura.htm>. Acesso em: 03 fev 2021.

TECH EXPERT TIPS. **Zabbix Monitorando um *Web*site**, 2015. Disponível em: <https://techexpert.tips/pt-br/zabbix-pt-br/zabbix-monitorando-um-*Web*site/>. Acesso em: 23 abr. 2021.

TECNOLÓGICA. **Veja as consequências de você não ter um serviço de TI sempre disponível**, 2017. Disponível em: <https://blog.teclogica.com.br/veja-as-consequencias-de- voce-nao-ter-um-servico-de-ti-sempre-disponivel/>. Acesso em: 28 abr. 2021.

VALID CERTIFICADORA. **SSL ou TLS: quais são as diferenças entre esses protocolos?.**, 2021. Disponível em:< https://blog.validcertificadora.com.br/ssl-ou-tls-quais-sao-as-diferencas-entre-esses-protocolos/>. Acesso em: 26 abr. 2021.

ZABBIX SIA. **3 Funcionabilidades do Zabbix,** 2020. Disponível em: <https://www.zabbix.com/documentation/current/pt/manual/introduction/features/>. Acesso em: 15 mar. 2021.

ZABBIX SIA. **2 Zabbix Agent No Microsoft Windows,** 2021. Disponível em: < https://www.zabbix.com/documentation/current/pt/manual/appendix/install/windows_agent/>. Acesso em: 12 abr. 2021.

ZABBIX SIA. **5 *Triggers*,** 2020. Disponível em: < https://www.zabbix.com/documentation/current/pt/manual/appendix/*triggers*/>. Acesso em: 15 mar. 2021.

ZABBIX SIA. **8 Monitoração *Web*,** 2020. Disponível em: < https://www.zabbix.com/documentation/current/pt/manual/*Web*_monitoring>. Acesso em: 15 mar. 2021.

ZABBIX SIA. **5 Verificações simples,** 2020. Disponível em: < https://www.zabbix.com/documentation/current/pt/manual/config/items/itemtypes/simple_checks>. Acesso em: 20 mar. 2021.

WAZLAWICK, R. S. **Metodologia de Pesquisa para Ciência da Computação**. 2. ed. Riode Janeiro: Elsevier, 2014.

ANEXO A – INSTALAÇÃO DO ZABBIX 5.0.9

***Hardware* Utilizado:**

- Raspberry Pi
- Micro SD

Sistema Operacional Utilizado: Raspberry Pi OS Lite

Pré-requisitos:

- Servidor
  - *Web* Apache/PHP
- Banco de Armazenamento
  - MySQL/MariaDB
- PHP com extensões necessárias

Etapa 1 – Realizar acesso via terminal o seu Rasp, usando como usuário *pi* e senha *raspberry* como mostra a Figura A. 1.

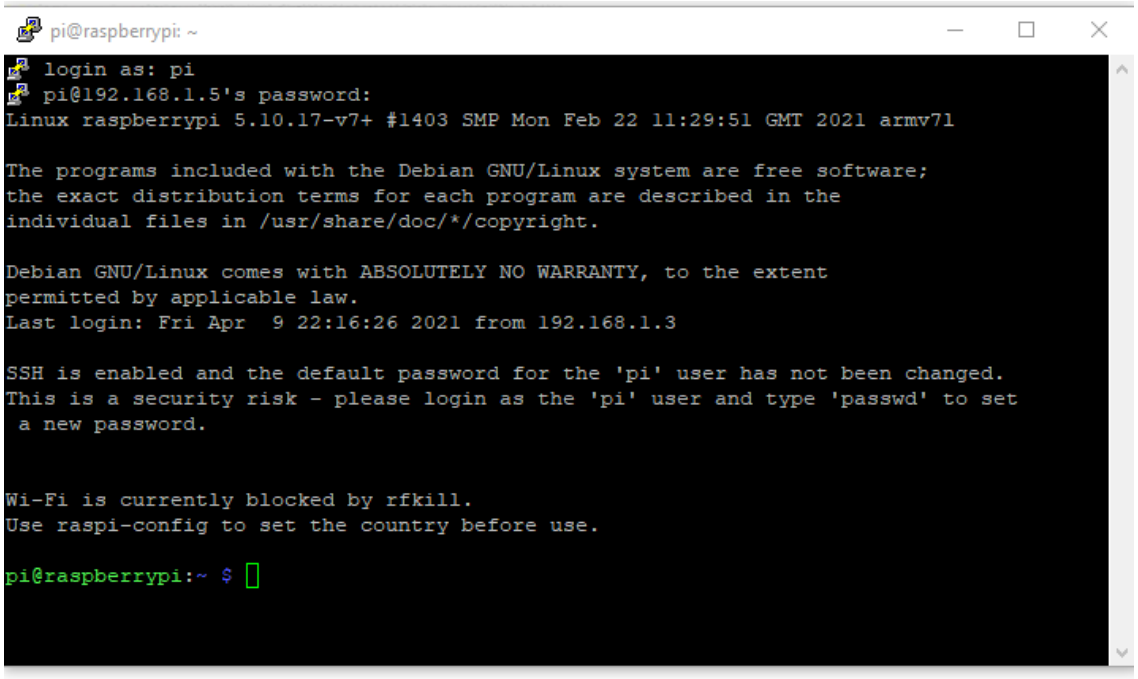


Figura A. 1 Primeiro Login no SO

Etapa 2 – Realizar atualização dos pacotes, após atualização instalar editor de texto Nano

Atualização:

- sudo apt update
- sudo apt upgrade

Instalação editor de texto:

- sudo apt install nano

Etapa 3 – Realizar Instalação Banco de Dados MariaDB

Vamos instalar e preparar o Banco de Dados MariaDB com os seguintes comandos:

- sudo apt -y install mariadb-server
- systemctl enable mariadb
- sudo mysql_secure_installation

Utilizamos o comando *mysql_secure_installation* que é um script shell disponível nos sistemas Unix e permite que você melhore a segurança do banco de dados, como mostra a figura A. 2.

Etapa 4 – Instalação Zabbix e seus pacotes necessários

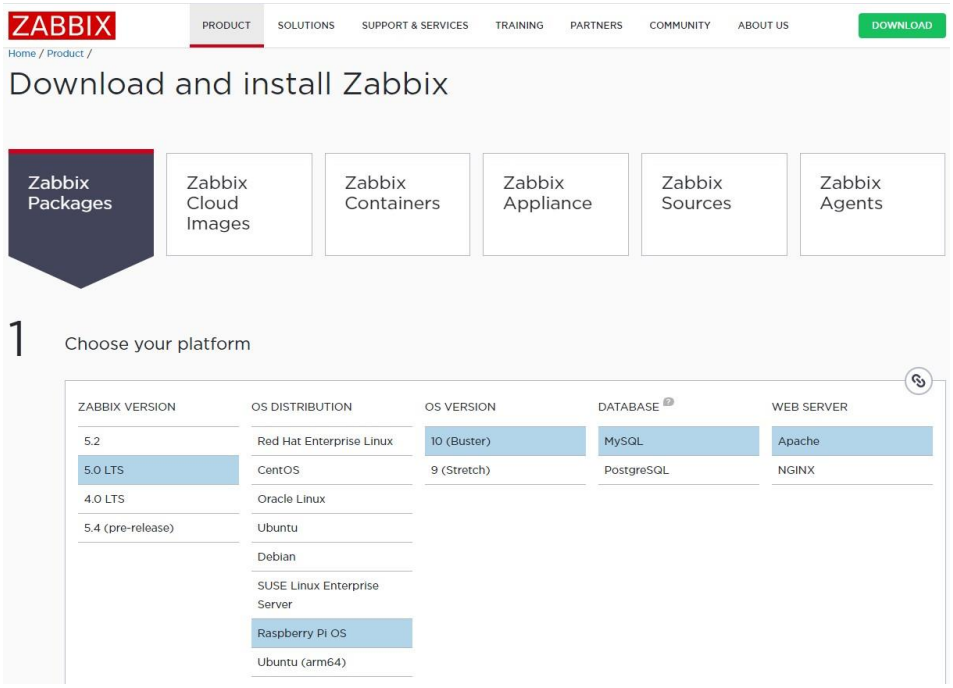


Figura A. 2 - Página para *download* Zabbix

Acessando a página oficial www.zabbix.com apresentado na Figura A. 2, a instalação realizada é oferecida pela própria documentação do Zabbix por ser mais simples e rápida.

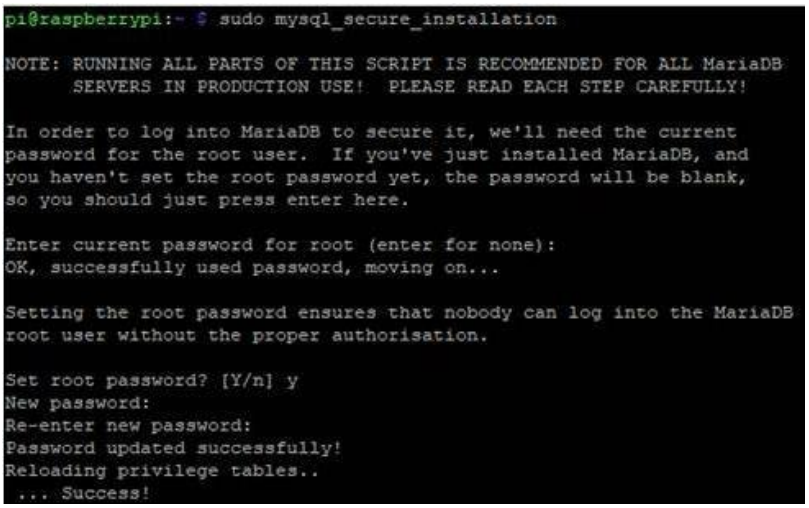


Figura A. 3 - Execução comando mysql_secure

a) Instalação e configuração do servidor Zabbix para sua plataforma

wget https://repo.zabbix.com/zabbix/5.0/raspbian/pool/main/z/zabbix-release/zabbix-release_5.0-1+buster_all.deb

dpkg -i zabbix-release_5.0-1+buster_all.deb

apt update

b) Instalação servidor Zabbix, Frontend, Agent

apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-agent

Etapa 5 – Criando banco de dados para o Zabbix

Certifique-se de que você tem o servidor de banco de dados em funcionamento.

systemctl status mariadb.service

Agora precisamos criar o banco de dados para o Zabbix. Use os seguintes comandos:

sudo mysql -uroot -p' zabbixrasp' -e "create database zabbix character set utf8 collate utf8_bin;"

Dando as permissões:

sudo mysql -uroot p'senha informada no passo anterior' -e "grant all privileges on zabbix.* to zabbix@localhost identified by ‘zabbixrasp’;"

Importando esquema inicial de dados:

zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p' zabbixrasp' zabbix

Etapa 6 – Modificar o arquivo de configuração do Zabbix com detalhes do banco de dados.

nano /etc/zabbix/zabbix_server.conf

Modificar parâmetros no arquivo

DBHost = localhost (linha 92)

DBName = zabbix (linha 101)

DBUser = zabbix (linha 117)

DBPassword = zabbixrasp (linha 125)

Etapa 7 – Alterar o fuso horário editando o arquivo do PHP.

nano /etc/zabbix/apache.conf

php_value date.timezone America/São_Paulo(linha 20)

php_value date.timezone America/São_Paulo(linha 30)

Alterar as linhas 20 e 30 trocando o fuso horário para "America/São_Paulo" conforme o comando abaixo, a Figura A.4 mostra a alteração realizada.

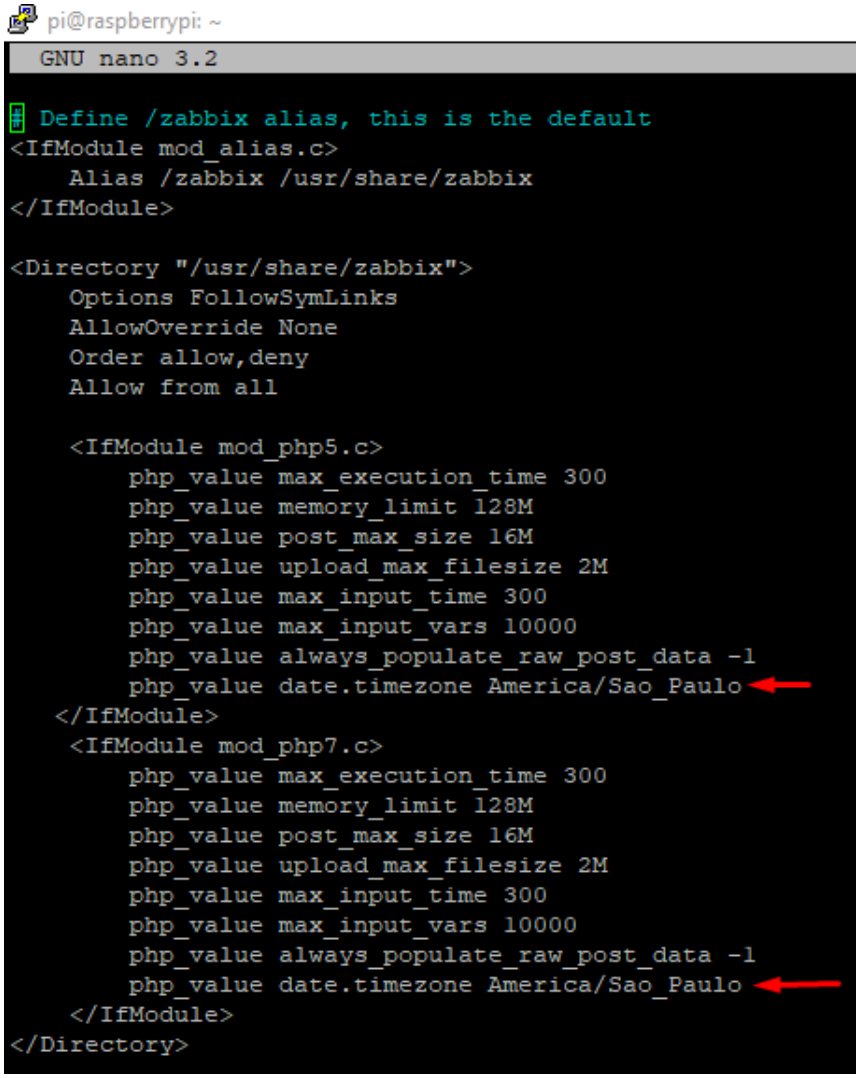


Figura A. 4 - Alterando TimeZone

Etapa 8 – Subir serviços do Zabbix, agente, httpd e php

systemctl restart zabbix-server zabbix-agent apache2

systemctl enable zabbix-server zabbix-agent apache2

Etapa 9 – Configurar Frontend Zabbix

Conecte-se ao seu frontend Zabbixrecém-instalado:

http://server_ip/zabbix ou http://Host_Name_Server/zabbix

Será apresentada a página de boas-vindas, conforme a Figura A.5.

Figura A. 5 - Tela de boas-vindas Zabbix

Clicar em *Next step.*

Verificar se não ouve alguma falha nos pré-requisitos que são exibidos na tela de *check of pre-requisites*, a Figura A.6 mostra a tela de checagem, caso não seja encontrado prosseguir para próxima etapa.

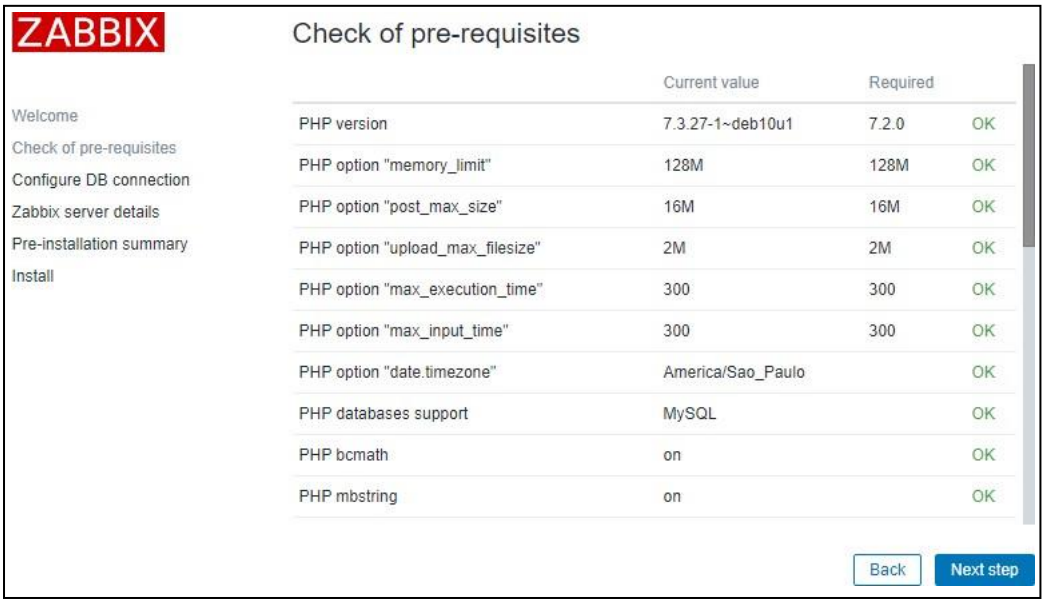


Figura A. 6 - Pré-requisitos instalação

Preencher de acordo com que foi definido no banco de dados, como mostrado na Figura A.7. Após, clicar em *Next step*.

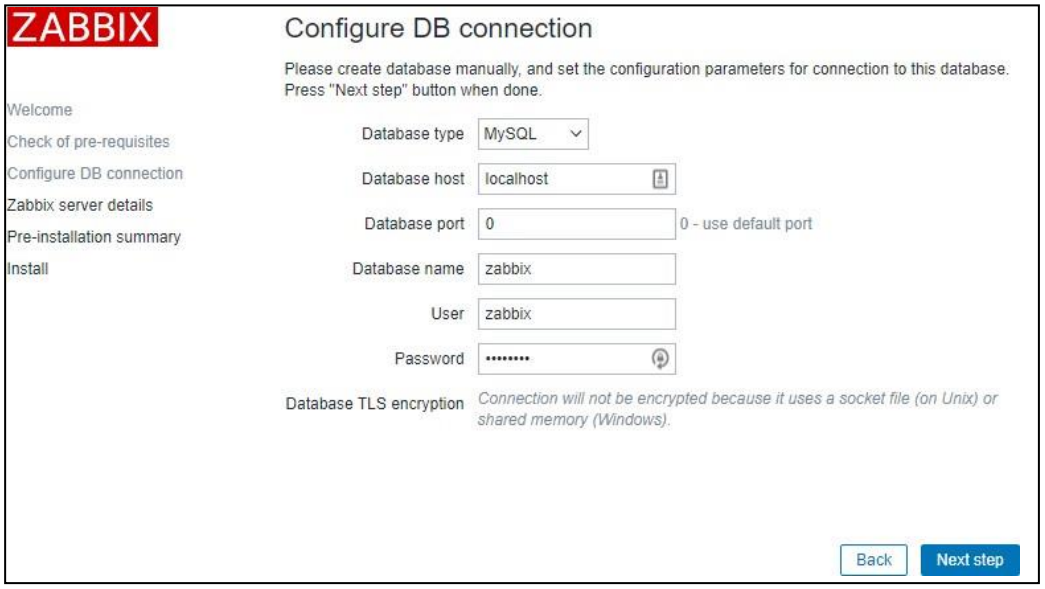


Figura A. 7 - Configuração banco no Zabbix

Verificar se todos os campos foram preenchidos, caso não encontre divergências nos dados apresentados como mostra a

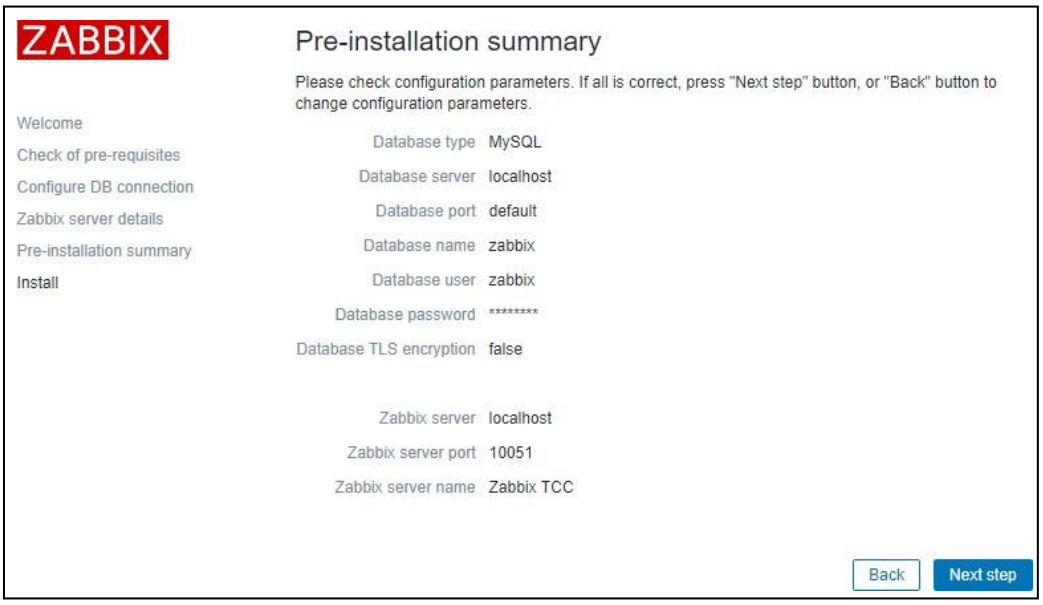


Figura A. 8 - Confirmação de Informações Zabbix

Após todo processo a ferramenta Zabbix já está pronta para seu uso como é apresentado na Figuras A. 9.

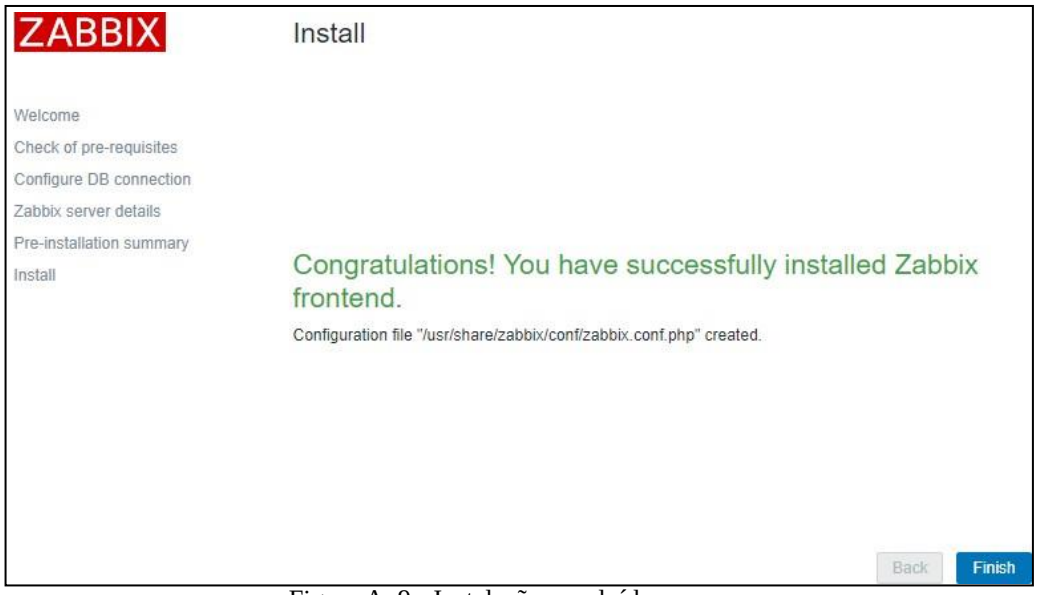


Figura A. 9 - Instalação concluída com sucesso

Por fim, será inicializada a tela de login do Zabbix como mostra a Figura A.10.

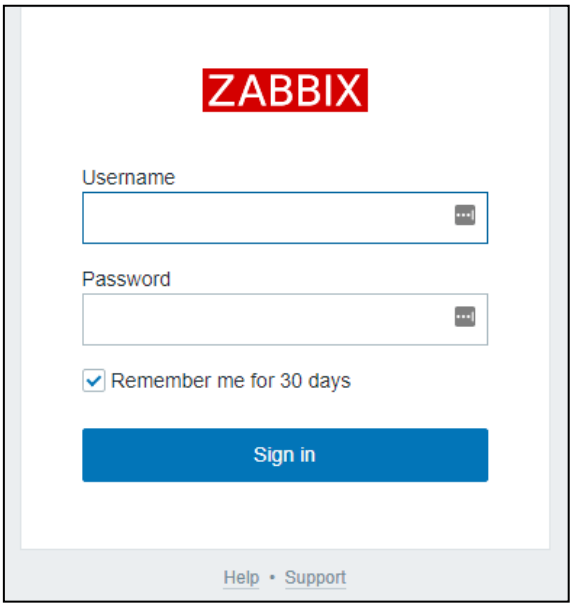


Figura A. 10 -Tela de *login* Zabbix

ANEXO B – CONFIGURAÇÃO MÍDIA DO TELEGRAM

Criando *bot* no Telegram

Acessando o Telegram foi feita a busca pelo seguinte usuário *@BotFather*. Em seguida foi iniciada a conversa.

Para iniciar a conversar foi digitado o comando /start onde a resposta foi a lista de comandos, como na figura B. 1:

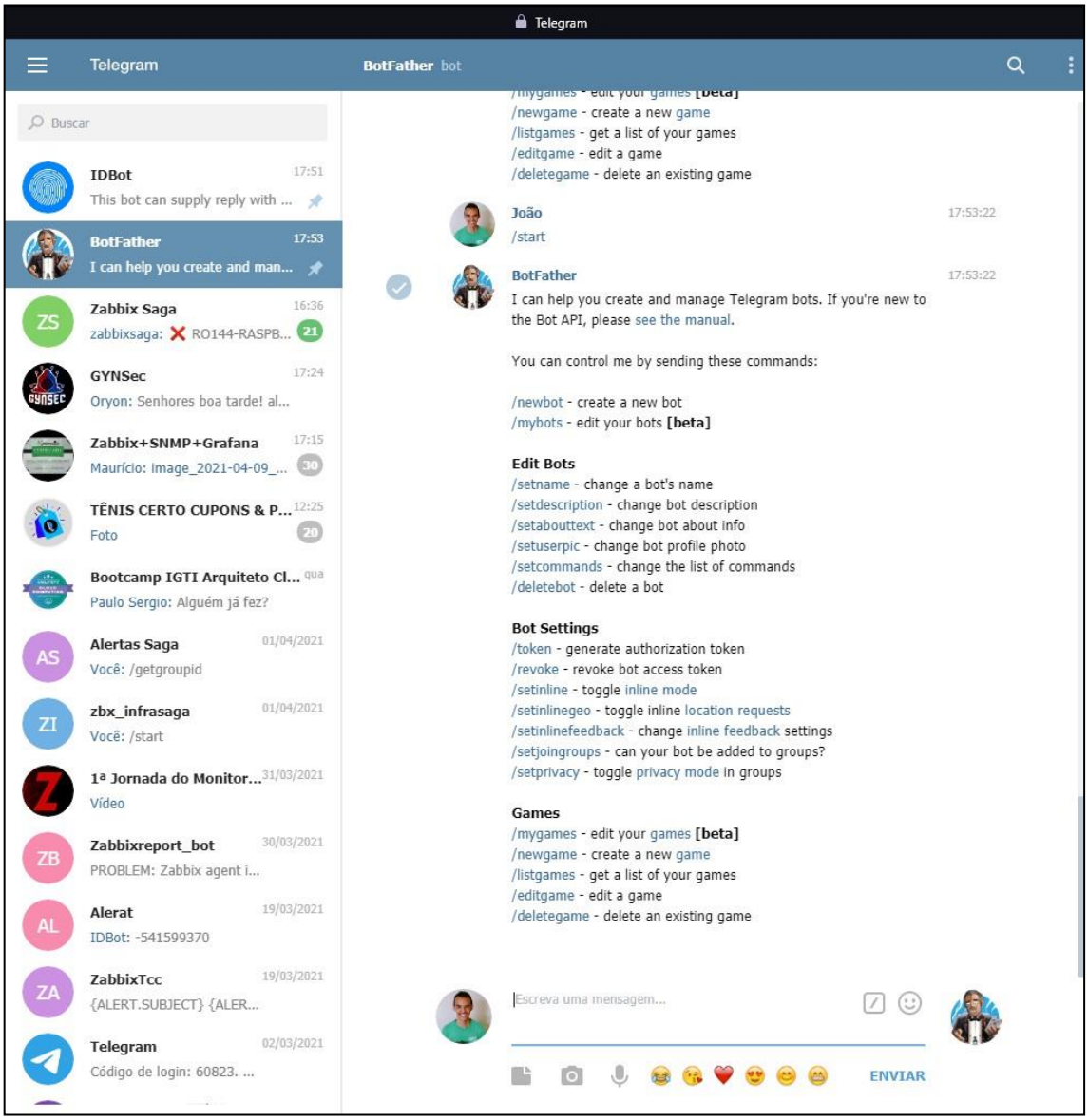


Figura B. 1 - Criando bot

O comando */newbot* foi usado para iniciar a criação de um novo *bot*.

Após o comando ser executado será informado o nome para o *bot*. Ex.: “Zabbix Report”.

Em seguida é necessário inserir o nome de usuário para o *bot*, sendo obrigatório terminar com *bot*. Ex.: “zbx_labtcc_bot”. A figura B. 2 mostra a conclusão da criação do *bot.*

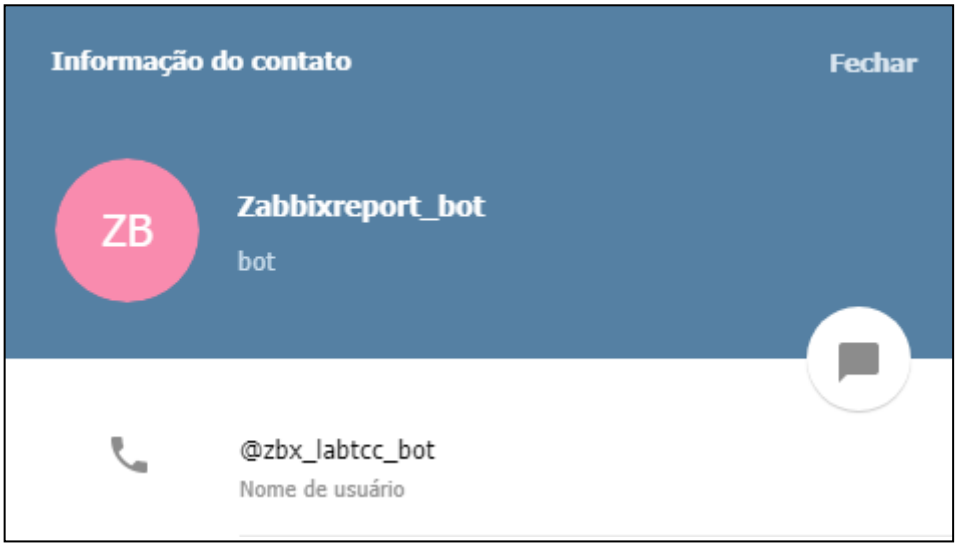


Figura B. 2 - Usuário criado no Telegram

Criando tipo de mídia no Zabbix para Telegram

A figura B. 3 demonstra como foi criado o tipo de mídia que será responsável por efetuar o envio dos alertas gerados pelo Zabbix.

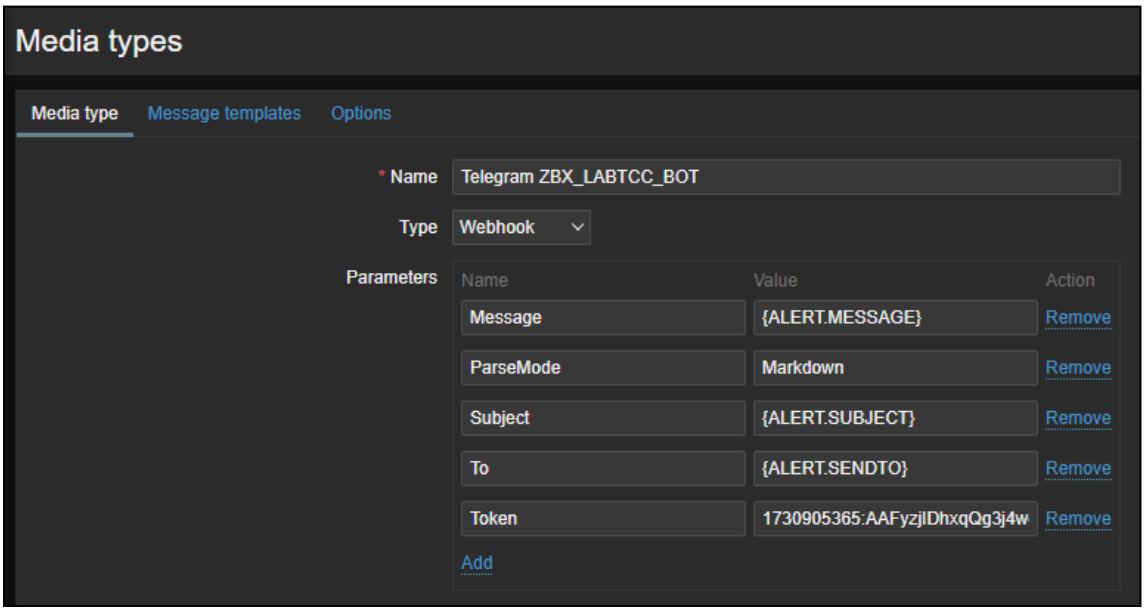


Figura B. 3 - Criando Tipo de Mídia

Após a criação do tipo de mídia é necessário realizar a configuração das ações no Zabbix, as quais são configuradas a partir da solicitação do usuário, apresentado na figura B. 4.

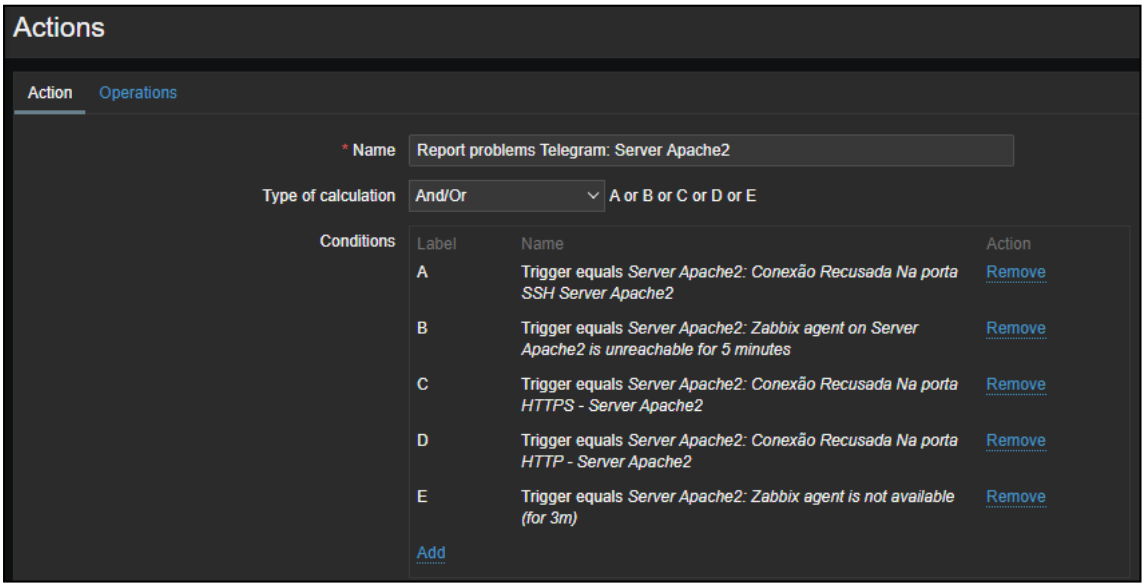


Figura B. 4 – Ações de ativação *Trigger*

Para que as notificações do Telegram sejam enviadas e necessário adicionar o tipo de mídia aos usuários ou grupos específicos, como a figura B. 5 demonstra.

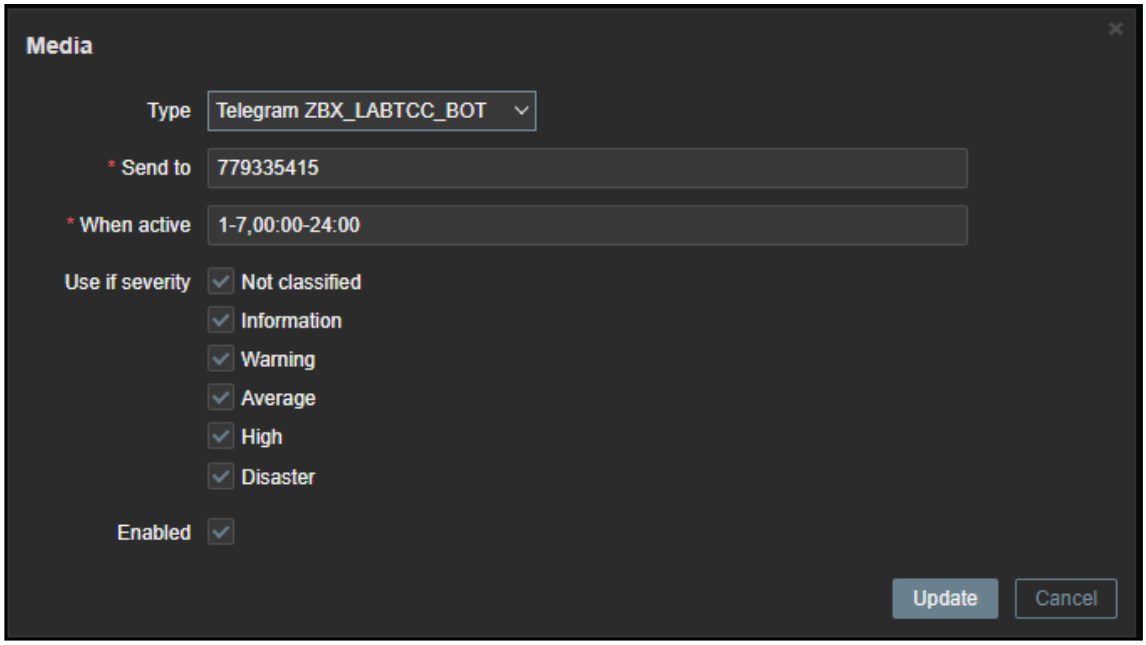


Figura B. 5 - Configurando Usuário que irá receber alertas

ANEXO C – CONFIGURAÇÃO MÍDIA DO DISCORD

Criando *Webhook* Discord

Acessando o aplicativo Discord via *Web* ou o aplicativo Discord *desktop*, selecione o servidor e o canal onde deseja obter as notificações do Zabbix. Pressione **o canal editar,**selecione a guia ***Webhooks*** e pressione criar botão ***Webhook*, como a figura C. 1 demonstra.**

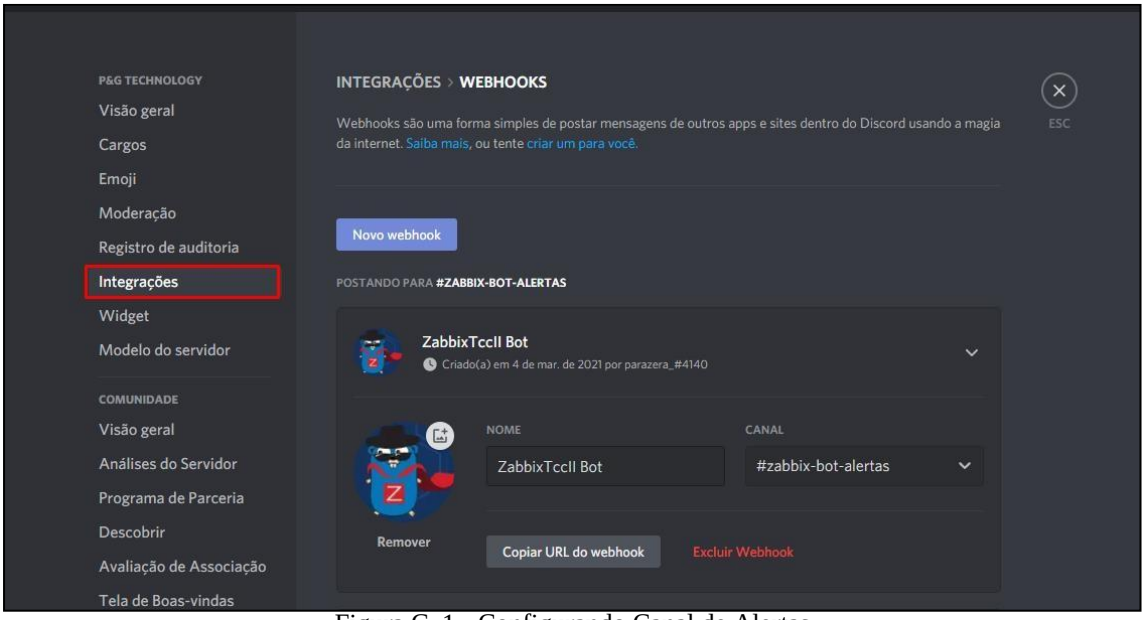


Figura C. 1 - Configurando Canal de Alertas

Após configurar o nome e canal onde deseja informar os alertas, copie o link do *Webhook.*

Criando tipo de mídia no Zabbix para Discord

A parti da do Zabbix 5.0 o tipo de mídia da plataforma Discord já vem pré-instalada, ao abrir o tipo de mídia será possível verificar os parâmetros já configurados, pode-se escolher entre dois modos de notificação modificando o valor do parâmetro "*use_default_mensagem":*

- **falso** (padrão)
  - receber notificações de problemas com conjunto de campos predefinidos (problema, nome do host, gravidade do evento, tipo de eventos etc.)
- **verdadeiro**

- receber mensagens padrão definida na Ação Zabbix que acionou a notificação

A figura C. 2 mostra os parâmetros citados acima.

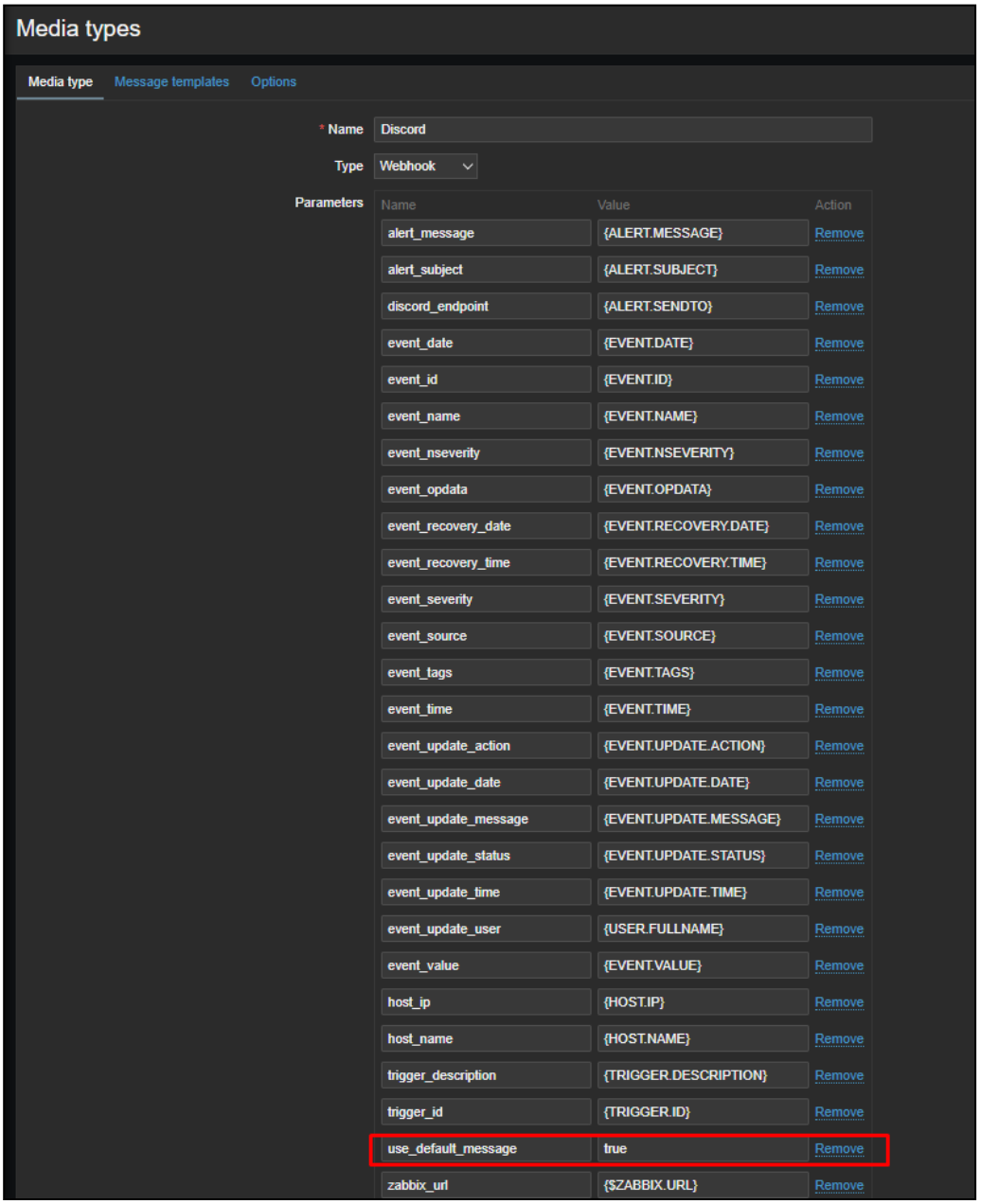


Figura C. 2 - Parâmetros mídia Discord

Para receber notificações no Discord, você precisa adicionar **mídia** com o tipo de mídia **Discord ao usuário ou grupo que recebera os alertas.** O campo "*Send to*" deve conter a URL do *Webhook* do Discord criada antes, como a figura C. 3 demonstra.

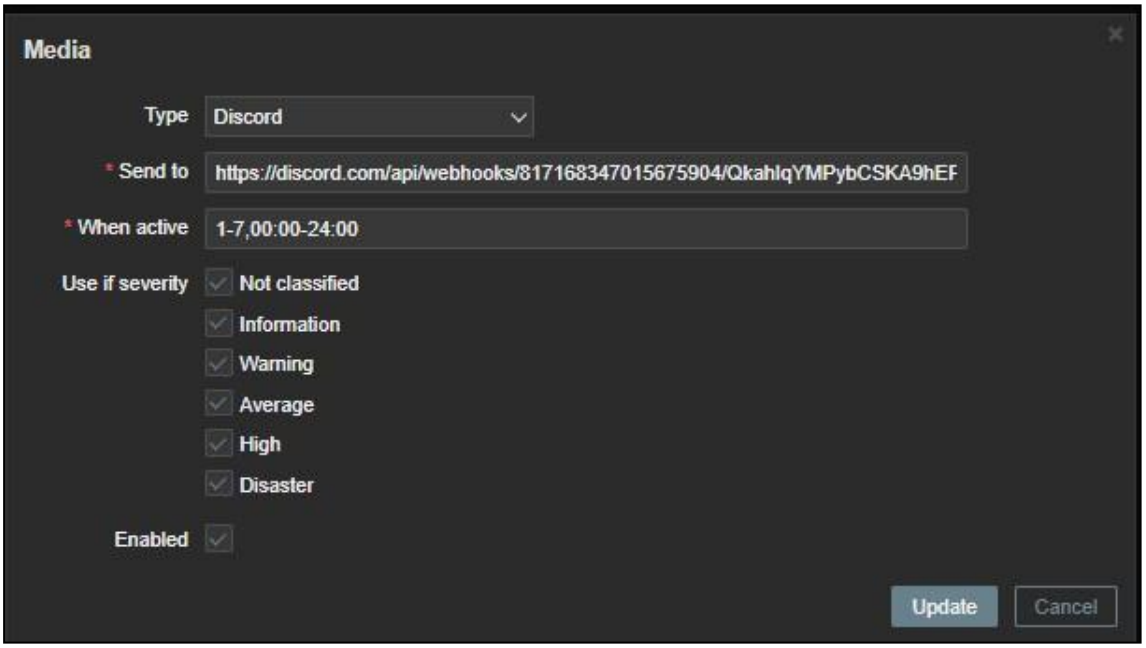


Figura C. 3 - Configurando Usuário que irá Receber Alertas

ANEXO D - CONFIGURAÇÃO MÍDIA DO GMAIL

Para criação do tipo de mídia do Gmail é necessário verificar as informações de *SMTP* do Gmail:

- **Endereço do servidor SMTP do Gmail**: smtp.gmail.com
- **Nome Gmail SMTP**: Seu nome completo
- **Gmail SMTP username:** Seu endereço completo do Gmail (ex.: you@gmail.com)
- **Senha Gmail SMTP:** A senha que você usa para fazer login no Gmail
- **Porta Gmail SMTP (TLS):** 587
- **Porta Gmail SMTP (SSL):** 465

Para que as notificações via Gmail sejam enviadas preencher os campos, como a figura D. 1 demonstra

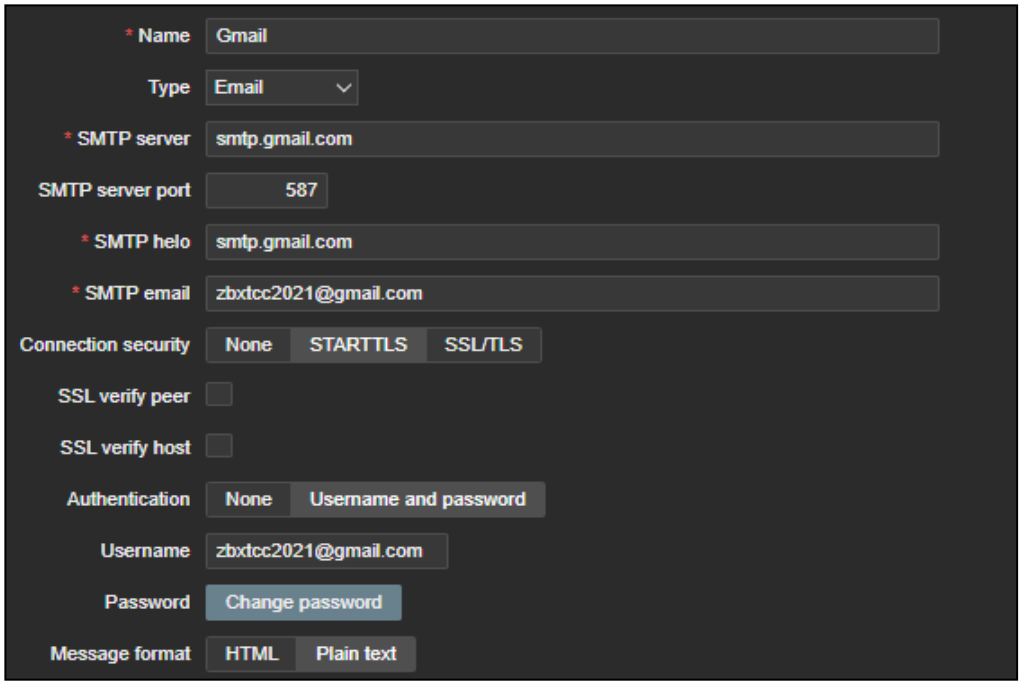


Figura D. 1 - Configuração Tipo de mídia Gmail

Posteriormente devemos adicionar o tipo de mídia ao usuário que deverá receber as notificações cadastradas no tipo de mídia, como a Figura D. 2 mostra:

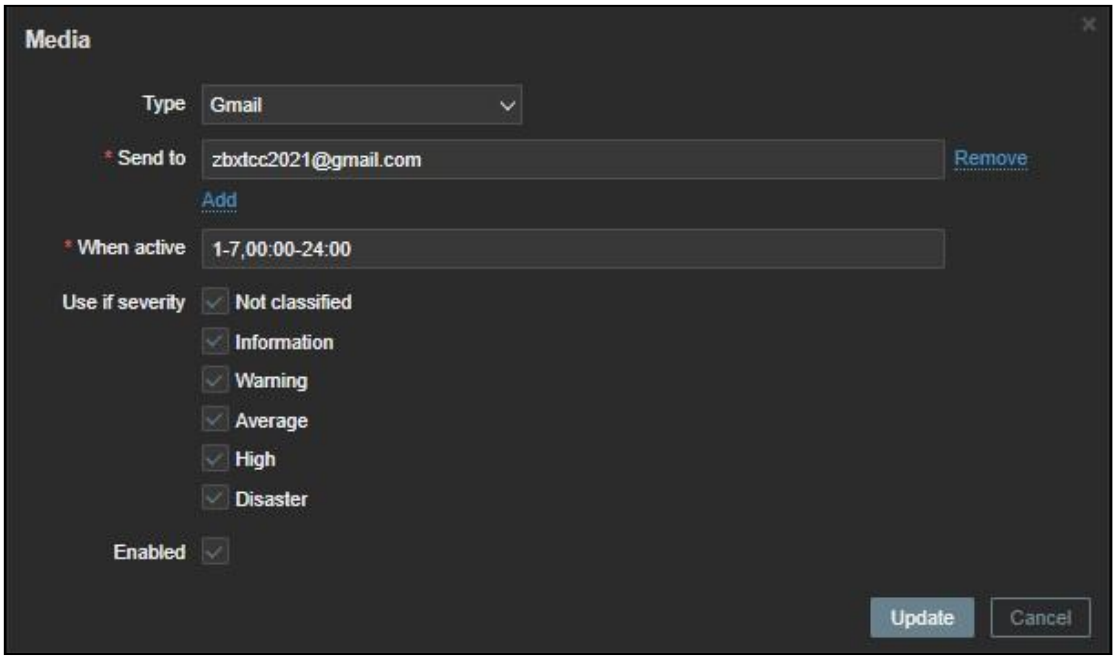


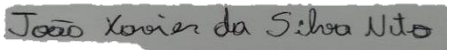
Figura D. 2 - Configuração do usuário que recebera notificação

ANEXO E – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA

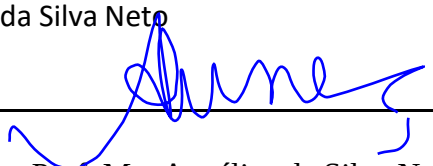
O(A) estudante João Xavier da Silva Neto do Curso de Engenharia de Computação, matrícula 2015.1.0033.0339-0, telefone: 62 9 82772870, e-mail: joaoxneto7050@gmail.com, na qualidade de titular dos direitos autorais, em consonância com a Lei n° 9.610/98 (Lei dos Direitos do autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado GERENCIAMENTO E MONITORAÇÃO DE REDES DE COMPUTADORES COM ÊNFASE EM DISPONIBILIDADE DE SERVIDOR WEB COM FERRAMENTA ZABBIX, gratuitamente, sem

ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto (PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da área; para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 12 de junho de 2021.

Assinatura do(s) autor(es): 

Nome completo do autor: João Xavier da Silva Neto

Assinatura do professor-orientador: 

Nome completo do professor-orientador: Prof. Ma. Angélica da Silva Nunes