Relatório de Projeto

Luís António Pinto de Barros

Engenharia Informática

ago | 2023

GUARDA
POLI
TÉCNICO

POLITÉCNICO GUARDA

Escola Superior de Tecnologia e Gestão

PROJETO DE INFORMÁTICA

SISTEMA DE MONITORIZAÇÃO DE REDES INTELIGENTE

RELATÓRIO DE ESTÁGIO
PARA OBTENÇÃO DO GRAU DE LICENCIADO(A) EM ENGENHARIA INFORMÁTICA

Luís António Pinto de Barros

Agosto / 2023

POLITÉCNICO GUARDA

Escola Superior de Tecnologia e Gestão

SISTEMA DE MONITORIZAÇÃO DE REDES INTELIGENTE

RELATÓRIO DE ESTÁGIO
PARA OBTENÇÃO DO GRAU DE LICENCIADO(A) EM ENGENHARIA INFORMÁTICA

Professor(a) Orientador(a): António Mário Ribeiro Martins

Luís António Pinto de Barros

Agosto / 2023

Agradecimentos

Primeiramente, gostaria de expressar a minha imensa gratidão à minha mãe e aos meus avós. Sem o amor deles, apoio e encorajamento contínuo, eu não estaria onde estou hoje. Foram a minha fonte de inspiração e força para enfrentar todos os desafios ao longo deste percurso académico.

Gostaria também de agradecer profundamente ao Prof. Dr. António Mário Ribeiro Martins, meu orientador, por seu inestimável apoio e orientação durante o meu estágio.

À equipa da Securnet, expresso a minha mais profunda gratidão. A experiência de trabalhar com vocês foi incrivelmente valiosa. Agradeço especialmente ao Eng. Pedro Boavida pela paciência, disponibilidade e pela oportunidade de aprender e crescer em cibersegurança.

Quero expressar minha gratidão aos meus colegas de estágio por proporcionarem um ambiente de trabalho amigável e acolhedor.

Agradeço igualmente aos meus professores e colegas pela motivação, ajuda e inspiração constantes. Criamos juntos um ambiente académico propício à aprendizagem e ao desenvolvimento, e por isso sou muito grato.

Um agradecimento especial aos meus amigos que partilharam este percurso comigo em cada etapa. O seu apoio foi fundamental para superar os desafios encontrados pelo caminho.

Adicionalmente, gostaria de agradecer a Carla Reis. A sua amizade, apoio e encorajamento desempenharam um papel fundamental nesta jornada. A sua paciência, compreensão e incentivo constante foram inestimáveis e estou profundamente grato.

Ficha de identificação

Aluno

Nome: Luís António Pinto de Barros

Número: 1700331

Licenciatura: Engenharia Informática

Estabelecimento de Ensino

Instituto Politécnico da Guarda (IPG)

Escola Superior de Tecnologia e Gestão (ESTG)

Entidade Acolhedora do Estágio

Nome: Securnet

Morada: Rua Monte da Bela Nº181 W 4445-294 Ermesinde Porto

Rua Carlos Alves Nº1, 2º Direito 1600-546 Lisboa

Contacto Telefónico: 213622204 | 224673094

Duração do Estágio: 2 meses

Supervisor de Estágio

Nome: Eng. Pedro Gabriel Bastardo de Miranda Boavida

Função: Diretor Técnico (Lisboa)

Docente Orientador de Estágio

Nome: António Mário Ribeiro Martins

Grau Académico: Doutor

Resumo

Durante o meu estágio na empresa Securnet, desenvolveu-se um projeto que envolveu a criação de um laboratório de rede com uma zona desmilitarizada (DMZ), onde se alojou servidores Secure File Transfer Protocol (SFTP), de *e-mail* e *web*. O objetivo principal foi estabelecer um ambiente controlado para a monitorização e otimização dos recursos da rede.

Inicialmente, configurou-se o laboratório de rede, criando uma rede com DMZ e procedendo à instalação e configuração dos servidores necessários.

Em seguida, foi implementado o sistema de monitorização Zabbix, que possibilitou a supervisão constante dos recursos da rede, incluindo a deteção de eventos e problemas. Além disso, configuraram-se os *hosts* para garantir a monitorização eficaz de todas as máquinas do laboratório.

Realizou-se a integração entre o Zabbix e o sistema de gestão de ativos GLPi, proporcionando uma visão mais abrangente e organizada das informações relacionadas com os dispositivos monitorizados.

Estabeleceu-se também a integração entre o Zabbix e o Grafana, o que permitiu a criação de painéis de monitorização personalizados e a análise visual dos dados recolhidos. Esta integração melhorou significativamente a apresentação dos resultados da monitorização.

Para facilitar o acesso às ferramentas de monitorização, foi desenvolvida uma aplicação móvel e web em React Native, centralizando assim a sua utilização e simplificando o acesso remoto às informações da rede.

Por fim, implementaram-se técnicas de machine learning nos dados recolhidos pelo Zabbix, com o propósito de reduzir a ocorrência de eventos irrelevantes**.**

Palavras-chave:

Engenharia de rede; Zabbix; GLPi; Grafana; React Native; *Machine Learning*

Abstract

During my internship at Securnet, I developed a project that involved creating a network laboratory with a DMZ, where I hosted SFTP, email, and web servers. The primary objective was to establish a controlled environment for monitoring and optimizing network resources.

Initially, I configured the network laboratory by creating a network with a DMZ and proceeding with the installation and setup of the necessary servers.

Next, I implemented the Zabbix monitoring system, which allowed continuous supervision of network resources, including event and issue detection. Additionally, I configured the hosts to ensure effective monitoring of all machines in the laboratory.

I integrated Zabbix with the GLPi asset management system, providing a more comprehensive and organized view of information related to monitored devices.

I also established integration between Zabbix and Grafana, enabling the creation of customized monitoring dashboards and visual analysis of collected data. This integration significantly improved the presentation of monitoring results.

To simplify access to monitoring tools, I developed a mobile and web application using React Native, centralizing their use and simplifying remote access to network information.

Finally, I implemented machine learning techniques on the data collected by Zabbix with the aim of reducing the occurrence of irrelevant events.

Keywords:

Network engineering; Zabbix; GLPi; Grafana; React Native; *Machine Learning*

Índice

Índice de Figuras

Lista de siglas

B

Bash ..... *Bourne Again Shell*

BD ..... Base de Dados

C

CPU ..... *Central Processing Unit*

CSIRT ..... *Computer Security Incident Response Team*

CSV ..... *Comma-Separated Values File*

D

DMZ ..... *Demilitarized Zone*

E

ER ..... Entidade Relacionamento

F

FTP ..... *File Transfer Protocol*

G

GLPi ..... *Gestionnaire Libre de Parc Informatique*

I

ICMP ..... *Internet Control Message Protocol*

IMAP ..... *Internet Messaging Access Protocol*

IP ..... *Internet Protocol*

IPMI ..... *Intelligent Platform Management Interface*

J

JMX ..... *Java Management Extensions*

L

LDAP ..... *Lightweight Directory Access Protocol*

M

MIB ..... Management Information Base

N

NAT ..... *Network Address Translation*

NVPS ..... *New values per second*

O

OID ..... *Object Identifier*

OpenSSH ..... *Open Secure Shell*

OSI ..... *Open Systems Interconnection*

P

PDF ..... *Portable Document Format*

PHP ..... *Hypertext Preprocessor*

POP3 ..... *Post Office Protocol*

R

RDBMS ..... *Relational Database Management System*

S

SFTP ..... *Secure File Transfer Protocol*

SLA ..... *Service Level Agreement*

SNMP ..... *Simple Network Management Protocol*

SSH ..... *Secure Shell*

T

TELNET ..... *Teletype Network*

TI ..... Tecnologias de Informação

U

URL ..... *Uniform Resource Locator*

V

VLAN ..... *Virtual Local Area Network*

VM ..... *Virtual Machine*

VPN..... *Virtual Private Network*

W

WAN..... *Wide Area Network*

Capítulo 1

1. Introdução

A era digital trouxe consigo inovações revolucionárias que transformaram praticamente todos os aspetos da vida moderna, inclusive a maneira como as empresas operam. No entanto, estas transformações não vieram sem os seus desafios. Na realidade atual, as organizações enfrentam um labirinto de complexidade na gestão e monitorização das suas redes. A necessidade de manter a segurança acrescenta uma camada adicional de complexidade a este cenário já intrincado. Esta dissertação de licenciatura em Engenharia Informática é resultado de um estágio numa proeminente empresa de consultoria de cibersegurança e aborda diretamente estes desafios.

No centro das operações da empresa, emergiu a necessidade de aperfeiçoar o processo de monitorização da infraestrutura de Tecnologias de Informação (TI) dos seus clientes. O panorama tecnológico atual, que é marcado por uma complexidade crescente e por um ritmo de evolução acelerado, exige soluções ágeis e robustas. Estas soluções devem ser capazes de garantir a segurança e eficiência, mantendo-se adaptáveis às mudanças constantes.

O problema central que se apresentou durante o meu estágio foi, portanto, como desenvolver e implementar uma solução eficaz para a monitorização e gestão de redes que não só enfrentasse os desafios da atualidade, mas que também fosse suficientemente flexível para se adaptar ao futuro. Este problema representa um desafio significativo dada a importância vital da cibersegurança e da gestão de redes do mundo empresarial moderno. Uma falha na monitorização ou uma violação de segurança pode resultar em interrupções dispendiosas, perda de dados ou até mesmo danos à reputação da empresa. Portanto, é de suma importância desenvolver soluções robustas e eficazes para estes problemas.

Durante o estágio, enfrentou-se este problema ao implementar uma variedade de estratégias e ferramentas. Inicialmente, foi desenvolvida e implementada uma rede

segura, incluindo servidores de email, web e SFTP dentro de uma *Demilitarized Zone* (DMZ). Este trabalho proporcionou uma compreensão aprofundada das técnicas e considerações envolvidas na criação de uma rede empresarial robusta e segura.

Um componente crítico do trabalho desenvolvido envolveu a implementação do Zabbix, um *software open-source* de monitorização de redes e aplicações. O Zabbix foi utilizado para gerir e monitorizar a infraestrutura de TI da rede implementada, fornecendo dados valiosos sobre o desempenho e a segurança dos sistemas envolvidos. Este trabalho ilustra o poder e a flexibilidade do Zabbix como ferramenta de monitorização, bem como as suas potencialidades quando integrado com outras ferramentas.

Para aumentar a eficácia e a eficiência da solução, foram integradas outras ferramentas – o Grafana e o GLPi – ao Zabbix, todas instaladas numa única máquina. Além disso, foi desenvolvido um "script" personalizado para visualizar tarefas rotineiras no Zabbix a pedido de um colaborador da empresa. Este "script" permitiu automatizar e simplificar tarefas administrativas, melhorando a eficiência de monitorização.

Desenvolveu-se também uma aplicação em React Native com o objetivo de centralizar o acesso aos serviços anteriormente falados. Esta aplicação, embora inicialmente concebida num contexto de laboratório, apresenta um grande potencial para facilitar o trabalho dos colaboradores em ambiente empresarial. Isso significa que, se levada para um contexto empresarial, a aplicação poderá desempenhar um papel fundamental ao proporcionar uma plataforma unificada para aceder serviços de diferentes clientes. Isso não apenas aumentaria a eficiência operacional, mas também reduziria a complexidade para os colaboradores, permitindo que eles se concentrem nas suas tarefas principais, em vez de lidar com procedimentos complexos de configuração de endereços de *Internet Protocol* (IP). Além disso, essa centralização contribuiria para melhorar a padronização e a segurança no acesso aos serviços, promovendo boas práticas de cibersegurança em todos os níveis.

Na fase final do estágio, a atenção voltou-se para a implementação de *machine learning* na base de dados do Zabbix. Através da análise de *triggers* que surgem e são considerados como resolvidos automaticamente e rapidamente no Zabbix, pretende-se criar um limiar *(threshold*) que permita gerir de forma mais eficiente os problemas e melhorar a monitorização. Este trabalho inovador na aplicação de *machine learning*

representa uma contribuição valiosa para aprimorar a monitorização de redes e para a gestão proativa de problemas.

Em suma, o estágio permitiu uma experiência prática significativa na aplicação de técnicas e conceitos modernos de cibersegurança e monitorização de redes. Este relatório detalha as soluções implementadas, os desafios encontrados e as lições aprendidas durante este processo, fornecendo uma visão da complexa tarefa de gestão de redes e cibersegurança.

1.1. Motivação

Numa era digital em rápida evolução, a cibersegurança tornou-se um campo de extrema importância. A crescente complexidade e volatilidade das ameaças exigem a implementação de soluções robustas e eficientes que se possam adaptar rapidamente aos ambientes de rede em mudança. Diariamente, empresas de tecnologia em todo o mundo investem recursos significativos para garantir a cibersegurança e a resiliência de suas infraestruturas de TI. A monitorização de redes desempenha um papel integral neste processo e é aqui que o software Zabbix se torna crucial. No entanto, a eficácia deste software depende não apenas do uso adequado, mas também de uma compreensão profunda das suas funcionalidades, bem como da habilidade para personalizar e adaptar as suas configurações de acordo com as necessidades específicas de uma empresa.

Além disso, a integração de tecnologias de *machine learning* para aprimorar os processos de monitorização é um território ainda pouco explorado, mas apresenta um potencial tremendo para aumentar a eficiência e a eficácia dos sistemas de monitorização.

Esta investigação visa contribuir para o desenvolvimento de estratégias que possam otimizar a utilização do Zabbix no contexto da cibersegurança. Por meio de um estudo aprofundado da implementação e integração desse software com outras tecnologias, bem como da exploração da implementação de *machine learning* nos processos de monitorização, almejou-se fornecer uma contribuição valiosa para o campo da cibersegurança e para a comunidade de utilizadores do Zabbix.

1.2. Objetivos

Neste trabalho, o objetivo é desenvolver e otimizar a implementação e utilização do Zabbix numa rede. De maneira a alcançar este objetivo, seguiu-se várias etapas: começando com uma pesquisa aprofundada sobre implementações anteriores bem-sucedidas do Zabbix passando pela definição da arquitetura de rede, até à implementação de soluções que envolvem a integração do Zabbix com outros sistemas e tecnologias.

Este trabalho visa estabelecer novos padrões na utilização do Zabbix numa rede. Em vez de tentar superar implementações anteriores que evoluíram ao longo dos anos, a abordagem concentra-se nas melhores práticas e na inovação dentro deste projeto. Tal esforço é direcionado para a descoberta de novas possibilidades e melhorias, explorando caminhos não tradicionais no uso do Zabbix.

A avaliação dos resultados irá proporcionar um entendimento do progresso alcançado, bem como uma projeção de como a eficiência e a eficácia da monitorização da rede poderiam ser aprimoradas por meio da implementação proposta. Essa avaliação será um indicador fundamental do valor e impacto deste trabalho na melhoria da cibersegurança.

Outro objetivo crucial deste projeto é explorar a implementação de *machine learning* nos processos de monitorização, visando aprimorar a deteção de ameaças. A eficácia dessa abordagem será testada em diversos cenários, variando as características e padrões de tráfego da rede.

1.3. Caracterização sumária da instituição de acolhimento

A Securnet foi fundada em 2002, focando-se em manter organizações "Always online, Always Secure". Ao longo de mais de duas décadas, a empresa diversificou a sua oferta, apresentando soluções e serviços abrangentes em infraestruturas de TI. Esta evolução é corroborada pelos seus clientes.

A empresa destaca-se pela inovação, com soluções e serviços que são reconhecidos pelo valor, inovação e competitividade que trazem a seus clientes e parceiros. A relação transparente e de confiança com os seus clientes é uma das marcas registadas da Securnet, que conta com uma equipa técnica experiente e certificada.

No âmbito dos seus valores, a Securnet está direcionada para otimizar e rentabilizar os negócios dos seus clientes, oferecendo serviços nas áreas de segurança de dados, comunicações e sistemas. Desde a sua fundação, a empresa tem registado um crescimento contínuo, tanto em número de colaboradores como em faturação.

A visão da Securnet engloba vários pilares, incluindo a segurança simplificada, a disponibilidade da rede, o incremento da performance, a diminuição da exposição ao risco, e uma gestão centralizada da infraestrutura.

Em termos de acreditações, a Securnet possui certificações de relevância nacional e internacional, estando credenciada pela Autoridade Nacional de Segurança em graus como "NACIONAL SECRETO", "UE SECRET" e "NATO SECRET". Adicionalmente, faz parte da rede nacional de CSIRT's.

Por fim, a Securnet atribui grande valor às suas parcerias estratégicas, trabalhando com líderes tecnológicos mundiais, buscando sempre a excelência no serviço prestado através da soma de competências. [1]

1.4. Organização do documento

Este documento está dividido em sete capítulos, cada um explorando aspetos chave do projeto e do contexto. Cada capítulo contribui para uma compreensão abrangente dos desafios e soluções abordados neste trabalho.

O primeiro capítulo é a introdução, que fornece uma visão geral sobre o projeto e o seu contexto, providencia também uma visão geral sobre o restante documento.

O capítulo dois refere-se ao estado de arte deste projeto.

O terceiro capítulo tem como objetivo introduzir o leitor ao universo das redes informáticas, destacando a importância da engenharia de redes como um componente essencial na infraestrutura de qualquer empresa moderna. Neste capítulo, será explorado a configuração e implementação de uma rede simulada, reproduzindo, na medida do possível, uma infraestrutura de rede empresarial. Será dada ênfase à

configuração prática de uma firewall, de um switch e de servidores de *e-mail, web* e *SFTP*. Uma atenção especial será dedicada à implementação de uma DMZ, crucial para a proteção dos ativos de TI sensíveis.

No quarto capítulo, será abordada a configuração de um servidor Zabbix para monitorizar a rede que foi criada. Discutiremos detalhes sobre sua integração com Grafana e GLPi, além de explorar a criação de um script para visualizar tarefas rotineiras.

O quinto capítulo fala sobre a criação de uma aplicação mobile e web em React Native visando centralizar o acesso aos serviços implementados dentro da rede.

O capítulo seis foca-se na aplicação de técnicas de *machine learning* aos dados extraídos da base de dados do Zabbix. O intuito é refinar e priorizar a apresentação de eventos, minimizando o "ruído" gerado por eventos de menor relevância. A meta é otimizar o sistema para que as equipes de TI se concentrem nos eventos mais críticos

Finalmente, o sétimo e último capítulo deste relatório apresenta as conclusões do trabalho, sintetizando os principais *insights* e abordando possíveis melhorias e expansões para futuras investigações.

Capítulo 2

2. Estado de Arte

Neste capítulo, a atenção é direcionada para a análise das escolhas fundamentais de tecnologias e softwares feitas no âmbito deste projeto. É essencial compreender as razões subjacentes à seleção de soluções específicas, bem como as considerações cuidadosas que levaram a optar por essas tecnologias em detrimento de alternativas disponíveis. Ao longo deste capítulo, será realizada uma análise crítica das decisões de design e das ferramentas selecionadas, destacando como essas escolhas contribuíram para o sucesso e eficácia do projeto.

2.1. Tecnologias

A seguir, fornecerei uma breve explicação de cada uma das tecnologias e softwares utilizados:

Inicialmente, temos o OpenSSH, uma ferramenta vital para a segurança da rede e administração dos servidores, permitindo acesso seguro e gestão de servidores remotos, especialmente em sistemas Unix e Linux. Sendo *open-source*, a sua segurança é continuamente melhorada pela comunidade.

O Apache2 é um software open-source, e globalmente utilizado para hospedar páginas e aplicações web, destacando-se pela sua robustez e adaptabilidade. Roundcube simplifica a gestão de e-mails com uma interface intuitiva baseada em navegador.

No âmbito da segurança, temos o FortiOS, desenvolvido pela Fortinet. Este sistema operativo é o núcleo dos dispositivos como as *firewalls* Fortigate, trazendo uma ampla gama de recursos de segurança e uma gestão centralizada, ideal para ambientes empresariais complexos.

No mundo das bases de dados, o MariaDB Server destaca-se como uma alternativa eficiente do MySQL, sendo amplamente utilizado em aplicações web e empresariais. E para a monitorização de rede, o Zabbix oferece supervisão constantes dos recursos e deteção de eventos em tempo real.

O GLPi facilita a gestão de ativos e o Grafana amplia a visualização de dados com painéis personalizados, integrando-se perfeitamente com ferramentas como o Zabbix.

A linguagem de script Bash é uma ferramenta potente para automação e interação em sistemas Unix e Linux. No desenvolvimento de aplicações, o React Native possibilita criar apps nativas para várias plataformas a parir de uma única base de código, e, por fim, o Python, com a sua simplicidade e biblioteca vasta, é amplamente utilizado em áreas como desenvolvimento *web*, automação e *machine learning*.

2.2. Comparação com alternativas

A abordagem de seleção de tecnologia foi baseada em análises detalhadas das alternativas disponíveis no mercado e na indústria para cada uma das tecnologias selecionadas.

No contexto de soluções seguras de comunicação disponíveis, a escolha do OpenSSH foi resultado de análises cuidadosas em comparação com alternativas. O OpenSSH é amplamente reconhecido por sua confiabilidade e capacidade de fornecer conexões seguras e autenticadas. Em comparação com outras opções, como o Dropbear (uma alternativa mais leve), PuTTY (um cliente SSH para Windows), e soluções comerciais como o SSH Tectia e o Bitvise SSH Server, o OpenSSH se destacou por sua adoção generalizada e suporte ativo da comunidade, oferecendo segurança e desempenho confiáveis para comunicações SSH.

Em relação à variedade de servidores web disponíveis no mercado, a escolha do servidor web desempenha um papel fundamental na hospedagem de conteúdo online de forma confiável e eficiente. Nesse contexto, o Apache2 emergiu como a escolha prefe-

rencial para muitas organizações e indivíduos que buscam uma solução robusta e altamente personalizável para atender às suas necessidades de hospedagem na web. Com uma história de desenvolvimento extensa e uma ampla gama de recursos, o Apache2 continua a ser um dos servidores web mais confiáveis e populares do mundo. Explorar-se-á então, as razões por trás da escolha do Apache2 como servidor web e como as suas características e flexibilidade atendem às necessidades de hospedagem de conteúdo online:

Hospedagem web: O Apache2 é projetado para hospedar uma variedade de conteúdo web, incluindo HTML, CSS, JavaScript, imagens e vídeos, tornando-o uma escolha versátil.

Compatibilidade de Protocolos: Oferece suporte aos protocolos HTTP e HTTPS, garantindo comunicações seguras por meio de criptografia SSL/TLS.

Configuração Flexível: Disponibiliza uma ampla gama de configurações personalizáveis, permitindo que administradores adaptem o servidor às suas necessidades específicas, incluindo segurança, diretórios, módulos e recursos.

Módulos e Extensões: Sua natureza modular permite expandi-lo com módulos e extensões, adicionando funcionalidades específicas, como suporte a linguagens de programação e autenticação avançada.

Virtual Hosts: Suporta configuração de múltiplos "Virtual Hosts," permitindo hospedar vários sites num único servidor físico.

Gestão de Acessos e Segurança: Oferece recursos avançados de gestão de acessos, incluindo autenticação, autorização baseada em regras e segurança de IP.

Documentação e Comunidade: Possui documentação online abrangente e uma comunidade ativa que oferece suporte e recursos adicionais.

Sistemas Operativos Suportados: É compatível com várias plataformas, incluindo sistemas Unix (Linux, BSD, macOS) e Windows.

Open-source: Sendo um projeto open-source, é gratuito e personalizável para atender às necessidades dos utilizadores.

Em comparação com alternativas como Nginx e LiteSpeed, o Apache2 destaca-se pela sua longa história de desenvolvimento e pela sua capacidade de adaptação, tornando-o uma escolha popular para empresas e indivíduos que buscam um servidor web confiável e altamente configurável para hospedar conteúdo online.

A escolha do Roundcube como servidor de e-mail foi óbvia, superando facilmente as alternativas disponíveis devido às suas excelentes características e desempenho. A seguir, será explorado em detalhe as características que tornam o Roundcube a opção preferencial para as necessidades de gestão de e-mails:

Interface Web Intuitiva: Em comparação com algumas alternativas que podem apresentar interfaces complicadas, o Roundcube oferece uma experiência de utilizador semelhante à de clientes de e-mail tradicionais, garantindo facilidade de uso para todos.

Funcionalidades Avançadas: O Roundcube oferece um conjunto completo de funcionalidades de e-mail, incluindo gestão de pastas, pesquisa, organização de marcadores e regras de filtragem.

Suporte a Múltiplas Contas: O Roundcube permite que os utilizadores gerem várias contas numa única *interface*.

Editor de E-mails: O Roundcube oferece um editor robusto para criar mensagens bem formatadas.

Segurança e Privacidade: Com opções de autenticação segura e sendo open source, os utilizadores têm mais controle sobre a segurança das suas comunicações em comparação com alternativas proprietárias.

Suporte IMAP e SMTP: O Roundcube é compatível com protocolos padrão, permitindo que os utilizadores acedam aos seus e-mails em servidores remotos e enviem e-mails de forma eficaz.

Comunidade Ativa: Ao contrário de algumas alternativas menos suportadas, o Roundcube possui uma comunidade ativa de utilizadores e desenvolvedores, garantindo atualizações regulares e suporte contínuo.

Comparado a alternativas como SquirrelMail e Horde, o Roundcube destaca-se devido à sua *interface* intuitiva semelhante a clientes de e-mail tradicionais. Este software oferece funcionalidades avançadas, como gestão de pastas, pesquisa de e-mails e suporte a múltiplas contas, tornando-o uma escolha mais abrangente. Além disso, o Roundcube prioriza a segurança, oferecendo opções de autenticação segura. Sendo uma aplicação open-source, os utilizadores têm mais controle sobre a privacidade de suas comunicações em comparação com alternativas proprietárias.

O sistema de gestão de base de dados escolhida foi o MariaDB Server. As principais características do RDBMS, que fundamentam a minha decisão de optar por este sistema, abrangem:

Open-source: MariaDB é de código aberto, o que significa que é gratuito para usar e modificar. Isso também implica um desenvolvimento transparente e a capacidade de uma comunidade ativa de contribuir para o seu progresso.

Compatibilidade com MySQL: MariaDB foi desenvolvido para ser um substituto direto do MySQL, o que significa que ele é, na sua maioria, compatível a nível binário com o MySQL.

Recursos Avançados: Em relação ao MySQL, MariaDB introduziu várias melhorias e novas funcionalidades, como o armazenamento de motor Aria, armazenamento colunar com o MariaDB ColumnStore e mais.

Segurança: MariaDB inclui várias características de segurança, como a encriptação de dados em repouso e durante a transmissão, e suporta uma vasta gama de mecanismos de autenticação.

Desempenho e Escalabilidade: O MariaDB é conhecido pelo seu desempenho superior, otimizações e a capacidade de lidar com grandes volumes de dados e solicitações.

Storage Engines: MariaDB suporta vários motores de armazenamento, permitindo aos utilizadores escolher o mais adequado para suas necessidades específicas. Alguns dos motores de armazenamento incluem InnoDB, Aria, MyRocks, e TokuDB, entre outros.

Replicação e *Clustering:* Para garantir alta disponibilidade e desempenho, MariaDB suporta várias opções de replicação e *clustering*.

No contexto de monitorização de rede, o Zabbix emergiu como escolha devido à imposição da empresa. É de notar que este software oferece uma interface amigável, facilidade de configuração e ampla comunidade de suporte. Esses fatores destacam o Zabbix em relação a alternativas como Nagios e SolarWinds, que frequentemente pedem configurações complexas e podem resultar em custos substanciais em comparação com a opção de código aberto do Zabbix.

Na gestão de ativos e integração com o sistema de monitorização, o GLPi surgiu como opção devido às escolhas de software já implementadas na empresa. Este software oferece uma interface intuitiva e funcionalidades abrangentes. Isso foi especialmente evidente em comparação com alternativas como o OCS Inventory, que apresenta algumas limitações em termos de gestão de ativos. Além disso, a capacidade do GLPi de integração direta com o Zabbix simplifica esta decisão.

Para a visualização de dados e criação de painéis de monitorização, o Grafana foi escolhido devido à sua flexibilidade e facilidade de uso, bem como pela integração perfeita com o Zabbix. Comparativamente, outras ferramentas como Kibana, focadas em análises de ficheiros log, e o Tableau, apesar de suas capacidades, não ofereciam a mesma combinação de recursos e integração.

A decisão de utilizar a linguagem Bash para o desenvolvimento do script, que será referenciado posteriormente no capítulo 4, foi motivada pela minha experiência prévia com essa tecnologia e pela eficiência comprovada que ela oferece. Embora existam alternativas como Python e Perl para scripts, a familiaridade prévia com a linguagem Bash e a sua natureza orientada a tarefas do sistema destacaram-na como a escolha preferencial. A linguagem Bash simplifica o processo de automação de tarefas do sistema, tornando-a altamente eficaz e economizando consideravelmente tempo de desenvolvimento.

Para o desenvolvimento de aplicações móveis e web, a escolha do React Native foi influenciada pela minha experiência prévia com a tecnologia e pela economia significativa de tempo que ela proporciona. Enquanto outras opções, como Flutter e Xamarin, também oferecem compatibilidade multi-plataforma, a maturidade da comunidade de desenvolvedores do React Native e a facilidade de integração com bibliotecas de terceiros destacaram esta tecnologia como a escolha preferencial.

A decisão de empregar a linguagem Python para este projeto foi influenciada pela sua notável versatilidade e pela sólida base de bibliotecas e frameworks disponíveis na comunidade. Embora existam outras linguagens adequadas para *Machine Learning*, como R ou Julia, a escolha do Python foi motivada pela sua ampla adoção na comunidade de *Data Science* e *Machine Learning*. A vasta gama de bibliotecas, como *TensorFlow*, *scikit-learn* e *PyTorch*, tornou o Python a escolha preferencial para desenvolver modelos de *Machine Learning* de alta qualidade e implementar soluções eficazes para este desafio específico.

Essas análises diretas com alternativas enfatizam por que se escolheu as tecnologias específicas para este projeto, destacando os benefícios exclusivos que cada uma delas oferece em relação às opções concorrentes.

2.3. Escolha de tecnologias e software

A seleção criteriosa das tecnologias e softwares desempenha um papel fundamental no sucesso de qualquer projeto. Nesta secção, detalharei minuciosamente as razões que motivaram a escolha específica das seguintes ferramentas: OpenSSH, Apache2, Roundcube, FortiOS, Zabbix, React Native, GLPi, Grafana, Python e Bash:

OpenSSH: A escolha do OpenSSH como parte essencial do projeto foi fundamentada na sua robusta capacidade de fornecer comunicações seguras através de protocolos criptografados. A sua ampla adoção e confiabilidade tornaram-no a escolha natural para estabelecer conexões seguras e autenticadas entre sistemas e dispositivos.

Apache2: O Apache2 foi selecionado como nosso servidor web devido à sua longa história de confiabilidade e sua capacidade de lidar com uma variedade de tarefas de hospedagem e servidores web. Sua flexibilidade e suporte à personalização são características cruciais para atender às necessidades específicas do projeto.

Roundcube: A escolha do Roundcube como cliente de email web foi motivada pela sua interface intuitiva e pela facilidade de integração com servidores de email. Isso proporcionou aos utilizadores uma experiência de email eficiente e acessível, simplificando a comunicação interna.

FortiOS: A escolha do FortiOS como sistema operativo de segurança foi determinada pela disponibilidade de equipamentos da marca Fortinet na empresa. Dada a compatibilidade intrínseca entre o FortiOS e os dispositivos Fortinet, a decisão de adotar esta solução revelou-se estratégica e eficaz. Isso permitiu aproveitar ao máximo os recursos e as capacidades dos dispositivos existentes, garantindo um ambiente de segurança coeso e de alto desempenho.

MariaDB Server: A implementação do MariaDB Server como base de dados foi uma escolha estratégica para otimizar o armazenamento e a gestão de dados críticos. O MariaDB Server destacou-se pela sua capacidade de oferecer um ambiente robusto e escalável para armazenar informações de forma organizada, simplificando o acesso e a gestão de dados essenciais para este projeto.

Zabbix: Optou-se pelo Zabbix devido à sua sólida reputação como uma das principais soluções de monitorização de rede disponíveis. Sua flexibilidade e capacidade de escalabilidade foram fatores cruciais, permitindo monitorizar eficazmente a infraestrutura de rede. Além disso, é importante notar que a escolha do Zabbix também foi influenciada pela política da empresa de padronização em sistemas de monitorização.

GLPi: A integração do GLPi em ambiente de monitorização foi uma escolha estratégica para otimizar a gestão de ativos. O GLPi destacou-se pela sua capacidade de fornecer uma visão organizada de dispositivos monitorizados, facilitando a identificação e resolução de incidentes. É importante notar que a escolha do GLPi foi também influenciada pelas diretrizes da empresa em termos de gestão de ativos e integração com o sistema de monitorização.

Grafana: Optou-se pelo Grafana para a visualização de dados e criação de painéis personalizados de monitorização. A decisão baseou-se na sua ampla gama de recursos de gráficos interativos e na capacidade de integração perfeita com o Zabbix. Isso permitiu apresentar informações complexas de forma acessível e altamente informativa.

Bash: O Bash foi selecionado para automação de tarefas e scripts devido à sua ampla disponibilidade em sistemas Unix e Linux. A sua simplicidade e poder tornam-no uma escolha sólida para a criação de scripts de administração e gestão de sistemas.

React Native: A escolha pelo React Native para o desenvolvimento da aplicação móvel e web foi motivada pela sua versatilidade e eficiência. Esta tecnologia permitiu criar uma aplicação única que funciona de forma consistente em diversas plataformas, economizando tempo e recursos de desenvolvimento. Também é relevante mencionar que a familiaridade prévia com o React Native, devido a experiências anteriores, contribuiu para a decisão de adotar essa tecnologia.

Python: O Python foi selecionado como uma das principais linguagens de programação devido à sua versatilidade, legibilidade de código e forte suporte à programação orientada a objetos. Sua sintaxe limpa e estruturada facilita o desenvolvimento de código robusto e de fácil manutenção. Além disso, a vasta biblioteca padrão do Python oferece um conjunto abrangente de módulos para lidar com tarefas como manipulação de dados, automação de processos e interação com APIs, economizando tempo e esforço no desenvolvimento. Sua capacidade de integração transparente com outras tecnologias e a portabilidade entre plataformas tornam-no uma escolha ideal para garantir a eficiência e a escalabilidade do projeto.

Essas escolhas foram motivadas pela clara convergência das tecnologias escolhidas com os requisitos do projeto, incluindo monitorização de rede em tempo real, centralização de acesso através da aplicação móvel e web, e integração eficaz entre todas as ferramentas para uma gestão eficiente da infraestrutura de rede. Ao longo deste capítulo, será explorado como essas decisões se traduziram em resultados concretos e benefícios tangíveis para o projeto, tendo em consideração tanto as exigências técnicas quanto as diretrizes da empresa.

Capítulo 3

3.Introdução às redes informáticas

Neste capítulo, será explorado o vasto universo das redes informáticas, desvendando os detalhes que tornam as redes tão fundamentais nas operações empresariais modernas. Será explicado, de forma abrangente, a importância intrínseca da engenharia de redes, que se estabelece como o alicerce que sustenta a arquitetura de comunicação no contexto empresarial. Esta explicação irá além da superfície, explorando as complexidades da configuração e implementação de uma rede, direcionando um olhar atento para a infraestrutura subjacente das redes empresariais. O foco será concentrado de forma intensa na configuração prática dos elementos vitais, tais como firewall, switch e servidores, evidenciando a relevância de destaque da implementação de uma DMZ como um elemento crucial para preservar a inviolabilidade dos recursos informáticos sensíveis da organização.

3.1. Importância das redes informáticas

A rápida evolução tecnológica trouxe consigo uma dependência cada vez maior das redes informáticas para garantir o funcionamento fluido das atividades quotidianas das empresas. A capacidade de comunicação instantânea e a partilha de recursos de forma eficaz transformaram-se numa coluna vertebral indispensável, que sustenta a otimização das operações organizacionais. Neste cenário dinâmico, a engenharia de redes emerge como um elemento de destaque, desempenhando um papel fundamental ao viabilizar a criação, implementação e manutenção de infraestruturas de rede altamente flexíveis, capazes de se adaptar com agilidade às mudanças incessantes nas exigências empresariais.

A interconexão global impulsionada pela revolução tecnológica promoveu uma interdependência cada vez mais profunda nas redes informáticas, que agora são essenciais para uma ampla gama de operações empresariais. As fronteiras tradicionais cederam espaço para a colaboração virtual e a troca instantânea de informações em escala global, permitindo às organizações explorar mercados distantes e interagir com clientes de maneiras nunca imaginadas. Essa conectividade ampla e instantânea exige redes ágeis e robustas, capazes de se ajustar aos picos de tráfego, manter a integridade dos dados e garantir a continuidade das operações em qualquer situação.

No centro dessa realidade dinâmica, a engenharia de redes desempenha um papel que vai além da simples gestão de dispositivos e cabos. Esta tornou-se numa arquitetura dos sistemas de comunicação modernos, projetando e implementando redes que não apenas atendem às necessidades imediatas das organizações, mas também antecipam as transformações que estão no horizonte. A engenharia de redes trabalha em estreita colaboração com as metas estratégicas da empresa, assegurando que a infraestrutura de rede seja um trampolim para o crescimento e a inovação.

Ao lidar com a complexidade das redes informáticas, a engenharia de redes enfrenta o desafio constante de equilibrar a acessibilidade com a segurança. A democratização do acesso à informação é uma faceta fundamental da era digital, mas também traz consigo ameaças crescentes de cibersegurança. Portanto, além de criar redes eficientes, a engenharia de redes também se concentra na implementação de medidas robustas de proteção. A criação de DMZ, como mencionado anteriormente, destaca-se como uma estratégia fundamental para preservar a confidencialidade e integridade dos dados sensíveis, mitigando riscos potenciais.

Em última análise, a engenharia de redes transcende o aspeto técnico para se tornar um componente vital do sucesso empresarial. Ela é responsável por habilitar a colaboração, a inovação e a resiliência num mundo onde a conectividade é a espinha dorsal das operações. A capacidade de criar, manter e aprimorar redes informáticas eficazes e seguras é o que permite que as organizações prosperem nu ambiente de negócios cada vez mais digital e interligado.

3.1.1. Projeto e planificação de redes

O processo intrínseco da engenharia de redes tem início com um planeamento minucioso e detalhado. Os engenheiros especializados mergulham na análise dos requisitos específicos da organização, abrangendo elementos cruciais como a topologia da rede, os dispositivos imprescindíveis para a operação, os protocolos que facilitam a comunicação e as medidas de segurança que resguardam os ativos digitais. A soma desses detalhes meticulosamente investigados culmina na elaboração de um plano estratégico que delineia não somente a arquitetura da rede, mas também os componentes fundamentais que a compõem. Esse plano não é apenas um esboço estático, mas sim um projeto dinâmico que forja uma base sólida e confiável para a subsequente implementação da infraestrutura de rede.

O plano elaborado pelos engenheiros de redes não é simplesmente um conjunto de diretrizes, mas sim um roteiro que traça o caminho para a construção de uma rede adaptável e eficaz. Cada aspeto do plano é cuidadosamente calibrado para garantir que a rede atenda não apenas às necessidades imediatas, mas que também se possa expandir e evoluir em sintonia com as mudanças no ambiente empresarial. Esse plano não é apenas um pedaço de papel, mas sim um guia que assegura que as decisões tomadas na implementação da rede estejam perfeitamente alinhadas com as metas da organização. Ao estabelecer uma base sólida, o processo de planeamento se revela como um passo crucial para uma engenharia de redes bem-sucedida, capaz de construir sistemas de comunicação que são robustos, escaláveis e capazes de se adaptar a um mundo empresarial em constante evolução.

3.1.2. Implementação e configuração

Com o plano meticulosamente elaborado em mãos, os engenheiros de redes dão o próximo passo e avançam para a fase de implementação. Esse processo engloba a configuração minuciosa de uma série de dispositivos fundamentais, tais como *switches, routers, firewalls* e servidores. O foco principal recai sobre a harmonização e a

cooperação desses dispositivos de modo a atender de forma fluída e eficaz às necessidades operacionais da organização. O cenário complexo da engenharia de redes exige não apenas a capacidade técnica para configurar esses componentes, mas também uma visão holística que garanta que cada peça do quebra-cabeça se encaixe perfeitamente.

Além da configuração estrutural, a engenharia de redes também abrange a implementação de políticas de segurança rigorosas e a segmentação estratégica da rede. A cibersegurança emerge como uma consideração primordial, tendo em vista o ambiente digital cada vez mais permeável a ameaças. A implementação de medidas de proteção abrange desde firewalls que atuam como sentinelas digitais até a criação de áreas isoladas para salvaguardar dados sensíveis. A engenharia de redes desempenha, assim, um papel duplo: não apenas assegura o funcionamento técnico da infraestrutura, mas também resguarda a integridade e a confidencialidade dos ativos valiosos da organização.

O processo de implementação é uma simbiose de habilidades técnicas e compreensão estratégica. É uma coreografia delicada que exige que cada movimento seja preciso e bem coordenado para que a orquestra de dispositivos funcione harmoniosamente. A meticulosidade é a tónica dessa etapa, onde cada ajuste, cada configuração e cada protocolo têm um propósito específico. A engenharia de redes não é apenas uma questão de conectar dispositivos, mas sim de criar um ecossistema digital que se adapte aos desafios do presente e esteja pronto para abraçar as oportunidades do futuro.

3.1.3. Monitorização e manutenção contínuas

A monitorização contínua emerge como um elemento de vital importância, tendo como propósito primordial a deteção precoce de problemas, a avaliação meticulosa do desempenho e o reforço ininterrupto da segurança. As ferramentas de monitorização assumem um papel crucial, permitindo que os engenheiros estejam vigilantes para identificar quaisquer *bottlenecks* que possam surgir, antecipar picos de tráfego iminentes e tomar medidas proativas para manter a rede num estado de otimização

constante. Essa abordagem proativa transcende a mera manutenção e transforma-se num sistema de defesa que mantém a rede resiliente e ágil em face das exigências em constante mudança.

No cerne desse processo, a manutenção regular emerge como um pilar fundamental. Através de verificações rotineiras, ajustes e atualizações, os engenheiros de redes garantem que a infraestrutura permaneça coesa e livre de problemas latentes. A adaptabilidade é a palavra de ordem, uma vez que as operações empresariais estão em constante evolução. À medida que novos requisitos surgem e os objetivos organizacionais se refinam, a engenharia de redes se ajusta de acordo, assegurando que a rede continue a ser uma ferramenta ágil e eficaz para a consecução dos objetivos da empresa.

O ciclo de monitorização e manutenção perpetua-se como uma dança constante, sincronizada com as oscilações das necessidades empresariais. A sua importância vai além do mero funcionamento técnico, incorporando uma perspetiva mais ampla: a rede é uma extensão dos objetivos da organização. Garantir a sua saúde e eficácia não é apenas uma questão de manter a tecnologia em ordem, mas também de assegurar que a empresa esteja bem preparada para prosperar num ambiente empresarial em constante evolução. Nesse contexto, a monitorização e manutenção constantes evoluem de tarefas técnicas para pilares essenciais que sustentam a agilidade, a resiliência e o sucesso duradouro da organização.

3.2. Montagem prática de uma infraestrutura de rede com paralelos empresariais

No contexto deste projeto, foi realizada a configuração de uma rede que espelha a complexidade de uma infraestrutura empresarial real. A concretização dessa configuração prática envolveu não apenas a instalação de dispositivos, mas também a criação de um ecossistema digital que se assemelha às redes utilizadas por organizações no mundo real. Essa simulação contemplou a implementação de servidores de *e-mail,*

web e *SFTP*, bem como a instalação meticulosa de uma firewall e de um switch. Esses elementos, cada um com sua funcionalidade específica, foram habilmente orquestrados para formar uma infraestrutura coesa que reflete a interconexão complexa que ocorre em ambientes empresariais.

Um dos ênfases primordiais nessa simulação concentrou-se na construção de uma *DMZ*, um aspeto crítico que fortalece a segurança da rede. A *DMZ* trata-se de uma sub-rede isolada projetada para proteger ativos informáticos sensíveis de possíveis ameaças. Por meio dessa configuração estratégica, recursos como servidores web e e-mail estão localizados numa área intermediária entre a rede interna e a externa, minimizando a exposição direta a potenciais riscos cibernéticos. Essa abordagem contribui para a defesa da integridade dos dados corporativos e da confidencialidade das informações, garantindo que a organização esteja preparada para lidar com ameaças digitais num ambiente cada vez mais complexo e conectado.

O resultado dessa configuração simulada não é apenas um exercício técnico, mas sim uma prova tangível da complexidade e importância das redes empresariais modernas. Ao realizar essa simulação, os engenheiros de redes não apenas dominam as habilidades técnicas, mas também internalizam a necessidade de criar ambientes digitais seguros, eficientes e altamente funcionais. Eles se tornam arquitetos digitais capazes de criar a base sólida sobre a qual as organizações modernas podem construir as suas operações, mantendo a integridade e a confidencialidade das informações num cenário digital em constante evolução.

3.2.1. Engenharia de rede

No âmbito deste projeto, a engenharia de redes foi empregue para conceber uma configuração que simula uma topologia de rede equiparada a um ambiente empresarial. Utilizando com destreza os recursos já existentes, foi-se capaz de delinear um ambiente que não apenas replica a complexidade inerente às infraestruturas reais, mas também incorpora os desafios singulares encontrados nessas configurações do mundo real.

Ao examinar os recursos à minha disposição, fiz uma meticulosa análise e escolhi a topologia que melhor se adequava ao contexto. A topologia escolhida encontra-se representada na Figura 1.

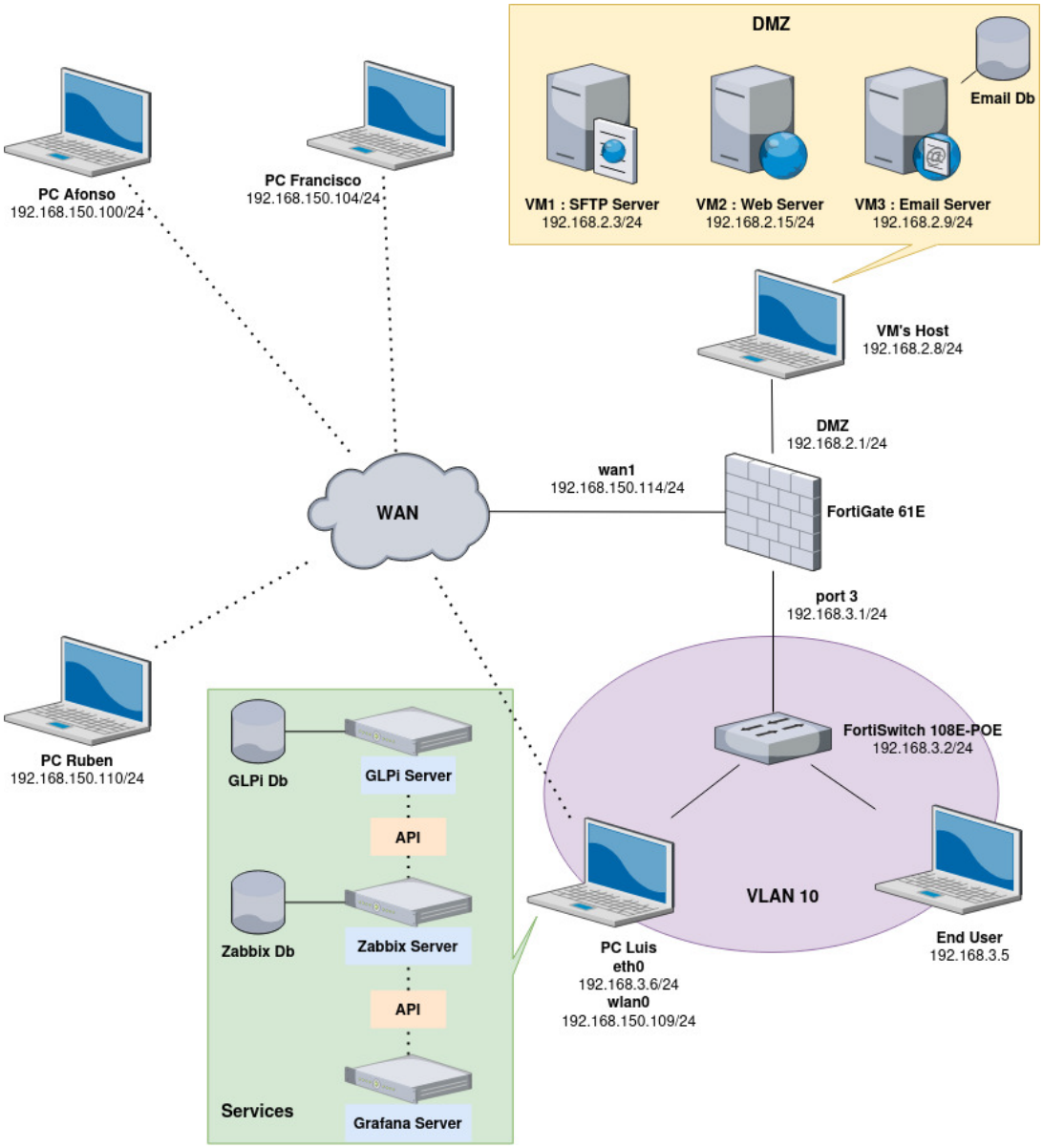


Figura 1 - Arquitetura de rede

A Figura 1 ilustra a concretização da configuração, a qual destaca a implementação estratégica de uma DMZ, a qual foi construída sobre um computador e sustentada por três VM's, cada uma designada para operar como servidor web, servidor de e-mail e servidor SFTP. Ademais, a topologia incorpora uma VLAN composta por um

computador equipado com softwares de monitorização de vanguarda, como o Zabbix, GLPi e Grafana, além de um utilizador final. É digno de notar que também são apresentados os computadores de outros membros da equipa, que estão interligados à WAN.

A conceção e implementação dessa topologia simulada não se limitam à manipulação de dispositivos e conexões, mas envolvem um entendimento profundo da estrutura subjacente da rede e a habilidade de traduzir esse conhecimento numa configuração funcional e coerente. Ao aplicar a engenharia de redes a essa simulação, obteve-se uma visão valiosa sobre a complexidade de orquestrar diversos elementos num ambiente unificado, ao mesmo tempo em que se reforçou a importância de escolhas estratégicas na criação de infraestruturas robustas e seguras. Em suma, essa aplicação prática da engenharia de redes demonstra a convergência entre conhecimento técnico e visão estratégica necessária para construir as bases sólidas das redes empresariais modernas.

3.2.2. Configuração da DMZ e Firewall

No centro da nossa abordagem está a criação de uma *DMZ*. A *DMZ* é uma sub-rede que age como uma barreira de segurança entre a rede interna e a rede exterior. Aqui implementamos uma *firewall* que atua de modo a proteger esta mesma *DMZ*, protegendo os recursos críticos da empresa de ameaças externas. A *firewall* é configurada com regras restritas para controlar o tráfego entre a *WAN*, a *DMZ* e a rede interna. Essas regras são fundamentais para garantir que apenas o tráfego autorizado seja permitido, minimizando o risco de ataques. A configuração das *VM*'s criadas de maneira a simular uma *DMZ* estão demonstradas em Anexo 1, sendo que o processo é demonstrado apenas uma vez visto que a configuração foi igual para os três servidores criados. As Figuras 2 e 3 demonstram a interação com o servidor *SFTP* criado e configurado, como dito anteriormente, a partir do computador alocado na VLAN10.

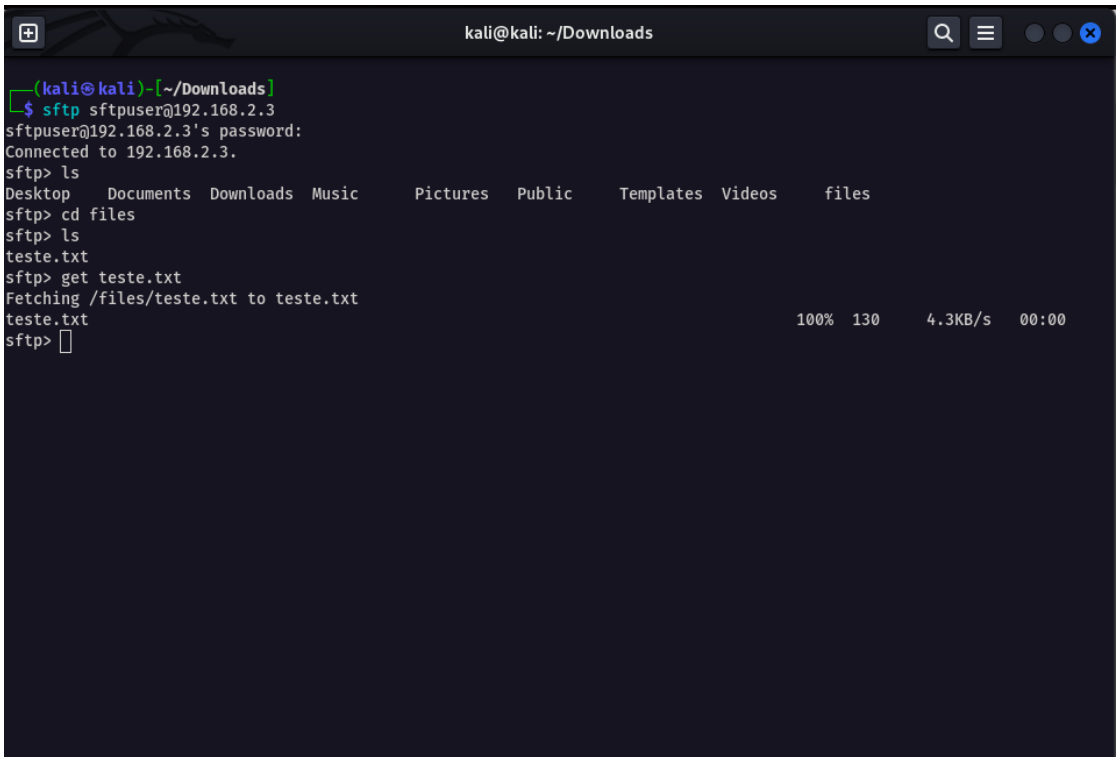


Figura 2 - Ligação SFTP ao servidor via terminal e download de ficheiro

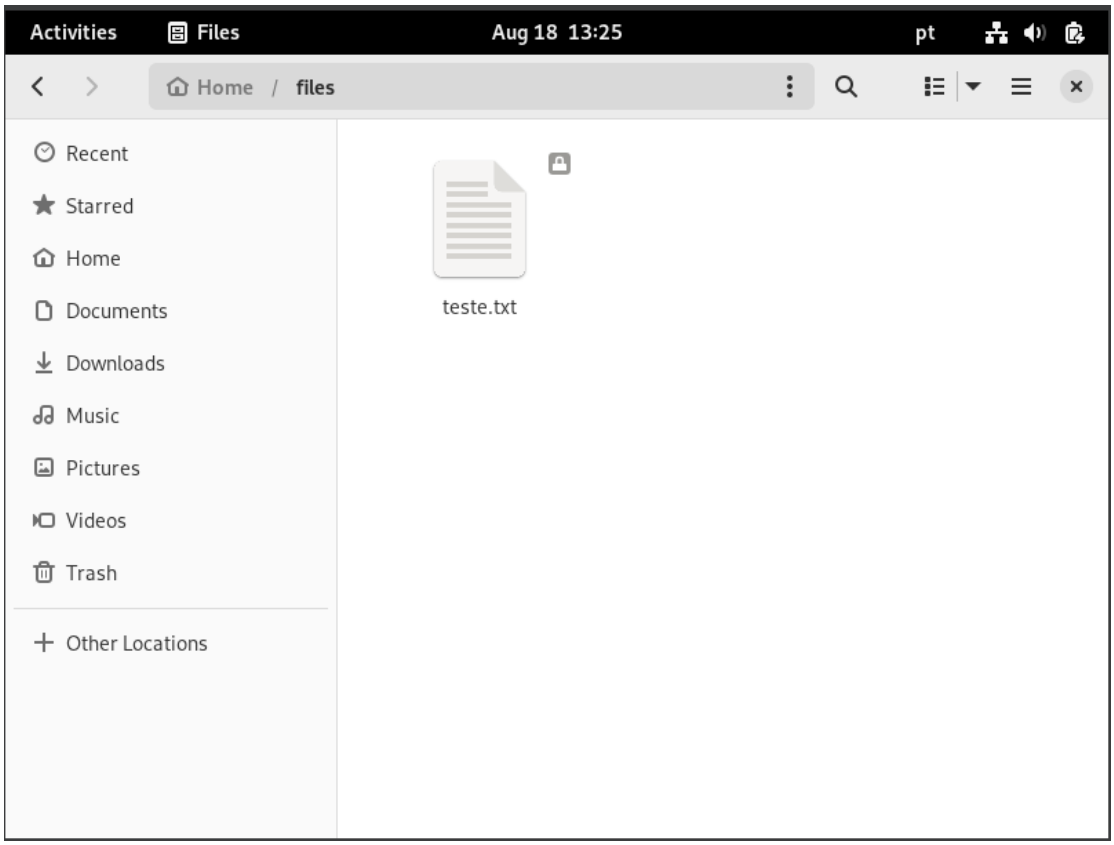


Figura 3 - Diretório do servidor SFTP com o ficheiro descarregado

O protocolo utilizado, SFTP, é um protocolo seguro de transferência de ficheiros entre sistemas remotos. Ao contrário do FTP convencional, que transmite dados, incluindo credenciais de autenticação de forma não encriptada, o SFTP foi projetado para garantir a confidencialidade e a integridade dos dados durante o processo de transferência.

O SFTP opera uma ligação SSH, que fornece uma camada adicional de segurança ao criar um túnel encriptado entre o cliente e o servidor. Isso significa que todos os dados transmitidos através de uma sessão SFTP são codificadas, protegendo-as de acesso não autorizado e potenciais ataques.

As principais características deste protocolo são:

Segurança: O SFTP utiliza encriptação para proteger os dados durante a transferência, minimizando o risco de acesso não autorizado ou roubo de informações sensíveis.

Autenticação: O SFTP aproveita as capacidades de autenticação do SSH. Isso significa que os utilizadores precisam de credenciais válidas (nome de utilizador e palavra-passe ou chaves SSH) para se autenticarem no servidor antes de transmitirem ficheiros.

Manipulação de ficheiros: O SFTP permite aos utilizadores transferir, descarregar, apagar, renomear e gerir ficheiros e diretórios no servidor remoto, de forma semelhante ao FTP tradicional.

Integridade: Além de encriptação, o SFTP também ajuda a garantir a integridade dos dados transferidos, assegurando que não sejam alterados durante o processo de transferência.

Portabilidade: O SFTP é amplamente suportado em diversas plataformas e sistemas operativos, tornando-o uma escolha popular para transferência segura de ficheiros.

Configuração: A configuração do SFTP pode envolver ajustes de permissões, definição de diretórios de acesso e configuração de chaves SSH para autenticação.

Sendo assim, este protocolo utilizado continua a ser uma solução confiável para assegurar que as transferências de ficheiros sensíveis ou confidenciais ocorram de maneira segura, garantindo a privacidade e a proteção dos dados durante todo o processo. A configuração feita para a criação deste servidor SFTP está disponibilizada em Anexo 2.

O servidor web foi configurado através do software Apache2 referido e caracterizado no Capitulo 2 - Estado de Arte.

O código implementado e a configuração feita para a criação do servidor web encontram-se disponíveis em Anexo 3. De referenciar que o código implementado já

tinha sido desenvolvido anteriormente, mas foi usado neste contexto de modo a tornar o servidor web mais apelativo esteticamente.

O acesso a este servidor a partir de outra máquina na rede é ilustrada na figura seguinte, Figura 4.

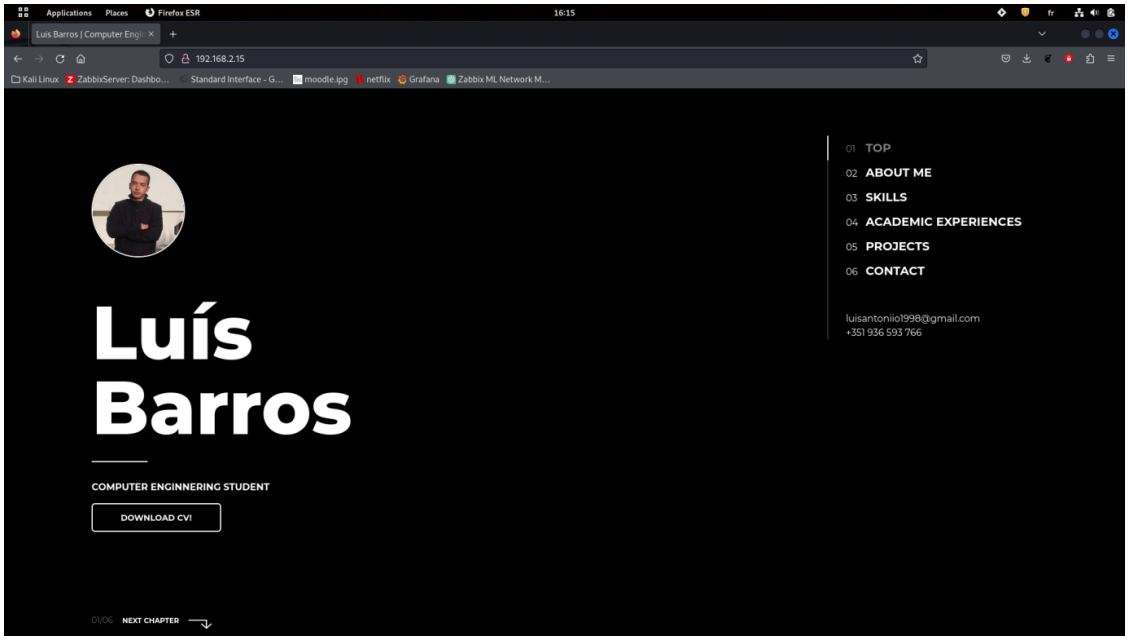


Figura 4 - Servidor web

O servidor de e-mail foi implementado a partir da instalação e configuração do *software open source*, Roundcube. Instalação e configuração essa demonstrada no Anexo 4. Este *software* encontra-se referenciado e caracterizado no Capitulo 2 - Estado de Arte.

Devido á sua natureza open source, o Roundcube pode ser personalizado e estendido para atender às necessidades especificas de uma organização ou de um utilizador individual, tornando-a uma solução de e-mail acessível e funcional.

As figuras seguintes, Figura 5 e 6 ilustram o acesso ao servidor de e-mail a partir de outra máquina que se encontra alojada na rede.

Figura 5 - Acesso login ao serviço E-mail

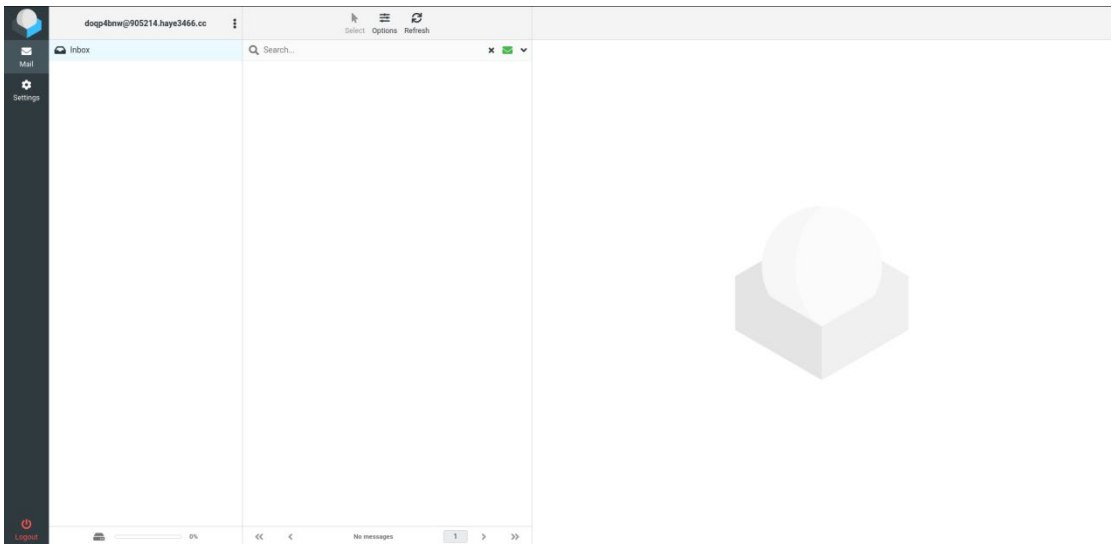


Figura 6 - Inbox do Roundcube

A configuração do servidor e-mail pode ser visualizada em Anexo 4.

Com a implementação de serviços de servidor *SFTP*, servidor *web* e servidor de *e-mail*, a infraestrutura de rede empresarial ganha um conjunto de recursos robustos para facilitar as operações internas e melhorar a comunicação e a colaboração entre os utilizadores. Cada serviço é configurado para garantir a segurança, integridade e privacidade de dados, permitindo uma utilização eficiente e protegida dos recursos informáticos sensíveis da empresa.

No cerne da minha abordagem de segurança está a configuração da firewall. A firewall atua como uma barreira defensiva entre as redes internas e externas, desempenhando um papel crucial na proteção dos recursos críticos da empresa contra ameaças externas. Neste projeto, foi implementado uma firewall Fortigate 61E para gerir o tráfego entre a *WAN*, a *DMZ* e a rede interna.

A FortiGate 61E é um dispositivo de segurança de rede fabricado pela Fortinet, projetado para fornecer proteção e gestão de segurança avançados para redes empresariais. Faz parte da série FortiGate, que fornece uma variedade de recursos de segurança, incluindo *firewall*, proteção contra ameaças, *VPN*, inspeção *SSL*, controle de aplicações e muito mais. De referenciar que esta firewall foi projetada para atender às necessidades de segurança de redes de pequenas e médias empresas. [2]

A configuração da firewall é um processo complexo que requer um entendimento profundo das necessidades de segurança da organização e das boas práticas de segurança. No processo da configuração da firewall, diversos passos cruciais foram implementados. Esses passos incluem:

Definição de interfaces: As interfaces de várias redes foram cuidadosamente configuradas, incluindo a WAN, DMZ e VLAN10. Endereços IP apropriados e máscaras de sub-rede foram atribuídos a cada interface, permitindo a segmentação adequada da rede.

Estabelecimento de políticas de firewall: Para gerir o fluxo de tráfego entre as redes, foram criadas políticas de firewall detalhadas. Essas políticas foram projetadas para regular o tráfego de entrada e saída com base em critérios como a origem, destino, protocolo e portas utilizadas. Isso permitiu uma abordagem granular para permitir ou bloquear o tráfego conforme necessário.

Configuração regras de NAT: A implementação de regras NAT permitiu que o tráfego da WAN fosse corretamente traduzido para os endereços IP da DMZ e da VLAN10. Isso possibilitou uma comunicação eficaz entre as redes, garantindo que os dispositivos na DMZ e VLAN10 pudessem comunicar com recursos externos de maneira segura.

Configuração SNMP: O SNMP permite a monitorização eficaz da firewall a partir do Zabbix. Isso proporciona a recolha em tempo real de informações sobre o estado e o tráfego da firewall, contribuindo para uma monitorização proativa e uma resposta ágil a quaisquer problemas ou anomalias que possam surgir. Essa abordagem integrada

fortalece ainda mais a segurança e o desempenho da infraestrutura de rede da empresa, garantindo uma operação contínua e confiável.

A configuração realizada na FortiGate foi minuciosamente detalhada e está devidamente documentada nos Anexo 5, proporcionando uma referência completa e detalhada para todas as etapas e configurações implementadas.

3.2.3. Configuração do Switch e VLANs

Com a sólida infraestrutura da DMZ estabelecida e a proteção da firewall efetivamente garantida, a jornada pela engenharia de redes prossegue com a configuração minuciosa do switch. Nesta etapa crucial, o FortiSwitch 108E-POE assume um papel central, assegurando que o tráfego seja direcionado com precisão e eficácia entre os dispositivos interligados na rede interna.

O switch, como componente fundamental de qualquer rede empresarial, desempenha a função de encaminhar o tráfego de dados, garantindo que estes alcancem o seu destino de maneira otimizada. No contexto deste projeto, a configuração detalhada do switch é uma peça essencial do quebra-cabeça da engenharia de redes, uma vez que sua operação eficaz é um dos pilares que sustenta a conectividade da rede.

Para elevar ainda mais a eficiência da segmentação da rede, introduziu-se o conceito estratégico das VLANs. Dentro desse cenário, implementou-se a VLAN 10, uma subdivisão virtual que isola o servidor Zabbix e o utilizador final, separando-os de outros componentes da rede. Essa abordagem não apenas aprimora a segurança, impedindo a comunicação não autorizada entre diferentes partes da rede, mas também aumenta a eficiência ao direcionar fluxos de tráfego específicos para as suas respetivas áreas.

A implementação da VLAN 10 demonstra a capacidade de criar segmentações lógicas dentro da infraestrutura física, proporcionando uma estrutura organizada que atende tanto às necessidades de segurança quanto à otimização do desempenho. Cada VLAN pode ser vista como um ambiente digital isolado, onde dispositivos relacionados e fluxos de dados específicos podem interagir livremente, sem interferência externa.

Além disso, essa configuração estratégica facilita a gestão e a monitorização da rede, uma vez que cada VLAN pode ser administrada de forma independente, reduzindo a complexidade das operações e permitindo um controle mais preciso sobre o tráfego e as comunicações. Portanto, enquanto se caminha em direção à implementação completa da nossa engenharia de redes, a configuração meticulosa do switch e a implementação estratégica das VLANs constituem passos cruciais para alcançar uma rede eficiente, segura e altamente funcional.

O FortiSwitch 108E-POE, o switch central na nossa configuração, oferece um conjunto robusto de recursos de hardware que contribuem para a eficácia geral da rede. Com esses recursos, o FortiSwitch 108E-POE se torna um componente vital na busca por uma rede altamente funcional e segura.

A configuração deste switch e a criação das VLAN's é demonstrada nos Anexo 6.

Capítulo 4

4.Configuração do servidor Zabbix e dos seus agentes

Neste capítulo, será abordado uma análise detalhada da configuração de um servidor Zabbix, mergulhando nos aspetos cruciais que sustentam seu funcionamento eficaz. Iremos explorar a seleção cuidadosa dos hosts a serem monitorizados, considerando os protocolos de monitorização como o software Zabbix-agent, SNMPv2 e SNMPv3, e compreendendo a interação de cada um na recolha precisa de dados.

Além disso, serão revelados os alicerces fundamentais do Zabbix, como os elementos de *triggers, itens e hosts*, e também a funcionalidade dos *templates*, que proporcionam agilidade na configuração em grande escala. Será aprofundado o entendimento sobre esses componentes, explorando as suas aplicações práticas na identificação de anomalias, na captura de métricas essenciais e na supervisão de ativos.

Na sequência, explora-se pela integração harmoniosa com ferramentas externas, nomeadamente o GLPi e o Grafana. Discutir-se-á a configuração desse processo de integração, enquanto se delineia os benefícios tangíveis dessa união. Examinaremos o papel do GLPi como solução de gestão de ativos e suporte, e a habilidade do Grafana em transformar dados em visualizações interativas, proporcionando ensinamentos valiosos sobre o ambiente monitorizado.

Por último, aborda-se a criação de um *script* personalizado destinado a fornecer informações sobre as tarefas rotineiras aos colaboradores. Esse script oferecerá uma visão clara e concisa das atividades regulares no Zabbix, simplificando a comunicação interna e otimizando o fluxo de trabalho.

À medida que se exploram estes tópicos, estabelece-se então uma base sólida para uma compreensão mais profunda e holística do cenário complexo da monitorização e gestão de redes, preparando para futuras melhorias e aprimoramentos

.

4.1. Zabbix

O Zabbix é um software open-source destinado á monitorização de infraestruturas de TI, aplicações, serviços e redes. Oferece uma abordagem abrangente para a monitorização, permitindo que as organizações monitorizem o desempenho, a disponibilidade e a integridade dos seus sistemas em tempo real.

O Zabbix permite monitorizar vários tipos de ativos, incluindo servidores, dispositivos de rede, aplicativos, base de dados, e muito mais. Ele oferece suporte a diferentes tipos de monitorização, como recursos de hardware, desempenho de aplicativos e disponibilidade de serviços.

Este software permite recolher dados usando vários protocolos, como SNMP, JMX, IPMI, agentes Zabbix e scripts personalizados. Esses dados podem incluir informações sobre CPU, memória, uso de disco, tráfego de rede e muito mais.

O sistema de notificação e alerta do Zabbix permite que os administradores sejam informados imediatamente sobre problemas ou condições anormais. Isso é essencial para a resolução proativa de problemas e a manutenção da disponibilidade do sistema.

O Zabbix oferece a capacidade de criar painéis personalizados e relatórios detalhados com base nos dados de monitorização. Isso ajuda a visualizar tendências, identificar pontos de estrangulamento e analisar o desempenho ao longo do tempo.

O Zabbix é escalável e pode ser usado para monitorizar desde ambiente pequenos até grandes infraestruturas. Ele também suporta configurações distribuídas para atender a diferentes requisitos da infraestrutura.

Os utilizadores podem personalizar o Zabbix de acordo com as suas necessidades específicas. Além disso, a automação é suportada por meio de triggers, ações, *scripts* personalizados que podem ser executados em resposta a eventos específicos.

O Zabbix possui uma comunidade ativa de desenvolvedores e utilizadores, o que significa que há uma ampla gama de recursos, documentação e suporte disponíveis.

O Zabbix é composto por diversos componentes que trabalham juntos para fornecer a monitorização. Alguns dos principais componentes incluem o servidor Zabbix, responsável pela recolha, processamento e armazenamento de dados de monitorização; os agentes Zabbix, que recolhem informações locais e as enviam para o servidor para análise; a base de dados, onde são armazenadas informações de configuração e histórico de dados; e a interface web do Zabbix, que permite aos utilizadores visualizarem dados de monitorização, criarem painéis personalizados, configurarem alertas e gerirem a configuração do sistema.

A Figura 7 [3], representa graficamente a arquitetura de como funciona o Zabbix, ou seja, a sua arquitetura.

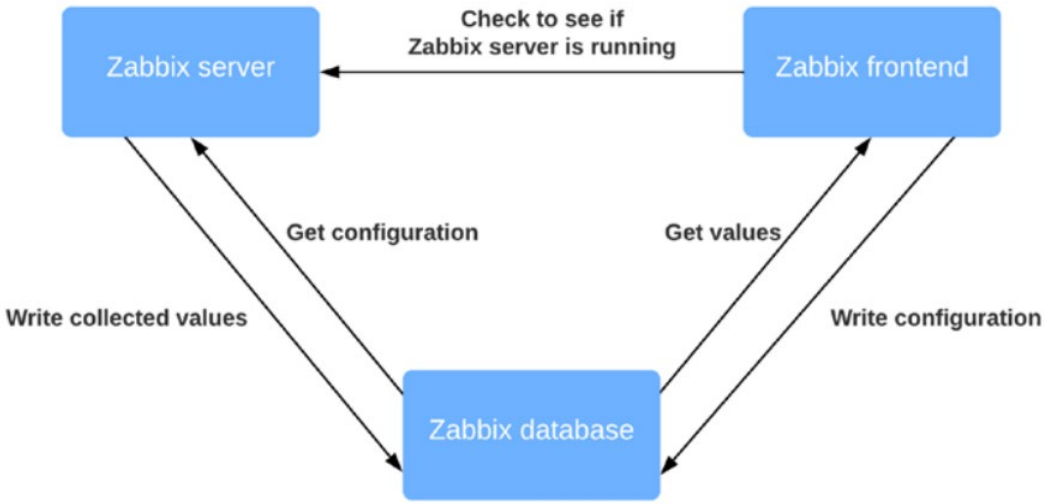


Figura 7 - Arquitetura do Zabbix

A Figura 8 apresenta o modelo ER da base de dados do Zabbix, um mapa que delineia a complexa relação entre as várias entidades e os seus atributos dentro deste software. É fundamental destacar que a estrutura desta base de dados não é moldada ou personalizada pelo utilizador final. Em vez disso, é predefinida e estabelecida durante a instalação do software, indicando um sistema que já foi meticulosamente planeado e estruturado pelos seus desenvolvedores.

Esta decisão de design por parte dos criadores do Zabbix revela uma intenção clara: garantir que a integridade, eficácia e robustez da base de dados se mantêm constantes, independentemente das especificidades do ambiente de instalação. A complexidade da base de dados, é um testemunho da profundidade e amplitude das funcionalidades oferecidas pelo Zabbix.

Por trás de cada entidade e relação na base e dados, existe um conjunto de funcionalidades e recursos do software que servem necessidades especificas de monitorização e análise. Portanto, ao analisar a intrincada rede de relações no modelo ER podemos perceber a vastidão de capacidades do sistema Zabbix. Esta complexa arquitetura não só sustenta as operações diárias do software, mas também sublinha o seu caráter avançado e a sua capacidade de se adaptar a cenários de monitorização multifacetados.

Agora que se tem uma compreensão sólida da estrutura e dos principais componentes do Zabbix, é possível avançar para a demonstração daquilo que é a instalação do software.

4.2. Instalação do Zabbix

Nesta secção irá ser abordado detalhadamente o processo de instalação do Zabbix, desde a preparação do ambiente até à configuração final.

Este processo, embora estruturado, pode variar dependendo do sistema operativo usado, dos recursos disponíveis e das preferências individuais. Ir-se-á deste modo detalhar o caso particular da instalação realizada. Este processo encontra-se documentado no site oficial do Zabbix.

4.2.1. Pré-requisitos do sistema

Antes de começar a instalação, é essencial garantir que o sistema atenda aos requisitos mínimos necessários para executar o Zabbix:

Sistema Operativo: Embora o Zabbix possa ser instalado em vários sistemas operativos, apenas pode ser instalado em sistemas UNIX. Os sistemas operativos para qual a instalação do Zabbix está disponível atualmente são: *Alma Linux, CentOS. Debian, Oracle Linux, Raspberry Pi OS, Red Hat Enterprise Linux, Rocky Linux, SUSE Linux Enterprise Server, Ubuntu, Ubuntu (arm64).* No meu caso foi usado um Sistema Operativo baseado em *Debian, Kali Linux.* Sistema operativo esse que já se encontrava operacional na minha máquina. [3]

Base de Dados: É possível usar dois sistemas de gestão de base de dados relacional de maneira que os mesmos sejam compatíveis com o software, esses sistemas de gestão são os seguintes: MySQL, PostgreSQL. O RDBMS usado foi MariaDB, que surgiu como um *fork* do MySQL após a aquisição do MySQL pela ORACLE em 2010. Desde então, MariaDB tornou-se uma das bases de dados *open-source* mais populares, com uma comunidade ativa e desenvolvimento contínuo. [3]

Assiom, o MariaDB-server torna-se uma solução viável e robusta a ser utilizada com o software Zabbix.

Servidor Web: Outro dos requisitos para a instalação do Zabbix é um servidor web. Os servidores web compatíveis com o software de acordo com a documentação oficial do mesmo são: Apache2 e Nginx. O servidor web que foi usado, já referido e caracterizado anteriormente neste relatório foi o Apache2. [3]

PHP: Trata-se de uma linguagem de programação open source amplamente utilizada, particularmente adequada para desenvolvimento web, e também de outro requisito para o funcionamento do Zabbix. Aqui estão alguns pontos chaves, de notar, sobre o PHP:

Linguagem de *Scripting* do Lado do Servidor: O PHP é executado no servidor, o que significa que a execução do código PHP é feita no lado do servidor, e apenas o resultado é enviado para o cliente ou navegador web.

Open-Source: O PHP é uma linguagem de programação de código aberto, o que significa que é gratuito para usar e também possui uma grande comunidade de desenvolvedores contribuindo para sua evolução e aprimoramento.

Plataforma Independente: Pode ser executado na maioria das plataformas, incluindo Windows, Linux, Unix, Mac OS X, etc.

Compatível com Servidores de Base de Dados: O PHP suporta muitos servidores de base de dados, incluindo MySQL, PostgreSQL, SQLite, Microsoft SQL Server e muitos outros.

Integração: PHP suporta uma ampla variedade de protocolos como POP3, IMAP e LDAP. Também pode ser facilmente integrado com outras tecnologias, como Java.

Simplicidade: Um dos pontos fortes do PHP é sua simplicidade. Com um breve conhecimento, os iniciantes podem desenvolver *sites* dinâmicos rapidamente.

Eficiente: Com *frameworks* e ferramentas adequados, o PHP pode ser tão eficaz e rápido quanto qualquer outra linguagem de programação.

Seguro: Embora todas as linguagens tenham vulnerabilidades, o PHP oferece múltiplos níveis de segurança para evitar ameaças e ataques mal-intencionados.

Flexível: Seja no desenvolvimento orientado a objetos ou procedimental, o PHP oferece uma flexibilidade considerável ao programador.

Comunidade Ativa: Dado seu status de código aberto e popularidade, o PHP tem uma comunidade muito ativa. Isso significa que há uma grande quantidade de documentação, fóruns, tutoriais e scripts prontos disponíveis. De referenciar que o PHP já se encontrava instalado na máquina na versão 8.2.7., compatível com a versão 6.4 do Zabbix a ser instalado.

O comando de terminal usado para que sejam instalados os requisitos foi o seguinte:

```
sudo apt install mariadb-server apache2
```

4.2.2. Instalação do software

Uma vez cumpridos os requisitos do sistema, o processo de instalação do Zabbix pode ser iniciado. Este guia proporcionará um passo a passo detalhado daquilo que foi feito de maneira a instalar o Zabbix 6.4 em Kali Linux. Será abordado desde a preparação do repositório até a inicialização do servidor e do agente.

Instalar o Repositório Zabbix:

```
sudo wget https://repo.zabbix.com/zabbix/6.4/debian/pool/main/z/zabbix-release/zabbix-release_6.4-1+debian12_all.deb
```

```
sudo dpkg -i zabbix-release_6.4-1+debian12_all.deb
```

```
sudo apt update
```

```
Instalação do servidor, frontend e o agente Zabbix:
sudo apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-sql-scripts zabbix-agent
```

```
Criar base de dados operacional:
```

```
sudo mysql -uroot -p
```

```
password
```

```
mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
```

```
mysql> create user zabbix@localhost identified by 'password';
```

```
mysql> grant all privileges on zabbix.* to zabbix@localhost;
```

```
mysql> set global log_bin_trust_function_creators = 1; mysql> quit;
```

```
Importar o esquema inicial e os dados no servidor Zabbix:
```

```
sudo zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4 -uzabbix -p zabbix
```

Desativar log_bin_trust_function_creators após importar o esquema da base de dados:

Nota: O parâmetro 'log_bin_trust_function_creators' permite a criação de funções armazenadas que podem ser consideradas inseguras, mas são necessárias durante o processo de importação. No entanto, por razões de segurança, é aconselhável desativá-lo após a importação. [4]

```
sudo mysql -uroot -p
```

```
password
```

```
mysql> set global log_bin_trust_function_creators = 0; mysql> quit;
```

De seguida será demonstrado uma seção do ficheiro de configuração do servidor Zabbix. Os comentários (linhas iniciadas com #) são parte integrante do ficheiro e fornecem informações adicionais sobre cada configuração. As linhas relevantes que requerem edição estão claramente destacadas:

```
sudo nano /etc/zabbix/zabbix_server.conf
DBName = zabbix
```

```
DBUser = zabbix
```

```
DBPassword=password
```

De referir, que aquilo aqui apresentado trata-se apenas de um "excerto" daquilo que é o ficheiro de configuração do servidor Zabbix, mas devido à sua extensividade coloquei apenas aquilo que é necessário alterar no mesmo durante o processo de instalação.

Iniciar o servidor Zabbix e os processos do agente:

```
sudo systemctl restart zabbix-server zabbix-agent apache2
```

```
sudo systemctl enable zabbix-server zabbix-agent apache2
```

4.2.3. Configuração do Zabbix frontend

Após o detalhado procedimento mostrado anteriormente, a próxima fase crucial é a configuração do *frontend* do *software*. Isso é feito acedendo o endereço http://localhost/zabbix através de um navegador web. Ao fazer isso, é apresentada a interface gráfica do software Zabbix, que é intuitiva e projetada para facilitar a instalação e configuração.

As ilustrações subsequentes, representadas pelas Figuras 8 a 12 elucidam detalhadamente o processo de instalação da *interface frontend* do Zabbix.

A Figura 9, destaca o primeiro passo da configuração do Zabbix. Aqui os utilizadores iniciam o processo, definindo a linguagem que desejam para a interface.

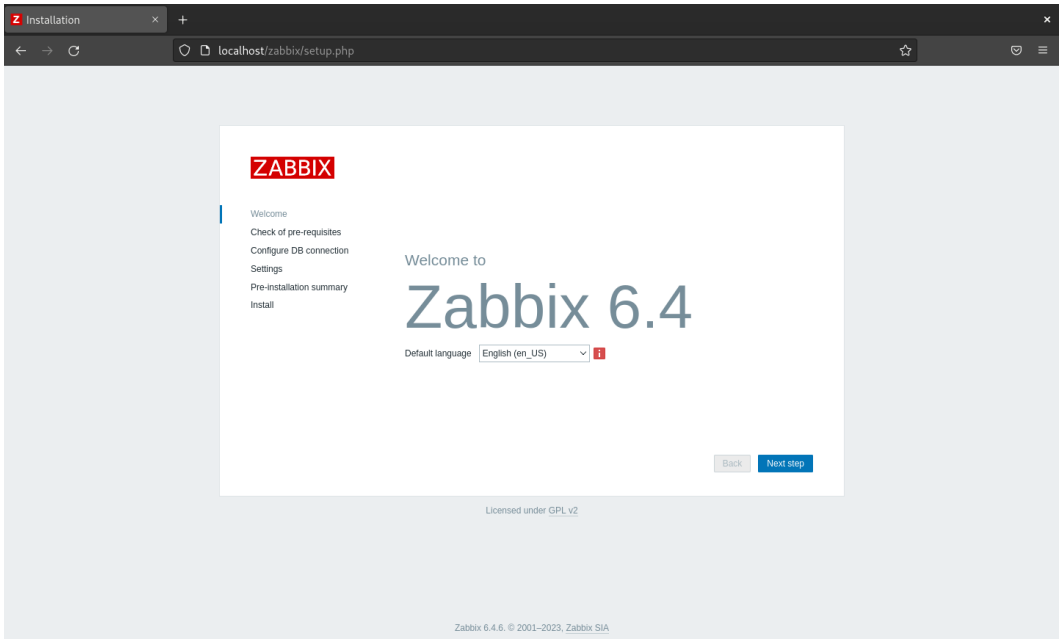


Figura 8 - Passo inicial da configuração do servidor Zabbix

A Figura 10, mostra a lista de pré-requisitos. Assegurar que todos os elementos necessários estejam presentes e corretamente configurados é fundamental para uma instalação bem-sucedida.

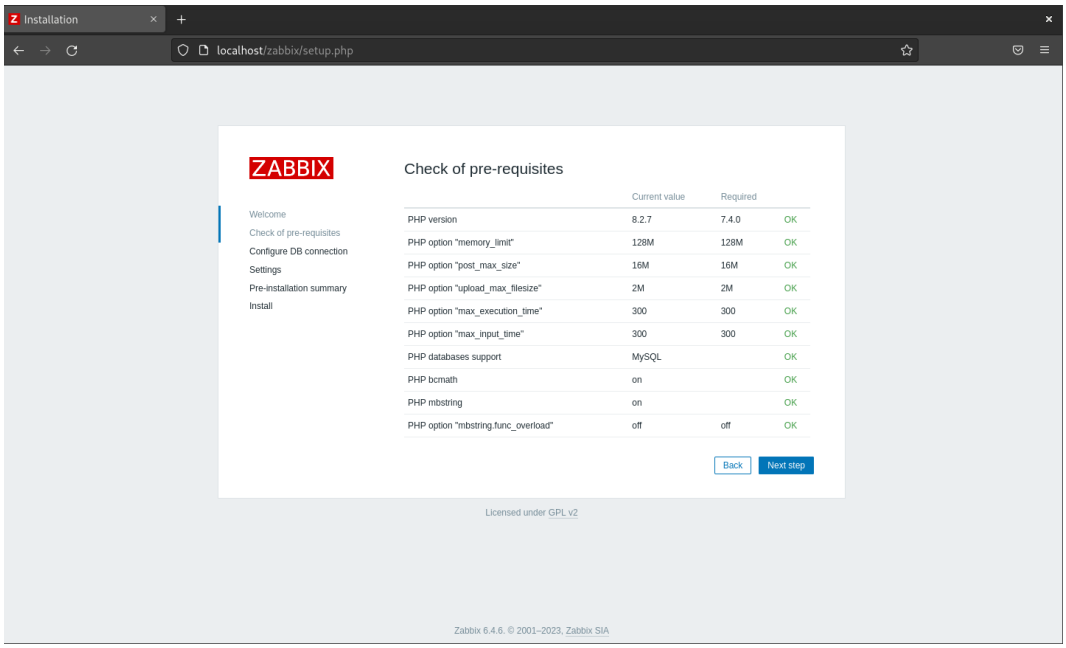


Figura 9 - Pré requisitos

A Figura 11 ilustra graficamente uma etapa vital, a conexão com a BD. Esta figura mostra como estabelecer essa conexão, garantindo que o Zabbix possa interagir eficientemente com a BD.

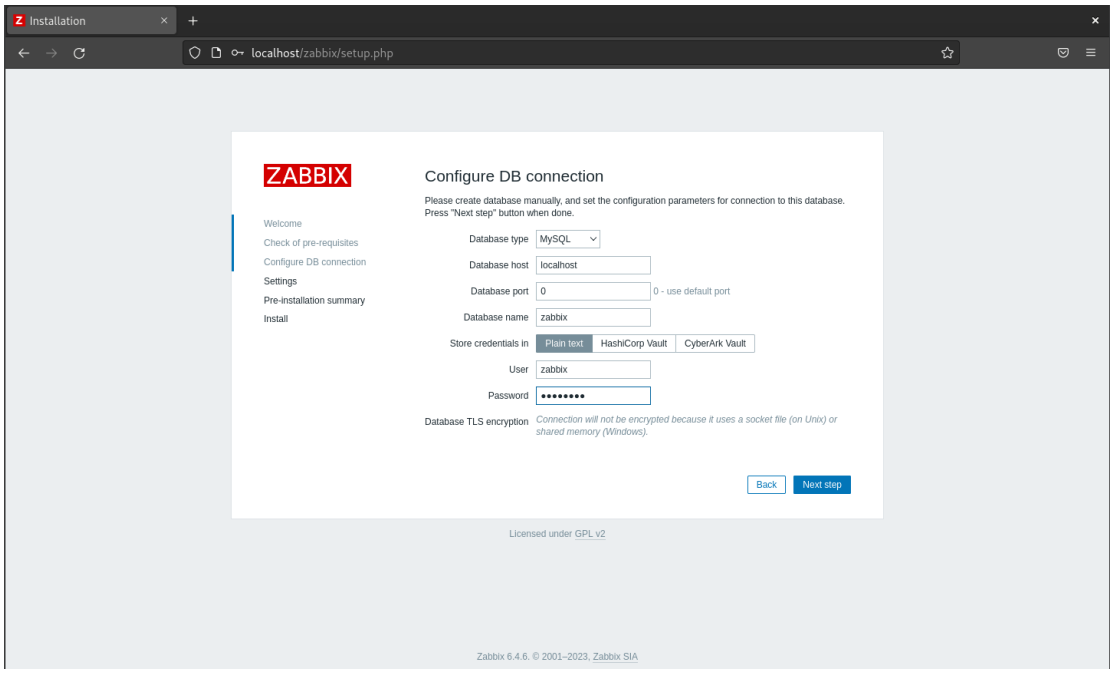


Figura 10 - Configuração da conexão com a BD

A Figura 12 ilustra as definições do servidor são ajustadas nesta etapa. Dando o nome desejado ao servidor Zabbix, escolhendo o horário fuso horário em que a máquina se encontra, e o tema desejado de acordo com as preferências do utilizador.

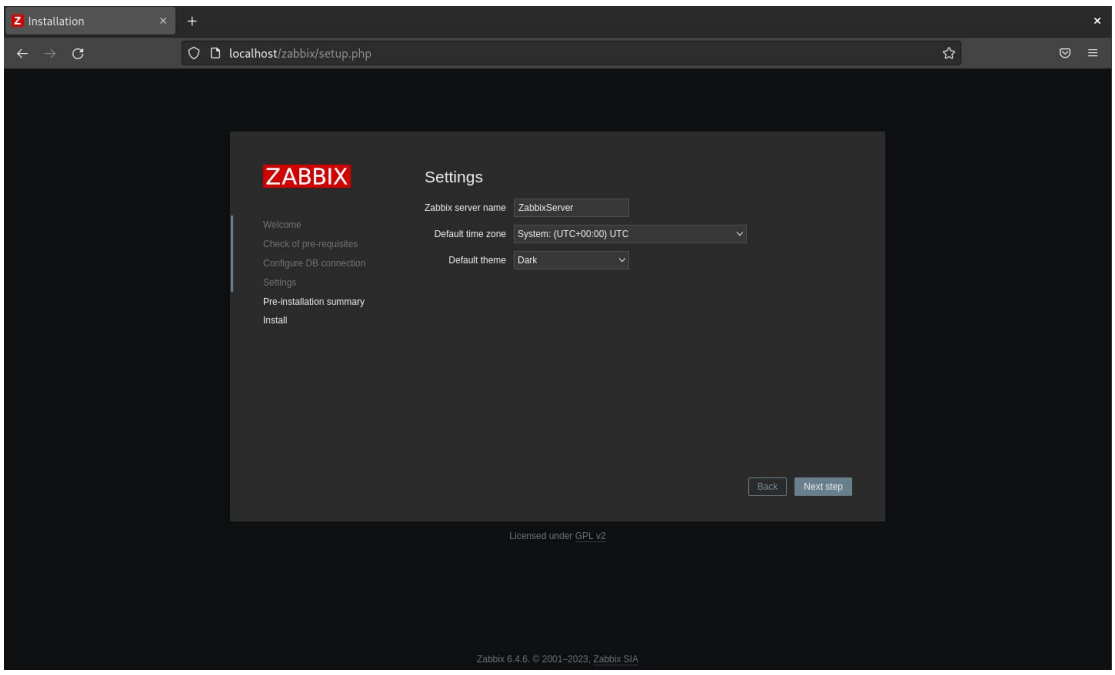


Figura 11 - Definições do servidor

Antes de concluir a instalação é apresentado um sumário, apresentado na Figura 13. Esta é uma oportunidade para os utilizadores reverem as escolhas e confirmar que tudo se encontra conforme desejado.

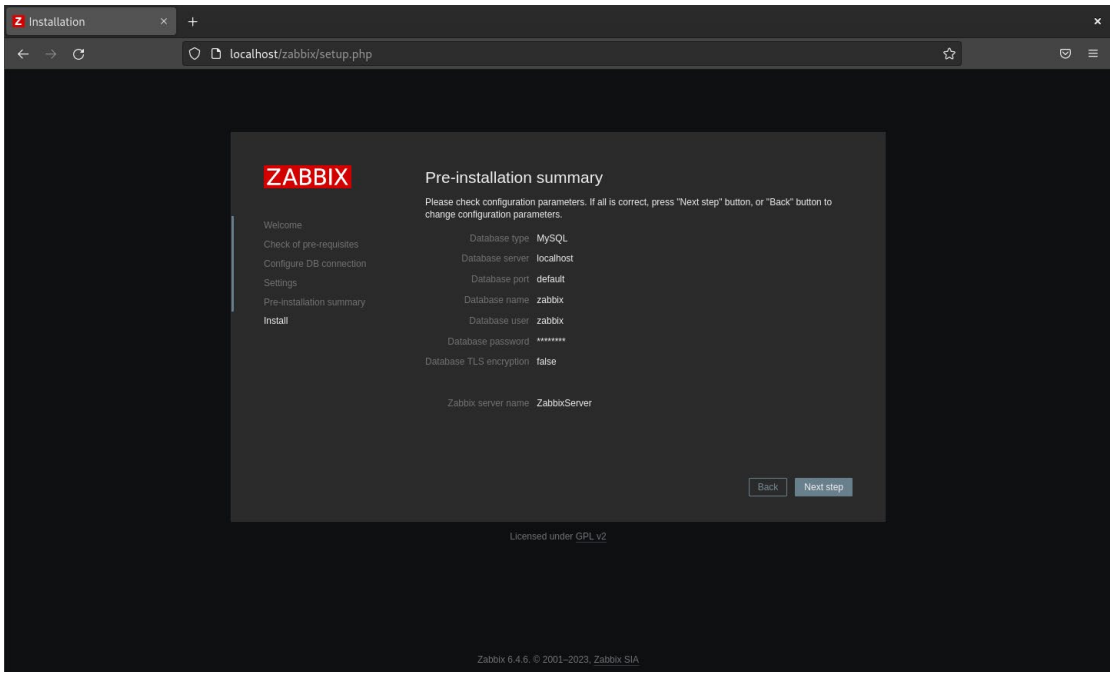


Figura 12 - Sumário pré-instalação

Após a instalação, a página de login do Zabbix frontend é exibida, ilustrada na Figura 13.

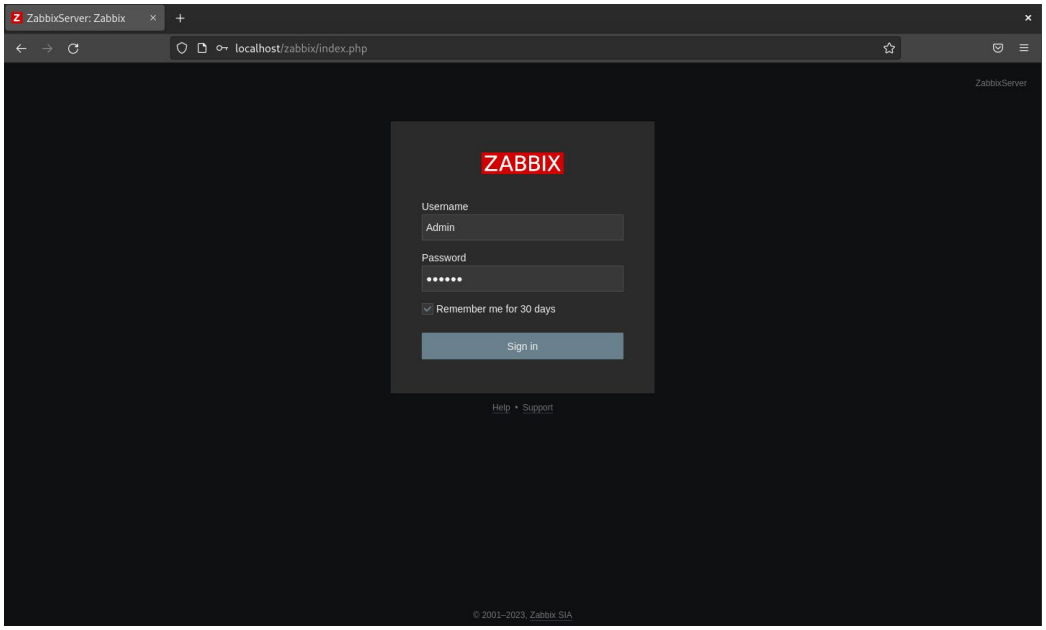


Figura 13 - Página login do Zabbix frontend

Este é o portal para a gestão e monitorização do sistema. É importante mencionar que, por padrão, as credenciais apresentadas nesta figura são predefinidas. [4]

4.3. Configuração do Zabbix

Nesta secção será explorado o processo de configuração do Zabbix e dos seus componentes essenciais. De realçar que a configuração pode variar significativamente dependendo da estrutura da rede em questão, da quantidade de recursos presentes, dos sistemas operativos utilizados e dos serviços específicos a serem monitorizados.

4.3.1. Zabbix frontend

Após a conclusão da instalação, tanto do software quanto da interface de utilizador, é apresentado o painel preestabelecido. Especificamente, este é o que o Zabbix denomina como "vista global", providenciando uma perspetiva abrangente das operações em curso. Embora seja possível personalizar este painel ou desenvolver outros conforme os requisitos específicos, é aconselhável familiarizar-se com a configuração preestabelecida antes de proceder a quaisquer modificações.

A Figura 14, ilustra graficamente o referido painel preestabelecido.

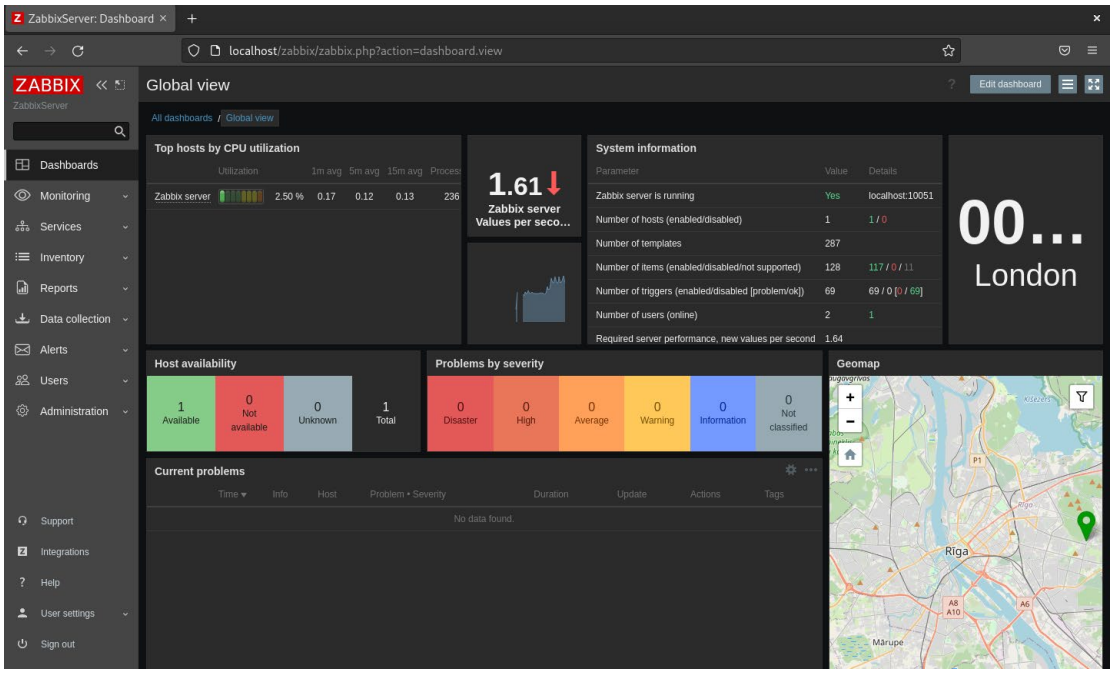
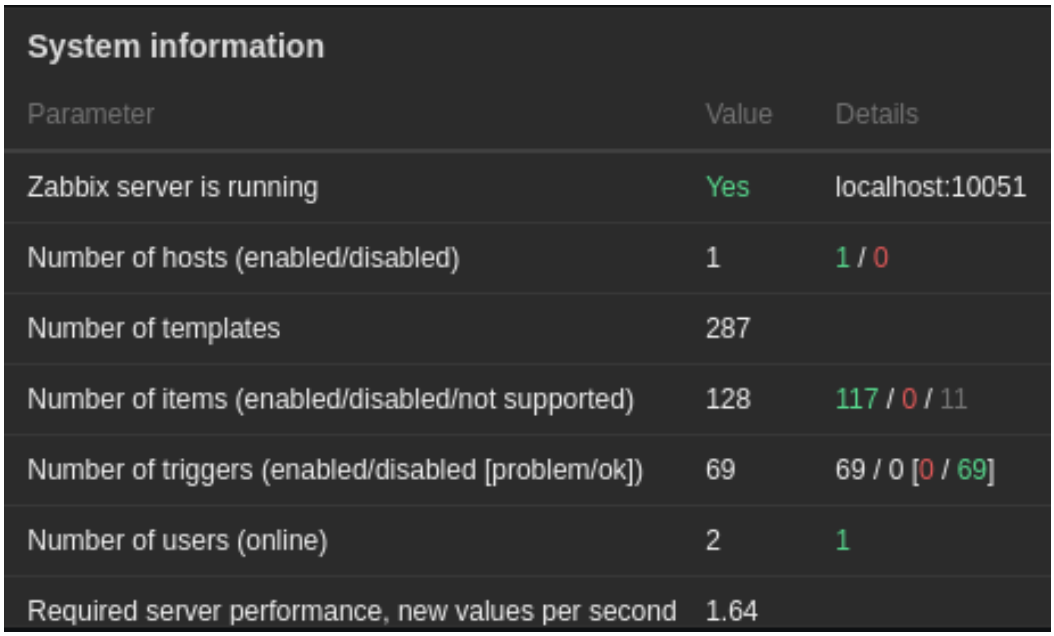


Figura 14 - Dashboard do Zabbix

Será abordada a análise dos *widgets* presentes no painel preestabelecido, iniciando pelo *widget* de informação do sistema ilustrado na Figura 15.



System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (enabled/disabled)	1	1 / 0
Number of templates	287	
Number of items (enabled/disabled/not supported)	128	117 / 0 / 11
Number of triggers (enabled/disabled [problem/ok])	69	69 / 0 [0 / 69]
Number of users (online)	2	1
Required server performance, new values per second	1.64	

Figura 15 - Widget sistema de informação

Este *widget* fornece métricas cruciais relacionadas ao servidor Zabbix, entre as quais:

Estado do Servidor Zabbix: Reflete a operacionalidade do *backend* do servidor Zabbix e sua localização de execução, atualmente em localhost:10051.

Número de Hosts e Templates: Representa a contabilização de *hosts* ativos, inativos e a totalidade de *templates*.

Número de Itens: Cataloga os itens do servidor, identificando-os entre ativos, inativos e aqueles não suportados.

Número de Triggers: Quantifica os triggers, categorizando-os com base no seu estado operacional.

Número de Utilizadores: Enumera a totalidade de utilizadores, bem como os que se encontram online no momento.

Desempenho Requerido do Servidor, *New values per second* (NVPS): Estima os valores rececionados pelo servidor Zabbix, um indicador chave para a escalabilidade futura. [4]

A Figura 16, ilustra o seguinte *widget* em discussão:

Figura 16 - Widget disponibilidade de hosts e problemas por severidade

O *widget* 'Disponibilidade de *Host*' proporciona uma síntese concisa, evidenciando toda a informação essencial sobre o estado de disponibilidade dos *hosts* monitorizados. Este *widget* indica se o *host* está Disponível, Não Disponível ou em Estado Desconhecido, fornecendo uma perspetiva unificada da disponibilidade de todos os *hosts* potencialmente monitorizados pelo servidor Zabbix.

Em complemento, o *widget* 'Problemas por severidade' descreve quantos *hosts* possuem, no momento, um *trigger* em determinado estado. O Zabbix classifica as situações em várias severidades: Catástrofe, Elevado, Médio, Alerta, Informação e Não Classificado.

Na Figura 17, é apresentado um *widget* de relevante utilidade perante a comunidade de utilizadores do Zabbix.

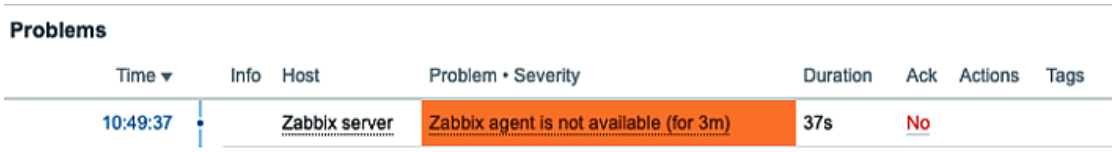


Figura 17 - Widget problemas

Esta ferramenta permite-nos visualizar os problemas atuais e, com uma configuração adequada dos triggers, extrair informações de grande valor. Uma breve síntese sobre a quantidade de *hosts* é disponibilizada, mas a página de Problemas complementa com detalhes adicionais:

Hora: Momento em que o problema foi primeiro detetado pelo servidor Zabbix.

Informações: Dados sobre o evento, incluindo os estados de Fechamento Manual e Suprimido.

***Host*:** Identifica em que *host* o problema ocorreu.

Problema/Severidade: Descreve o problema e a sua gravidade. A severidade é representada por uma cor, neste caso, laranja, que indica médio.

Duração: Indica o período em que o problema persistiu.

Reconhecimento: Informa se o problema foi reconhecido.

Ações: Medidas adotadas após a ocorrência do problema, por exemplo, a execução de um script personalizado. Ao posicionar o cursor sobre qualquer ação, são fornecidos detalhes sobre todas as medidas tomadas para esse problema específico.

Etiquetas: Mostra as etiquetas associadas ao problema. [4]

O *widget* de Problemas é extremamente útil. Existem diferentes variantes deste *widget* disponíveis e, como mencionado anteriormente, é completamente personalizável, dependendo de como desejamos que os problemas sejam exibidos.

Essa possível personalização é demonstrada na figura 18, mostrando a página de edição deste *widget*.

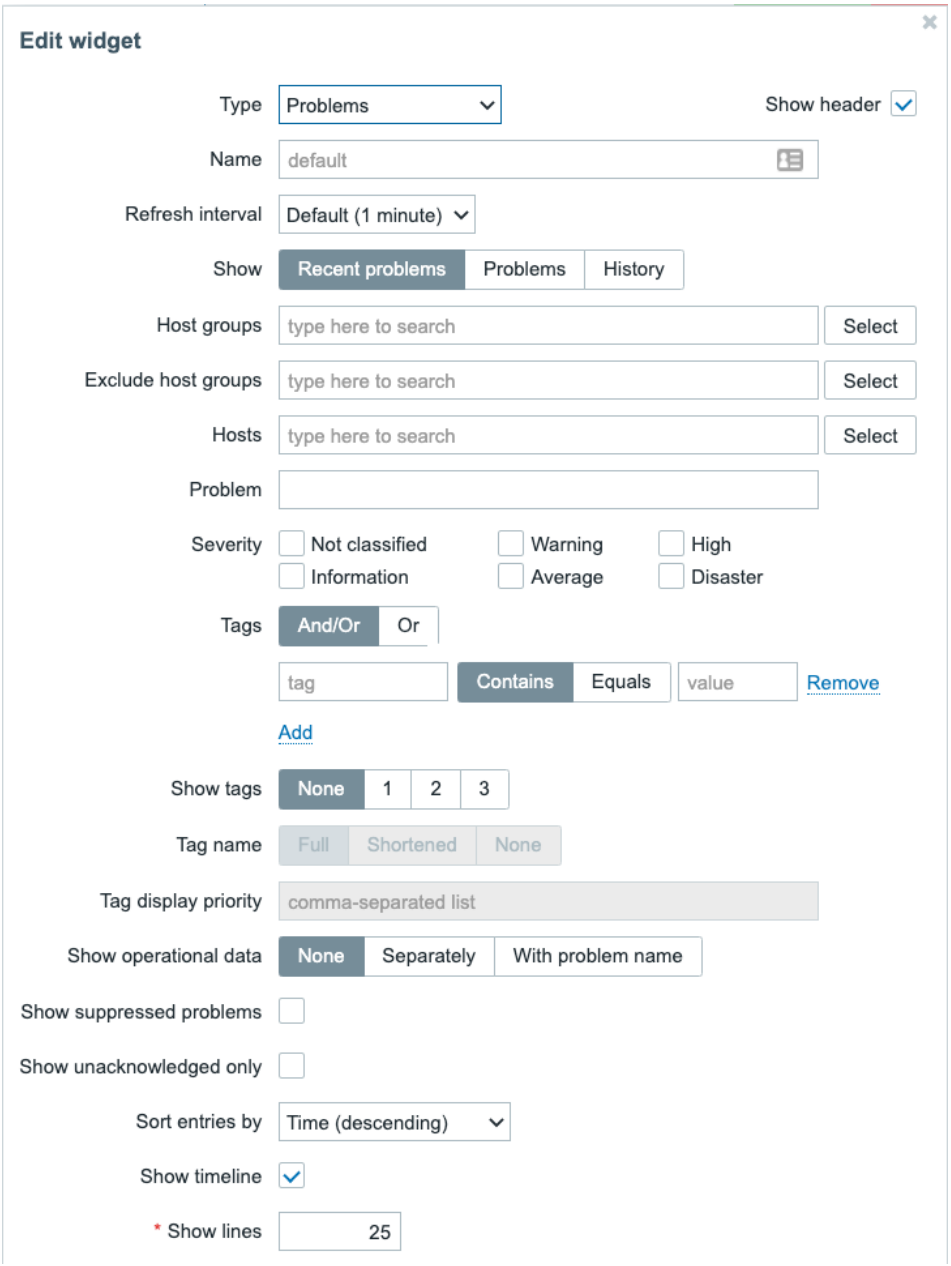
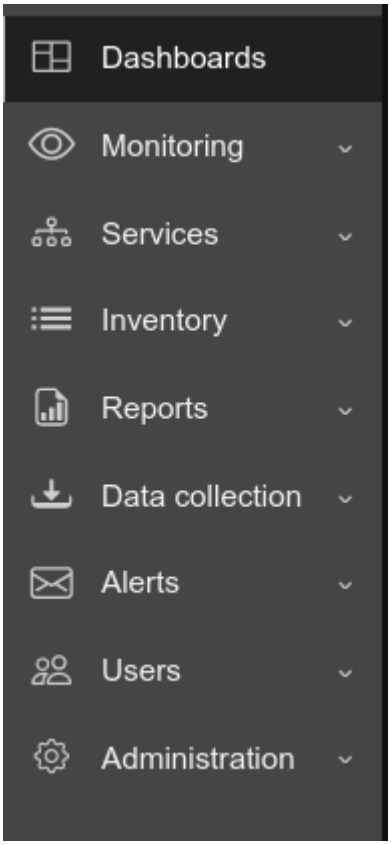


Figura 18 - Página edição widget 'Problemas'

O Zabbix oferece uma interface robusta com várias categorias, permitindo que os utilizadores gerem e monitorizem eficazmente os seus sistemas. A Figura 19 ilustra o menu principal e as categorias possíveis de escolher para o utilizador:

Monitorização: Permite ao utilizador visualizar e analisar os dados recolhidos, desde um vislumbre geral no *Dashboard* até aos detalhes específicos dos problemas e *hosts*. Também é possível visualizar dados recentes, mapas de infraestrutura e dispositivos descobertos.

Serviços: Introduzido no Zabbix 6, esta categoria foca-se na monitorização de serviços empresariais, incluindo a configuração de SLAs e ações relacionadas a serviços.



Inventário: Um recurso adicional que oferece a capacidade de gerir informações relacionadas ao inventário, como versões de software, e números de série de hosts.

Relatórios: Oferece uma variedade de relatórios, desde informações do sistema até auditorias, ajudando os utilizadores a ter uma visão clara e estruturada do ambiente monitorizado.

Coleção de dados: *O núcleo da personalização, onde se definido tudo que será monitorizado e relatado. Aqui os utilizadores podem configurar hosts, templates e seus grupos, regras de descobrimento, correlação de eventos e manutenção.*

Figura 19 - Menu do Zabbix

Alertas: Serve como um *hub* centralizado para gerir notificações e respostas a diferentes eventos e situações. Além disso, os administradores podem configurar *Media Types*, que definem os canais através dos quais as notificações serem enviadas. Dentro desta seção também existe a possibilidade de ampliar as funcionalidades do Zabbix, incorporando scripts personalizados que podem ser executados em resposta a eventos específicos, oferecendo assim uma camada adicional de automação e personalização.

Utilizadores: Este é o coração administrativo do Zabbix. Nesta secção, os administradores podem definir utilizadores, estabelecer permissões, gerenciar autenticação e definir muitos outros parâmetros que moldam a maneira como o Zabbix opera e como os utilizadores interagem com o mesmo. Desde a gestão de *proxies* até à implementação de expressões regulares para filtrar dados, esta área é essencial para manter o Zabbix otimizado. [4]

Dentro deste panorama global do *frontend* do Zabbix, torna-se evidente a profundidade e a flexibilidade que a plataforma oferece. Cada seção do menu principal desempenha um papel crucial, permitindo que empresas de todos os tamanhos e complexidades monitorizem os seus sistemas com eficácia, eficiência e precisão.

Para uma adoção bem-sucedida do Zabbix, é fundamental entender a finalidade e funcionalidades de cada uma dessas seções.

4.3.2. Configuração dos agentes do Zabbix

No atual panorama tecnológico, a crescente exigência por sistemas sempre disponíveis e eficientes realça a importância de possuir ferramentas de monitorização competentes. No entanto, a eficácia de uma ferramenta não se mede apenas pelas suas funcionalidades, mas sim pela mestria com que é configurada e operada.

Imagine-se uma vasta biblioteca, repleta de livros valiosos, mas sem um sistema de organização. A busca por um único livro poderia transformar-se num desafio interminável de percorrer incontáveis corredores. Esta é uma analogia que se aplica ao Zabbix, onde os *hosts* são comparáveis a estes livros. Sem uma configuração adequada, corre-se o risco de não só perder informação vital, mas também ser sobrecarregado por dados irrelevantes.

No domínio da monitorização de sistemas e redes, garantir uma comunicação fluida entre agentes e servidor é imperativo para uma recolha de dados precisa e oportuna. No Zabbix, esta comunicação é assegurada através de vários protocolos, cada qual com as suas nuances e vantagens. Entre eles, destaca-se o protocolo SNMP, um padrão da internet desenhado para a gestão e monitorização de dispositivos. Atuando na camada do modelo *Open Systems Interconnection* (OSI), este protocolo estabelece uma dinâmica de pedido e resposta: o servidor requisita e o agente responde. Adicionalmente, existe uma funcionalidade de notificação, o *trap*, permitindo ao agente alertar o servidor sobre ocorrências específicas, como se observa na Figura 20 [4].

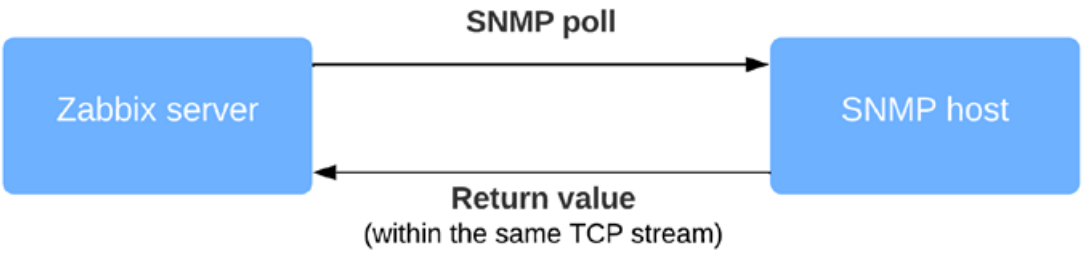


Figura 20 - Comunicação entre servidor Zabbix e host SNMP

Para este mecanismo de comunicação operar de forma eficaz, o conceito de MIB é crucial. Trata-se de um conjunto de informações organizado hierarquicamente, utilizado pelos dispositivos SNMP. Cada objeto dentro de uma MIB possui um identificador único: o OID. Estas sequências exclusivas representam itens específicos na estrutura MIB, com cada OID vinculando-se a uma informação distinta, passível de consulta ou configuração via SNMP. Como ilustração, um OID pode denotar a utilização do CPU num dispositivo ou o consumo de memória. Relativamente a este protocolo, as versões mais comuns em integração com o Zabbix são o SNMPv2 e o SNMPv3, sendo relevante destacar as diferenças entre os mesmos:

SNMPv2 (Simple Network Management Protocol versão 2):

Lançada como uma evolução do SNMP original, esta versão trouxe consigo melhorias significativas tais como:

Operações de Protocolo Aprimoradas: O SNMPv2 introduziu novas operações como o GetBulk, que permite a um gestor solicitar grandes volumes de informação numa única solicitação. Isto melhorou a eficiência da coleta de dados ao reduzir a quantidade de mensagens trocadas entre o gestor e o agente.

Melhor Gestão de Erros: O SNMPv2 introduziu códigos de erro mais detalhados, que permitiam identificar com maior precisão a razão do fracasso de uma operação, facilitando a resolução de problemas.

Formato de Mensagem Aprimorado: O SNMPv2 modificou o formato das mensagens para torná-las mais consistentes e eficientes, reduzindo assim a quantidade de largura de banda necessária para a troca de mensagens.

SetRequests Mais Rígidos: Em SNMPv1, quando um gestor enviava um SetRequest com múltiplas variáveis e uma delas falhava, todo o pedido era rejeitado. SNMPv2 aprimorou isso, permitindo que as variáveis válidas fossem configuradas mesmo se outras falhassem no mesmo pedido.

Contadores de 64 bits: O SNMPv2 introduziu contadores de 64 bits, permitindo a monitorização de redes mais rápidas sem o risco de que os contadores ficassem sobrecarregados rapidamente.

Segurança: Embora o SNMPv2 tenha introduzido mecanismos de segurança como comunidades de escrita e leitura mais distintas, ainda era criticado por sua falta de medidas de segurança robustas, o que levou ao desenvolvimento do SNMPv3.

Apesar desta lacuna de segurança, o SNMPv2 é amplamente reconhecido e adotado em vários dispositivos e sistemas devido à sua simplicidade operacional e eficácia na coleta de informações.

SNMPv3 (Simple Network Management Protocol versão 3):

Com o lançamento desta versão em 1998, o protocolo SNMP alcançou um novo patamar, especialmente no que toca à segurança. O SNMPv3 foi concebido para suprimir as falhas de segurança evidentes nas versões anteriores, introduzindo medidas robustas de segurança, que incluem autenticação e encriptação de dados. Ao invés da comunidade usada no SNMPv2, o SNMPv3 adota um modelo que combina nome de utilizador e senha. Adicionalmente, para fortificar ainda mais a segurança, dispõe de mecanismos opcionais de encriptação. Por ser um protocolo com um enfoque mais seguro, é naturalmente mais complexo em termos de configuração comparativamente ao SNMPv2. Contudo, é a escolha ideal para ambientes onde a segurança é primordial.

Após o entendimento abrangente do protocolo SNMP e as suas versões no ambiente Zabbix, é essencial explorar outros protocolos integrados para uma monitorização completa e eficaz. O Zabbix, com sua flexibilidade, permite que os administradores aproveitem uma variedade de métodos para obter insights sobre a saúde e o desempenho dos seus sistemas.

Zabbix Agent:

Uma das principais ferramentas à disposição dos administradores de rede é o próprio agente do Zabbix. O Zabbix Agent é um programa instalado em sistemas que precisam ser monitorizados. O agente recolhe dados operacionais e envia-os ao servidor Zabbix para posterior análise. Este método de recolha de dados é considerado um dos mais precisos, pois o agente é desenvolvido especificamente para se integrar ao Zabbix. Sendo um agente ativo, ele pode tanto enviar dados ao servidor quando solicitado (modo passivo) quanto por iniciativa própria (modo ativo). [4]

Agente Passivo:

O agente passivo recolhe dados do *host* através do agente Zabbix. Quando um item no nosso *host* atinge o seu intervalo de atualização, o servidor Zabbix consulta o agente Zabbix para obter o valor atual. A Figura 21 [4] ilustra isso mesmo.

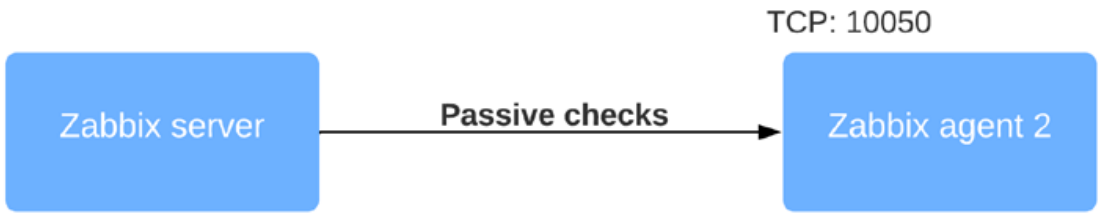


Figura 21 - Funcionamento do Zabbix Agent em modo passivo

Este tipo de agente é ideal em ambientes onde a comunicação é iniciada pelo servidor Zabbix ou pelo *proxy* Zabbix, como em situações com *firewalls* que permitem apenas tráfego de saída.

Agente Ativo:

O agente ativo envia dados do agente Zabbix para o servidor ou proxy. Quando um item no agente atinge seu intervalo de atualização, o agente envia esse valor ao servidor. Este tipo de comunicação é ilustrada na Figura 22. [4]

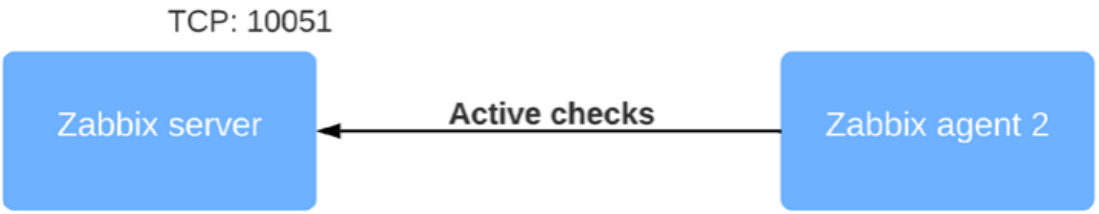


Figura 22 - Funcionamento do Zabbix Agent em modo ativo

Este agente é útil em ambientes onde *firewalls* só aceitam conexões de saída, mitigando preocupações de segurança comuns em monitorização. Além disso, o modo ativo pode ser mais eficiente, pois a carga de envio de dados é maior no lado do agente. Ao ter mais agentes do que servidores/*proxies*, distribuir essa carga é vantajoso.

Ambos os tipos de verificação podem ser usados simultaneamente, como ilustrado na Figura 23 [4], permitindo flexibilidade na configuração.

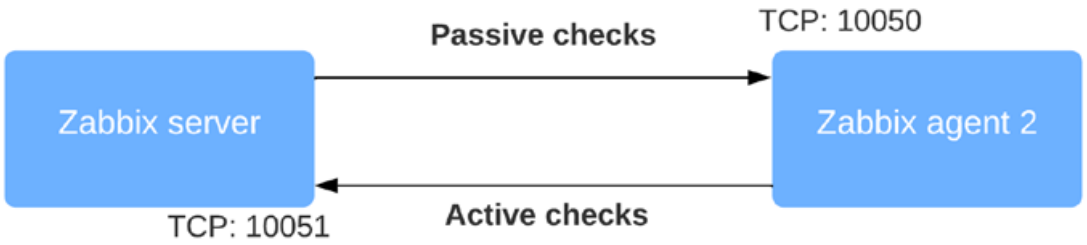


Figura 23 - Funcionamento do Zabbix Agent usando os dois modos em simultâneo

Isso é útil em cenários onde a monitorização é principalmente passiva, mas algumas tarefas, como monitorização de *logs*, requerem um agente Zabbix ativo. Assim, podemos combinar os modos para maximizar as funcionalidades do agente Zabbix.

No entanto, nem sempre é viável ou necessário instalar um agente em cada dispositivo a ser monitorizado. Em tais situações, protocolos como o ICMP se tornam instrumentos valiosos. Para uma representação detalhada das configurações de cada um dos três protocolos mencionados, consultar os respetivos Anexos 7 a 11. É importante salientar que, embora tenham sido realizadas diversas configurações de hosts, optei por apresentar em anexos apenas uma configuração representativa para cada tipo de agente, considerando as distinções entre os protocolos.

Com uma compreensão clara da importância dos protocolos e das configurações de agentes no Zabbix, é natural evoluir para a próxima etapa crucial: a criação de um host. A configuração de um host no Zabbix é o ponto central de monitorização, servindo como a interface entre as configurações e dispositivos ou sistemas que se pretende monitorizar. A criação de hosts no Zabbix, encontram-se de igual modo nos Anexos 7 a 11.

Após a configuração de toda a rede mencionada anteriormente, a página de hosts visível no Zabbix é ilustrada na Figura 24.

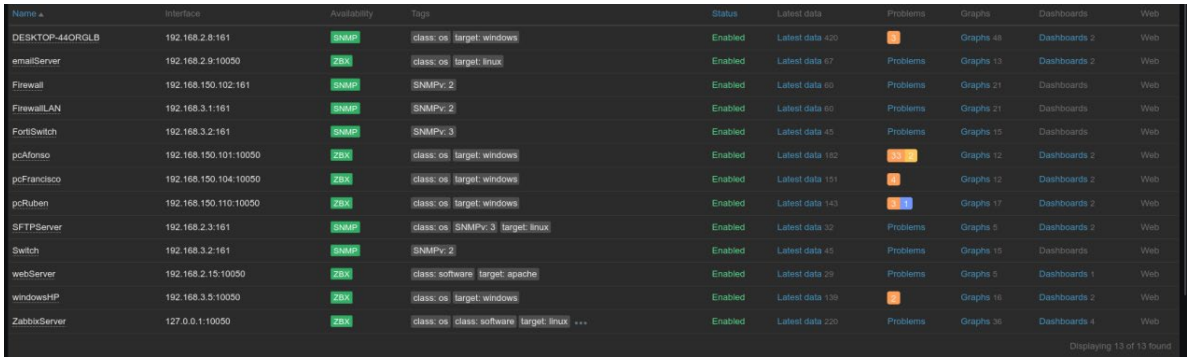


Figura 24 - Hosts monitorizados pelo Servidor

De referir que quando feita a configuração é essencial relacionar um *host* tanto a um grupo de *hosts* quanto a um *template*. Essa associação depende de variáveis como o tipo de máquina, o sistema operativo e o protocolo de comunicação utilizado. Os grupos de *hosts* no Zabbix desempenham um papel fundamental na organização, permitindo a categorização e gestão eficiente de *hosts* com características similares. Paralelamente, os *templates* são centrais para a monitorização no Zabbix. Quando associamos um *host*, que é um dispositivo a ser monitorizado, a um *template*, este último define que métricas específicas serão recolhidas. Embora o Zabbix ofereça *templates* predefinidos para uma ampla gama de dispositivos e sistemas, há casos, como com os dispositivos FortiGate e FortiSwitch, onde *templates* personalizados são necessários. Para esses dispositivos específicos, foi imperativo importar *templates* da comunidade do Zabbix, aproveitando o conhecimento coletivo e especializado disponível. A importação desses *templates* pode ser observado em Anexo 12.

4.3.3. Configuração de itens

Os *itens* já referidos anteriormente são métricas individuais que o sistema recolhe dos dispositivos monitorizados. Cada item representa uma informação especifica. A configuração adequada de itens é vital para garantir que o Zabbix recolha dados relevantes, valiosos, permitindo uma análise apropriada. Abaixo é ilustrado o processo de configuração desses mesmo *itens*:

Aceder à configuração de itens:

A figura seguinte, Figura 25, demonstra como aceder o painel principal do Zabbix e navegar até ao menu *Data Collection* seguindo pela seleção de *Hosts*.

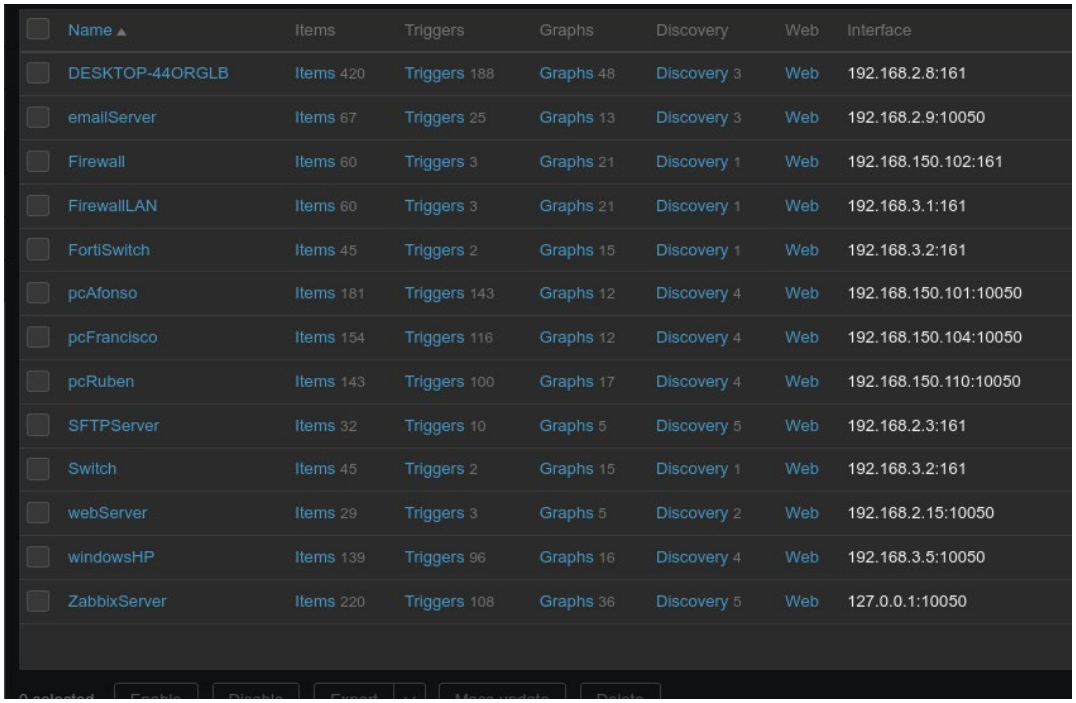


Figura 25 - Hosts

Subsequentemente, procedeu-se à seleção dos itens associados ao *host* em questão, nomeadamente, o "Zabbix Server". As Figuras 26 e 27 ilustram, respetivamente, a interface de configuração dos itens e a lista dos itens associados ao *host* "Zabbix Server".

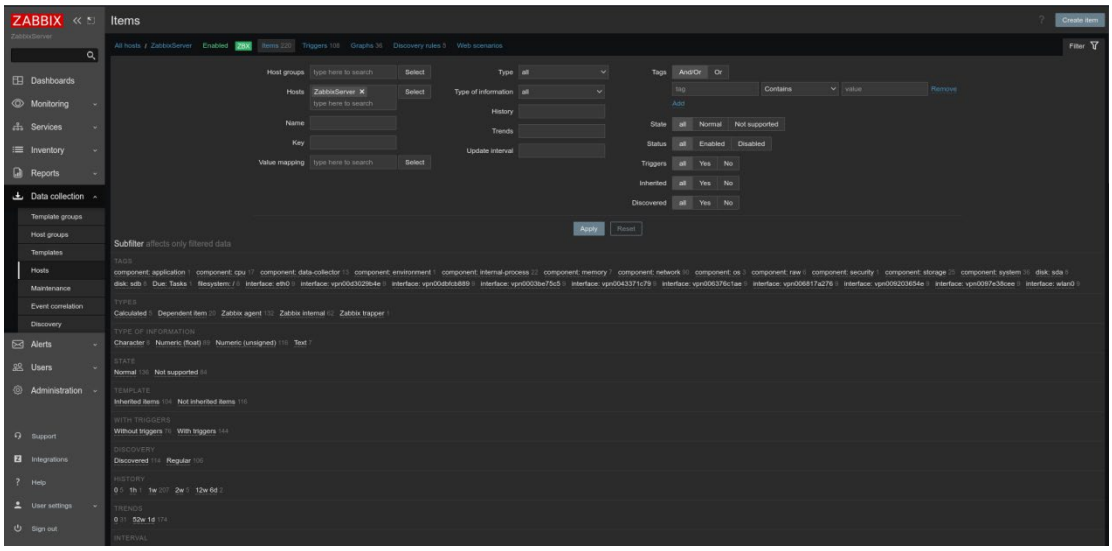


Figura 26 - Página de configuração dos itens do host «ZabbixServer»

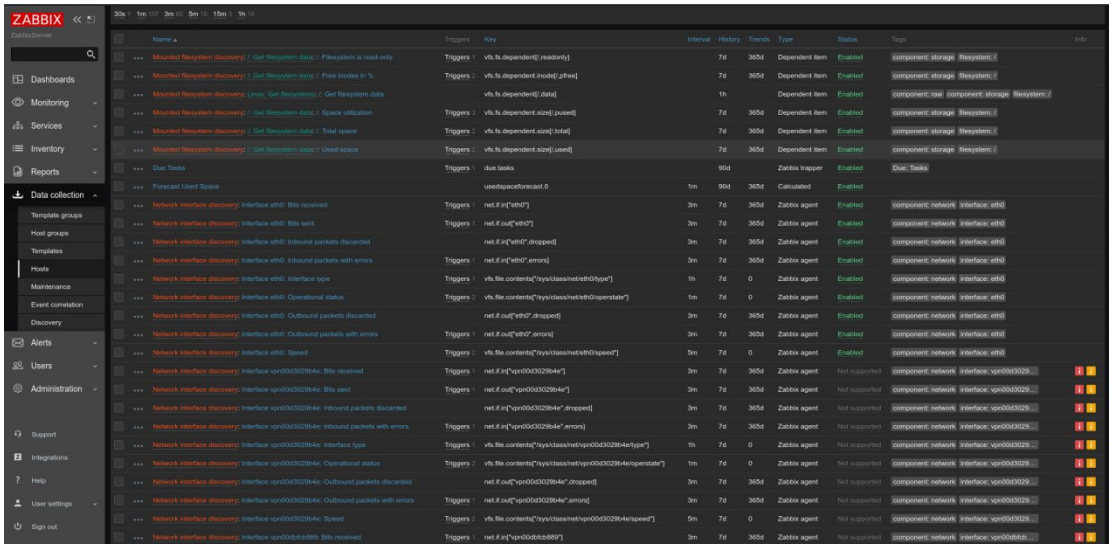


Figura 27 - Itens do host "ZabbixServer"

Ao optar pela funcionalidade "Criar item", é apresentada a interface de configuração do item. Destacam-se como atributos críticos, o nome - que por melhores práticas deve ser especificado de maneira concisa e elucidativa, tal como "Uso da CPU" - e o tipo de item, para o qual o sistema disponibiliza múltiplas alternativas, sendo as primordiais:

Zabbix Agent: Obtém métricas mediante um software agente alocado no dispositivo alvo, operando de forma passiva.

Zabbix Agent (active): Analogamente ao anterior, contudo, o agente requisita ativamente informações ao servidor Zabbix.

Simple check: Habilmente verifica serviços, como *ping* ou portas TCP, independentemente de um agente.

Zabbix Trapper: Aguarda o envio de informações de um *trapper* externo.

HTTP Test: Destinado à monitorização do rendimento e disponibilidade de aplicações web.

SNMPv1, SNMPv2c, e SNMPv3 Agent: Coletam informações de dispositivos compatíveis com o protocolo SNMP, como roteadores e *switches*.

Zabbix Internal: Supervisiona a performance e condição do servidor ou proxy Zabbix.

JMX Agent: Designado para a monitorização de aplicações Java.

IPMI Agent: Viabiliza a supervisão de hardware, como servidores, que adotam a tecnologia *Intelligent Platform Management Interface*.

SSH Agent: Implementa comandos via SSH e recolhe os resultados.

TELNET Agent: Opera similarmente ao *SSH Agent*, porém emprega o protocolo TELNET.

Calculated: Processa dados de outros itens a fim de calcular e registar um novo valor.

SNMP trap: Espera por mensagens SNMP *trap* remetidas por dispositivos.

Dependent item: Recolhe informações com base no resultado de outro item, otimizando a largura de banda e desempenho.

HTTP Agent: Executa pedidos HTTP/HTTPS visando a monitorização de serviços REST API ou outras aplicações web.

Após a determinação tipo de item, dois outros elementos cruciais que exigem a configuração de um item são a *Key* e o *Update Interval*.

***Key*:** Esta é essencialmente, um identificador único para cada item, agindo como uma espécie de «endereço» para a recolha de dados específicos, seja do sistema operativo, software ou hardware. A precisão na definição desta chave é vital para garantir a recolha correta das métricas pretendidas. Muitas vezes, o Zabbix já oferece chaves pré-

definidas para métricas comuns, mas a capacidade de personalizá-las oferece uma flexibilidade valiosa para monitorizar métricas específicas ou personalizadas.

***Update Interval*:** Esta configuração determina a frequência com que o Zabbix irá consultar o item específico para obter os dados atualizados. É expresso em segundos e dita a periodicidade das recolhas. A definição adequada deste intervalo é critica para equilibrar a necessidade de informações atualizadas com o potencial de sobrecarregar o sistema ou rede com requisições frequentes. [4]

Todos estes aspetos são componentes essenciais na configuração de um item e devem ser definidos com cuidado para assegurar uma monitorização eficaz e eficiente.

Durante o decorrer do estágio, várias configurações foram postas em prática, adaptadas às necessidades e especificidades do ambiente em questão. Para uma compreensão mais concreta e detalhada dessas implementações, pode-se consultar em Anexo 13.

4.3.4. Configuração de triggers

No contexto da monitorização de redes e sistemas, os *triggers* representam uma parte fundamental do Zabbix. Os mesmos são essencialmente condições ou regras definidas que, quando satisfeitas, indicam um problema ou uma situação que requer atenção.

Um *trigger* no Zabbix é uma expressão lógica que define uma condição anormal para os valores recolhidos de itens monitorizados. Quando a expressão se torna verdadeira, o *trigger* é ativado, indicando que um problema foi detetado. Por outro lado, quando a condição retorna ao seu estado normal e a expressão torna-se falsa, o *trigger* passa para o estado «RESOLVIDO».

A expressão de um *trigger* é baseado em funções e operações. Estas funções são aplicadas aos valores recentemente recolhidas dos itens, permitindo que os administradores definam condições.

Ao configurar um *trigger*, pode-se definir um nível de severidade, variando de «Informação» a «Desastre». Este nível de severidade é usado para categorizar e

priorizar alertas, facilitando a resposta e a alocação de recursos para solucionar problemas.

Quando um *trigger* é ativado, é possível associá-lo a ações especificas no Zabbix. Estas ações podem incluir o envio de notificações para os administradores, a execução de um *script* remoto ou até mesmo a alteração de um valor num dispositivo monitorizado.

As dependências permitem que os administradores evitem uma enorme quantidade de alertas em cascata. Se um *trigger* primário for ativado, os *triggers* dependentes não serão ativadas mesmo que as suas condições sejam atendidas.

Concluindo, os *triggers* são ferramentas poderosas para deteção proativa de problemas e automação de resposta. Uma configuração bem planeada de *triggers*, adaptada às necessidades específicas do ambiente monitorizado, é essencial para maximizar a eficiência do sistema e garantir uma resposta rápida a eventos adversos.

Para uma análise detalhada e visual das configurações de *triggers* realizadas podem ser visualizadas em Anexo 14.

4.3.5. Configuração e criação de scripts

O Zabbix dispõe de capacidades de extensão via *scripts*, permitindo uma flexibilidade notável na recolha de dados e notificações. Estes *scripts* são frequentemente empregados em duas vertentes cruciais ‘Scripts de Alerta’ e ‘Scripts de Item Externo’.

Scripts de alerta:

Estes *scripts* de alerta são rotinas executadas pelo servidor Zabbix como reação a condições específicas, muitas vezes acionadas por *triggers*. Estes ficam armazenados no diretório definido pela configuração ‘AlertScriptsPath’ no ficheiro de configuração do servidor Zabbix. Por padrão muitas instalações utilizam ‘/usr/local/share/zabbix/alertscripts’. Ao serem executados, o Zabbix fornece três argumentos padrão ao *script*: destinatário de alerta, e o assunto e a mensagem propriamente dita. Com estes dados em mão, o *script* pode realizar ações como enviar

alertas via e-mail, mensagens SMS, comunicar-se com ferramentas de *chat* corporativo ou até mesmo interagir com sistemas de gestão de incidentes.

Scripts de item externo:

Por outro lado, os *scripts* de item externo ao Zabbix oferecem a capacidade de coletar dados que o Agente Zabbix padrão não conseguiria recolher por conta própria. Estes *scripts*, situados no diretório determinado pela diretiva 'ExternalScripts', são invocados pelo Zabbix com parâmetros específicos, permitindo-lhes adquirir e retornar dados de diversas fontes e formatos. [4]

Scripts de ação:

Os *scripts* de ação são uma parte vital de estratégias de automação e autorrecuperação de monitorização. Em situações onde o serviço falha, por exemplo, o Zabbix pode ser configurado para automaticamente reiniciar esse serviço através de um script de ação. Esta abordagem não só reduz o tempo de inatividade, mas também liberta os administradores de sistemas de terem que intervir manualmente em problemas rotineiros. [4]

Quando um determinado *trigger* é ativado, uma ação associada pode ser invocada. Se essa ação for configurada para executar um *script* remoto, o Zabbix faz uso do seu agente para executar o *script* no *host* de destino.

Quando se trata da integração e gestão destes tipos de *scripts*, há várias considerações técnicas a serem observadas. A segurança é, sem dúvida, de suma importância: os *scripts* devem ser examinados quanto a vulnerabilidades para evitar potenciais explorações mal-intencionadas. Além disso, é essencial que todas as dependências requeridas pelos *scripts* estejam instaladas, e que sua execução seja otimizada para não sobrecarregar os sistemas em questão. A capacidade de registar e depurar as operações dos *scripts* não só auxilia na identificação e resolução de problemas, como também é uma prática recomendada. Finalmente, a questão das permissões não pode ser negligenciada, garantindo que os *scripts* operem de maneira corrida e segura dentro do ecossistema do Zabbix.

Sendo assim, a integração de *scripts* personalizados no Zabbix representa uma forma poderosa de expandir e personalizar as capacidades de monitorização desta plataforma.

Contudo, tal extensibilidade exige uma abordagem técnica cuidada e rigorosa para garantir a eficácia e a segurança do sistema como um todo. A implementação e criação de *scripts* personalizados e de *itens* e *triggers* referentes aos mesmos encontram-se documentadas em Anexo 15.

4.3.6. Visualização e análise de dados no Zabbix

O Zabbix sendo uma ferramenta de monitorização, um dos seus pontos fortes é a capacidade de visualizar dados de forma eficaz. A visualização de dados é essencial para a compreensão rápida dos estados e tendências dos sistemas monitorizados, permitindo que os administradores tomem decisões informadas e reajam prontamente a quaisquer incidentes.

Painel de Controlo: Como já falado anteriormente, o Zabbix possui um painel de controlo personalizável que permite visualizar, em tempo real, os principais indicadores de saúde e desempenho dos sistemas. Podem ser adicionados *widgets*, como gráficos, mapas, listas de problemas, entre outros, que dão uma visão geral do estado da infraestrutura.

Gráficos: O Zabbix gera gráficos a partir dos dados recolhidos, o que permite analisar tendências e identificar picos ou quedas normais no desempenho. Estes gráficos podem ser personalizados em termos de período de tempo, tipo de gráfico (área, linha) e muito mais.

Mapas: Os mapas no Zabbix são representações visuais de redes ou infraestruturas onde cada nó ou elemento pode representar um dispositivo, servidor ou qualquer outro item monitorizado. Estes mapas podem mostrar o estado atual dos dispositivos e também podem ser configurados para exibir informações especificas quando se passa o cursor sobre os nós.

Relatórios: No contexto de monitorização de sistemas, a geração de relatórios é uma funcionalidade importante que proporciona uma análise retrospetiva e compreensiva do comportamento dos recursos monitorizados. Dentro do ambiente do Zabbix, existem diversos tipos de relatórios, destacando-se:

Relatórios de status: Saúde geral dos sistemas.

Relatórios de disponibilidade: Tempo de atividade vs. inatividade dos *hosts*.

Relatórios de desempenho: Métrica chave, como uso de CPU.

Relatórios TOP 100: Principais consumidores de recursos.

Relatórios de auditoria: Rastreia alterações na configuração do Zabbix.

Os utilizadores podem personalizar relatórios conforme as necessidades, e estes podem ser exportados em formatos como PDF e CSV. O Zabbix ainda facilita a distribuição automatizada desses relatórios aos interessados. [4]

Para melhor compreensão e contextualização destas funcionalidades descritas, cada componente descrito será ilustrado nas figuras subsequentes, Figuras 28 a 31.

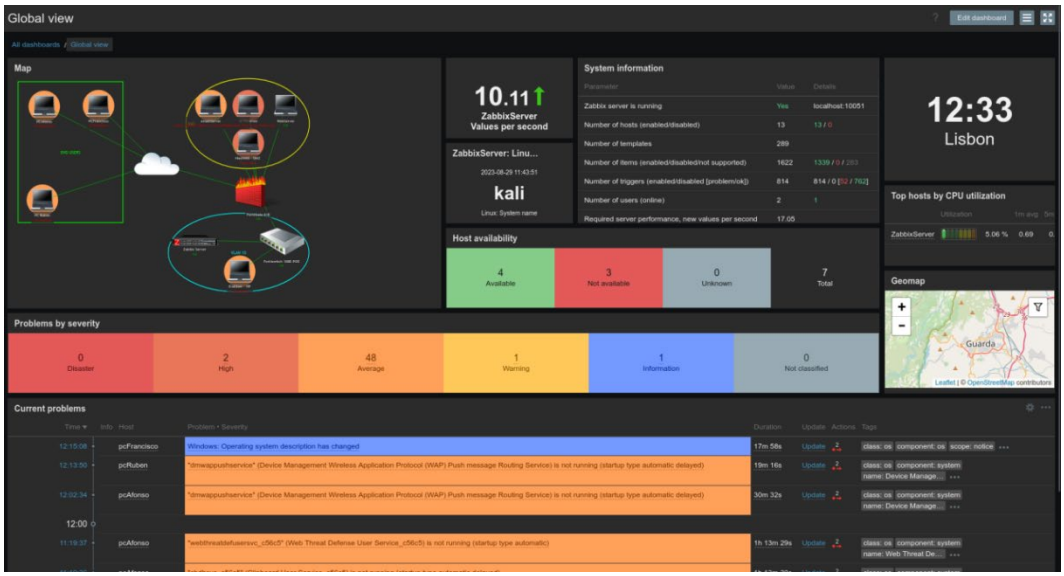


Figura 28 - Dashboard personalizado

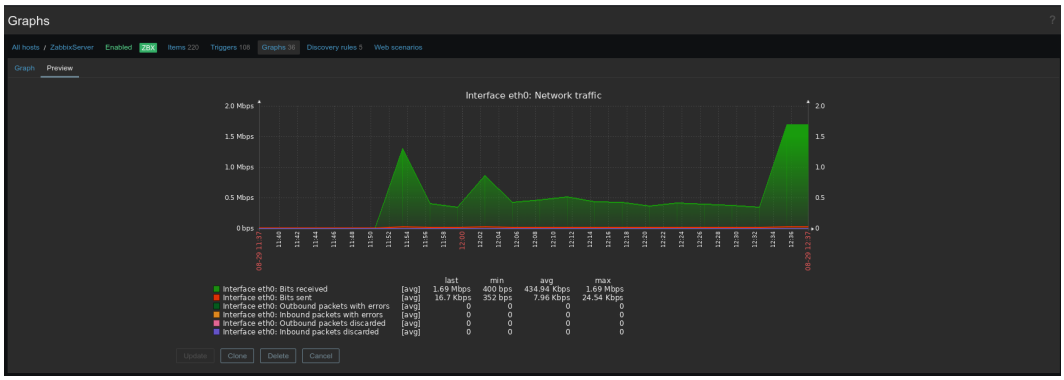


Figura 29 - Exemplo de gráfico

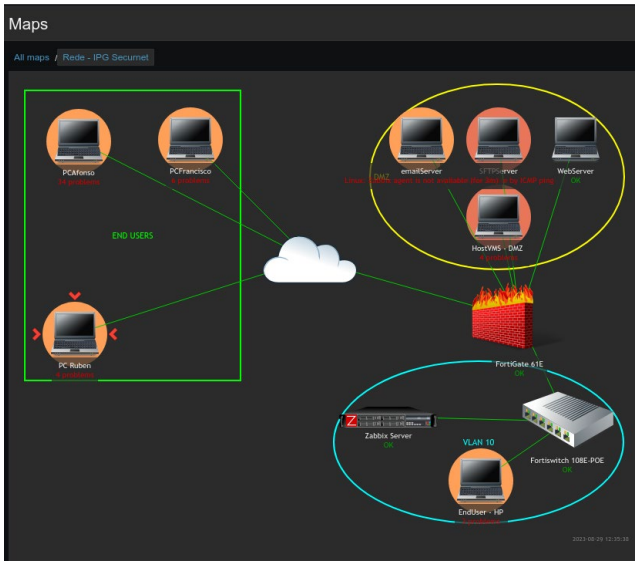


Figura 30 – Mapa da rede no Zabbix

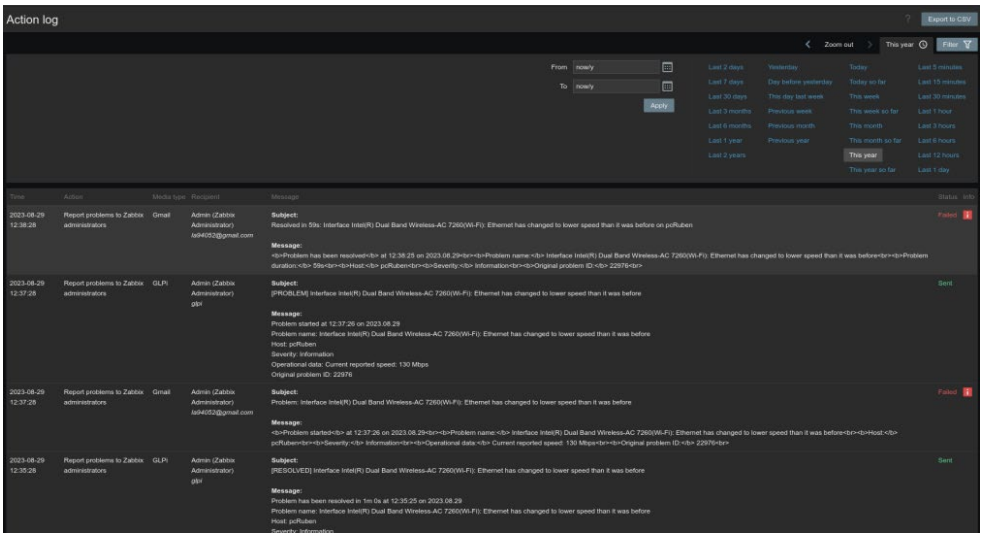


Figura 31 - Relatório de ações

Esta capacidade do Zabbix de fornecer *insights* visuais robustos sobre uma vasta gama de sistemas e infraestruturas torna-o uma ferramenta inestimável. Seja através de painéis personalizados, gráficos detalhados, mapas iterativos ou relatórios abrangentes, os administradores podem obter uma visão clara do desempenho e da saúde dos seus sistemas, otimizando a eficiência operacional e minimizando os tempos de inatividade. Ao adotar e implementar adequadamente as capacidades de visualização e análise de dados oferecidas pelo Zabbix, as organizações estão bem posicionadas para enfrentar os desafios tecnológicos de hoje e amanhã.

4.4. *Media Types* e integrações

Uma das características mais valiosas de uma solução de monitorização é a sua capacidade de comunicar eficazmente os alertas e notificações aos administradores. O Zabbix aborda esta necessidade através dos seus *Media Types*.

4.4.1. Media Types

No Zabbix, os *Media Types* atuam como definidores dos canais pelos quais os alertas são transmitidos. Estes canais podem variar desde e-mails a mensagens SMS e aplicativos de mensagens instantâneas. Ao utilizar o *Gmail* como canal, por exemplo, os administradores garantem que receberão notificações em tempo real diretamente na sua caixa de entrada, sempre que eventos específicos ou problemas surgirem. Esta capacidade de notificação não apenas permite uma reação rápida aos incidentes, mas também estabelece um elo essencial entre a identificação de um problema e sua subsequente solução. Este elo é crucial para garantir a continuidade do serviço e a

satisfação do cliente. O exemplo de um *media type* encontra-se ilustrado nas figuras subsequentes, Figuras 32 a 34.

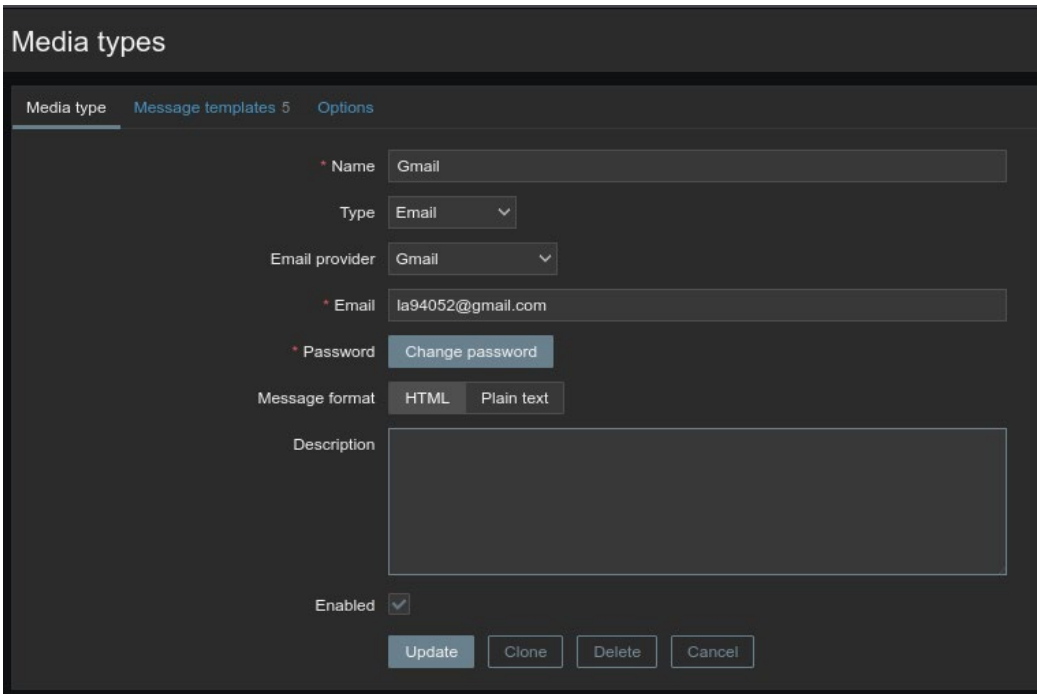


Figura 32 - Configuração Media Type

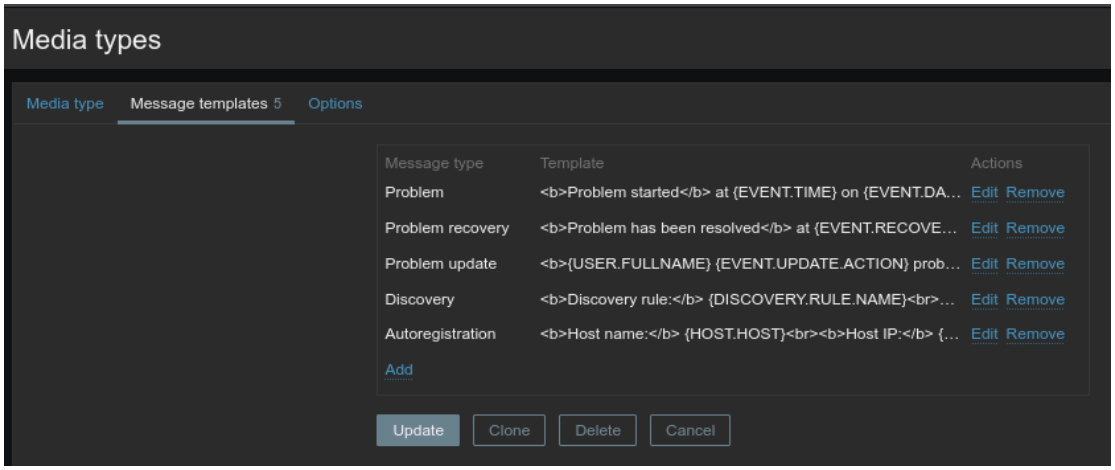


Figura 33 - Templates das mensagens relativas ao media type configurado

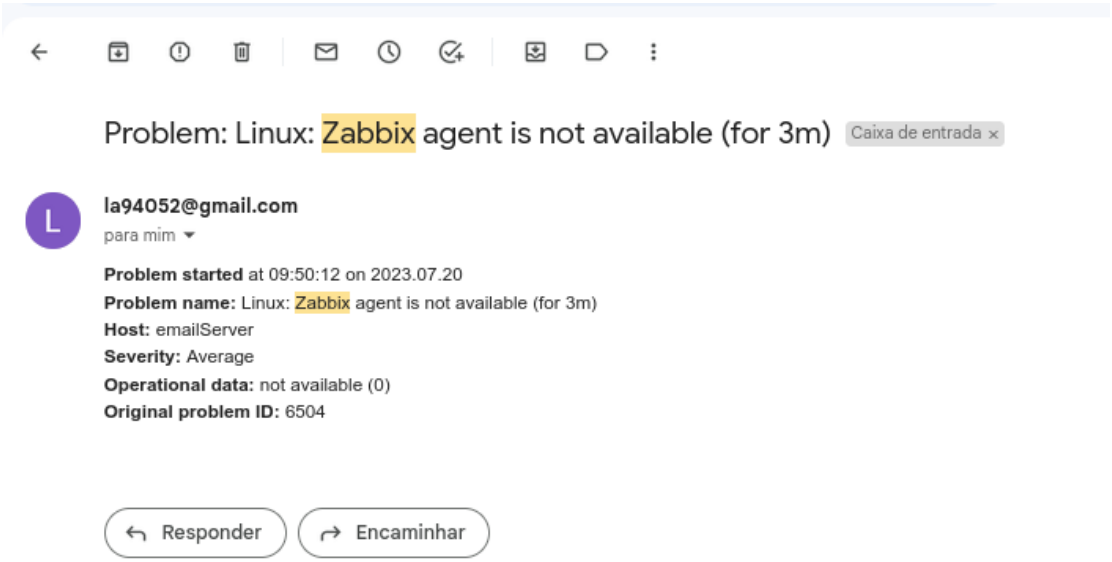


Figura 34 - Exemplo de media type implementado

4.4.2. Integração com GLPi

No universo das soluções de monitorização e gestão de TI, a integração entre diferentes plataformas torna-se uma vantagem competitiva, agregando maior funcionalidade e eficácia nas operações. Uma das integrações mais conhecidas e valiosas é entre o Zabbix e o GLPi.

A integração com o Zabbix oferece a capacidade de estabelecer uma ligação direta aos alertas de monitorização do Zabbix, com o sistema de gestão de incidentes do GLPi. Isto significa que, quando um alerta é disparado no Zabbix, um problema é automaticamente criado no GLPi, agilizando assim o processo de resposta e garantindo que os problemas sejam documentados e tratados de forma adequada.

Esta integração proporciona diversos benefícios:

Automação de *workflows*: Minimiza a intervenção manual, reduzindo erros humanos e garantindo que todos os alertas sejam tratados.

Centralização de registos: Todos os incidentes são registados no GLPi, assegurando uma sequência de eventos experimental e permitindo análises retrospetivas.

Melhoria na resposta de incidentes: Ao criar problemas automaticamente, garante-se que as equipas relevantes são notificadas imediatamente, o que acelera o tempo de resolução.

Documentação apropriada: Cada alerta do Zabbix é acompanhado de detalhes técnicos que são registados no problema do GLPi, ajudando no diagnóstico e resolução.

Para efetivar essa integração, muitas organizações utilizam *plugins* ou *scripts* que facilitam a comunicação entre as duas plataformas. É importante mencionar que, para uma integração bem-sucedida, é essencial uma configuração adequada e testes rigorosos, garantindo que os alertas sejam convertidos em problemas de forma consistente e confiável.

Em conclusão, a integração do Zabbix com o GLPi combina a robustez da monitorização do Zabbix com eficiência de gestão do GLPi, oferecendo às empresas uma solução compreensiva para a gestão de seus recursos de TI e respostas a incidentes.

O processo de integração encontra-se documentada em Anexo 16.

4.4.3. Integração com Grafana

Dentro do ecossistema de monitorização e visualização de TI, a combinação eficaz de diversas ferramentas pode amplificar a capacidade de uma organização de compreender e responder ao seu ambiente operacional. Uma das combinações mais poderosas e reconhecidas é a integração entre o Zabbix e o Grafana.

Ao integrar o Zabbix ao Grafana, as organizações podem:

Visualizar dados em profundidade: Através do Grafana, os dados recolhidos pelo Zabbix podem ser visualizados com uma riqueza de detalhes e variedade que vai além das visualizações padronizadas do Zabbix.

Flexibilidade na apresentação: O Grafana permite uma ampla personalização, garantindo que as métricas mais relevantes sejam destacadas conforme as necessidades específicas de cada organização.

Consolidação de dados: Grafana pode combinar dados do Zabbix com outras fontes, permitindo uma visão unificada de várias métricas e sistemas.

Alertas avançados: Com a combinação do Zabbix e Grafana, as empresas podem criar alertas mais complexos e visualmente atraentes.

A integração é geralmente realizada através de um *plugin* Zabbix para Grafana, que facilita a comunicação entre as duas plataformas. Esse *plugin* permite que o Grafana aceda e visualize dados diretamente do servidor Zabbix, transformando-os em gráficos e painéis iterativos.

A importância desta integração reside na capacidade de melhorar a visibilidade operacional e oferecer ensinamentos mais profundos sobre o desempenho e a saúde dos sistemas. A visualização do Grafana alimentada pelos dados robustos do Zabbix, podem ajudar as equipes de TI a detetar padrões, identificar problemas e otimizar o desempenho de seus sistemas.

O detalhe do processo de integração é documentado nos Anexo 17.

4.5. Caso de estudo com script em bash

Nesta parte do capítulo vou referenciar o caso de estudo realizado a pedido de um colaborador da empresa com o objetivo de mostrar as tarefas rotineiras aos demais membros da equipa através do Zabbix tendo um ficheiro CSV como origem dos dados.

Para viabilizar este processo foi necessário criar tanto um item como um trigger associados ao mesmo. As configurações detalhadas desses elementos estão documentadas em Anexo 13.2 e Anexo 14.1. De referir que a expressão do trigger foi definida para verificar o último valor recebido do item desejado e, caso esse valor não for igual a zero, irá acionar um alerta classificado á minha escolha com a severidade de "Desastre".

A conceção relacionada a este caso de estudo fundamenta-se na execução do script que envia as linhas com a mesma data da máquina para o item do Zabbix, desencadeando, posteriormente, o *trigger* uma vez que o último valor recebido se torna distinto de zero.

De seguida mostro o script em bash usado para este efeito.

```
#!/bin/bash

# Formato da data

DATE_FORMAT="+%d-%m-%Y"

# Data da máquina

TODAY=$(date "$DATE_FORMAT")

# Não ler primeira linha e fazer loop pelas restantes

tail -n +2 "/home/kali/Documents/scriptTasksZabbix/tasks.csv" | while
IFS= read -r line

do

      # Se começar com a data igual à data da máquina envia para o Zabbix

      if [[ $line = "$TODAY"* ]]

      then

            zabbix_sender -z 127.0.0.1 -s ZabbixServer -k due.tasks -o
      "$line"

      fi

done
```

A figura seguinte, Figura 35 ,demonstra o conteúdo do ficheiro CSV referido anteriormente.

	A	B	C
1	Date	Task	Client
2	30-08-2023	Set up new server rack	Acme Corp
3	30-08-2023	Configure VPN access for new users	Techtonic Ltd
4	30-08-2023	Audit cybersecurity protocols	Global Solutions
5	30-08-2023	Install software patches on workstations	Nexa Enterprises
6	31-08-2023	Configure firewall settings	Web Innovate
7	02-09-2023	Upgrade data center cooling system	Blink Labs
8	03-09-2023	Perform network vulnerability scan	Creative Spark
9	04-09-2023	Replace old routers	FastTech Industries
10	06-09-2023	Implement cloud backup solution	SolarPower Inc.
11	08-09-2023	Install new UPS units in server room	EcoBuild Consortium
12	10-09-2023	Optimize server virtualization	TechForge Inc.
13	12-09-2023	Patch OS vulnerabilities	Global Web Services
14	14-09-2023	Review and update DRP (Disaster Recovery Plan)	Titan Tech
15	16-09-2023	Configure network load balancer	SecureNet
16	18-09-2023	Plan for end-of-life hardware replacement	CyberCore
17	20-09-2023	Perform database replication tests	DataHub Enterprises
18	22-09-2023	Optimize storage area network (SAN)	NetGen Solutions
19	24-09-2023	Document IT infrastructure topology	Quantum Innovations

Figura 35 - Contéúdo do ficheiro tasks.csv

De seguida demonstro a execução do script no terminal, o seu output e o resultado na interface do servidor Zabbix ilustrado na Figura 36.

```
┌──(kali㉿kali)-[~/Documents/scriptTasksZabbix]
└─$ ./script2.sh
Response from "127.0.0.1:10051": "processed: 1; failed: 0; total: 1; seconds spent: 0.000072"

sent: 1; skipped: 0; total: 1

Response from "127.0.0.1:10051": "processed: 1; failed: 0; total: 1; seconds spent: 0.000118"

sent: 1; skipped: 0; total: 1

Response from "127.0.0.1:10051": "processed: 1; failed: 0; total: 1; seconds spent: 0.000278"

sent: 1; skipped: 0; total: 1

Response from "127.0.0.1:10051": "processed: 1; failed: 0; total: 1; seconds spent: 0.000183"

sent: 1; skipped: 0; total: 1
```

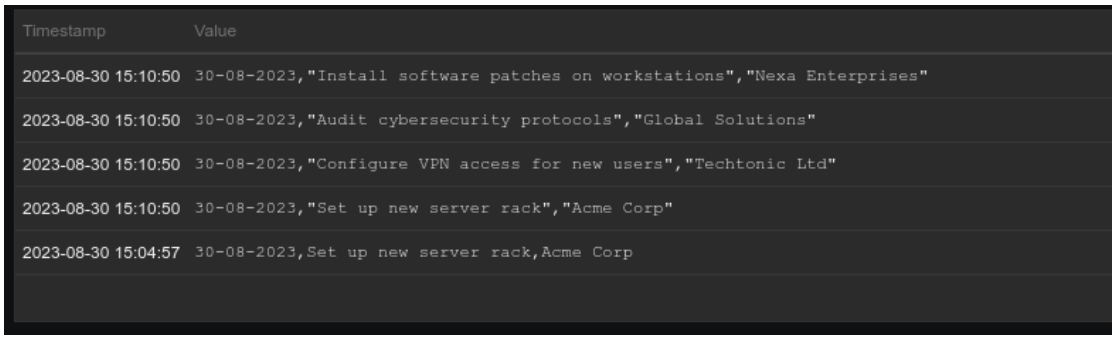


Figura 36 - Resultado do caso de estudo na interface do servidor do Zabbix

Além das etapas já mencionadas, uma medida adicional que poderia ser implementada para otimizar ainda mais esse processo é a criação de um cron job. Um cron job é uma tarefa programada que pode ser configurada para ser executada automaticamente em intervalos específicos de tempo. Neste contexto, a configuração de um cron job permitiria a execução do script de forma regular e programada, garantindo que as tarefas de atualização dos dados no Zabbix sejam realizadas em momentos predefinidos.

A programação do cron job poderia ser configurada para executar o script diariamente, semanalmente ou conforme necessário, de acordo com os requisitos do projeto e as necessidades operacionais da empresa. Além disso, seria possível definir o horário exato para a execução do script, garantindo que as atualizações ocorram no momento mais apropriado.

Essa abordagem automatizada não apenas aumentaria a eficiência da atualização de dados no Zabbix, mas também reduziria a intervenção manual, garantindo a consistência e a precisão das informações monitorizadas.

A implementação de um cron job, portanto, representa uma estratégia sólida para a gestão eficaz das tarefas rotineiras de monitorização e é altamente recomendada para aprimorar ainda mais a automação e a eficiência do sistema Zabbix.

Capítulo 5

5.Centralização de acesso aos serviços

A crescente complexidade das infraestruturas tecnológicas corporativas tem impulsionado a necessidade de soluções eficientes que facilitem o acesso e a gestão de serviços críticos de forma unificada. No contexto das operações de monitorização e gestão de sistemas, a capacidade de aceder rapidamente a ferramentas como o Zabbix, Grafana e GLPi assume um papel central na otimização do fluxo de trabalho e na tomada de decisões informadas.

Este capítulo representa um estudo detalhado sobre a criação de uma aplicação móvel e web em React Native, projetada para fornecer uma solução centralizada e acessível para colaboradores interagirem com as mencionadas ferramentas. Ao unificar a experiência de acesso em várias plataformas, incluindo web, Android e iOS, a aplicação proporciona uma interface intuitiva que agiliza a interação com as funcionalidades oferecidas por cada sistema de monitorização e gestão.

5.1. React Native

React Native tem ganhado destaque no desenvolvimento de aplicações móveis devido à sua abordagem inovadora de criar interfaces nativas usando componentes reutilizáveis e uma linguagem familiar para a *web: JavaScript*. Ao adotar o React Native para este projeto, foi aproveitado vantagens significativas que a tecnologia oferece.

5.1.1. Desenvolvimento multi-plataforma

O React Native possibilita a criação de aplicações para múltiplas plataformas (iOS, *Android e web*) a partir de um único código-fonte. Isso reduz drasticamente o esforço de desenvolvimento e manutenção, permitindo que a aplicação seja acessível em diferentes dispositivos.

5.1.2. Interface nativa e desempenho

Ao contrário das abordagens de desenvolvimento híbridas, o *React Native* oferece uma interface de utilizador verdadeiramente nativa. Isso resulta numa experiência de utilizador mais fluida e responsiva, comparável àquelas desenvolvidas utilizando linguagens nativas.

5.1.3. Reutilização de componentes

O React Native promove a reutilização de componentes entre as diferentes plataformas. Isso permite que partes significativas do código sejam compartilhadas, diminuindo o tempo de desenvolvimento e garantindo consistência na interface e funcionalidades.

5.2. Implementação da aplicação

A aplicação foi projetada com um design de interface intuitivo, com botões distintos para cada serviço. A implementação desses botões permitiu que os utilizadores acedessem rapidamente às funcionalidades específicas de monitorização e gestão. O código fonte desta aplicação encontra-se documentada em Anexo 18.

5.3. Importância da facilitação de acesso

Uma das principais preocupações em lidar com ambientes tecnológicos complexos é a simplificação do acesso a serviços críticos. Através da criação de uma aplicação React Native, eliminamos a necessidade de memorizar ou registar manualmente os endereços dos servidores relacionados com cada serviço. Isso resulta numa experiência de utilizador mais fluida, onde os colaboradores podem aceder rapidamente os serviços sem a necessidade de se lembrar de URLs ou endereços IP.

5.4. Vantagens em ambientes de produção

Em ambientes de produção que envolvem múltiplos servidores Zabbix, GLPi e Grafana, a centralização do acesso oferece uma vantagem significativa. A aplicação em React Native proporciona uma interface única para acesso a todos esses servidores, desde que o equipamento tenha uma ligação estabelecida com a rede interna da empresa. Isso elimina a complexidade de lidar com diferentes endereços e garante que os colaboradores possam interagir com os serviços relevantes de forma mais eficiente.

O resultado desta aplicação desenvolvida em funcionamento é ilustrado nas figuras seguintes, Figuras 37 a 41.

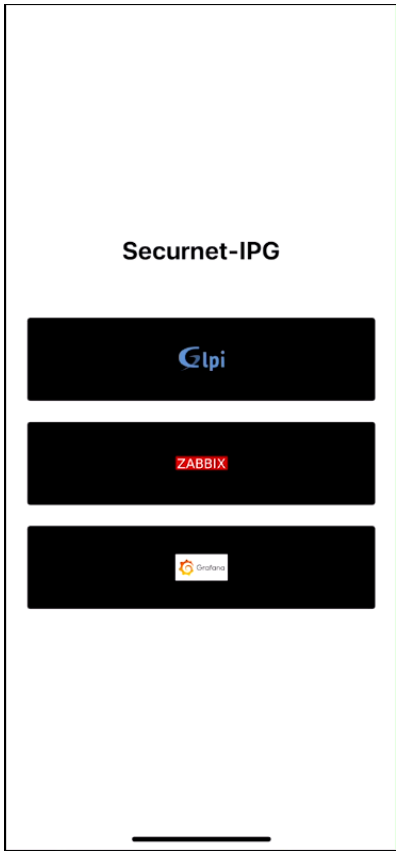


Figura 37 - App em iOS

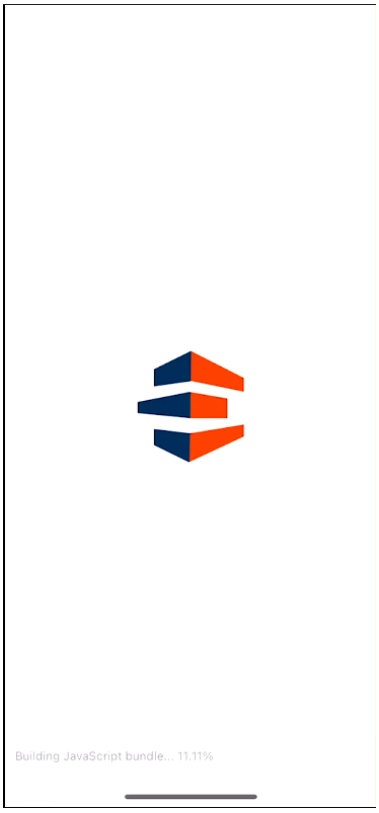


Figura 38 - Splash Screen em iOS

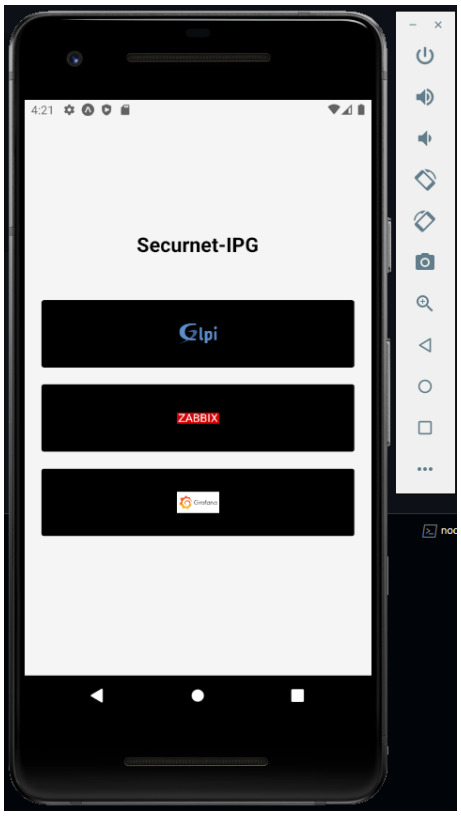


Figura 39 - App em Android

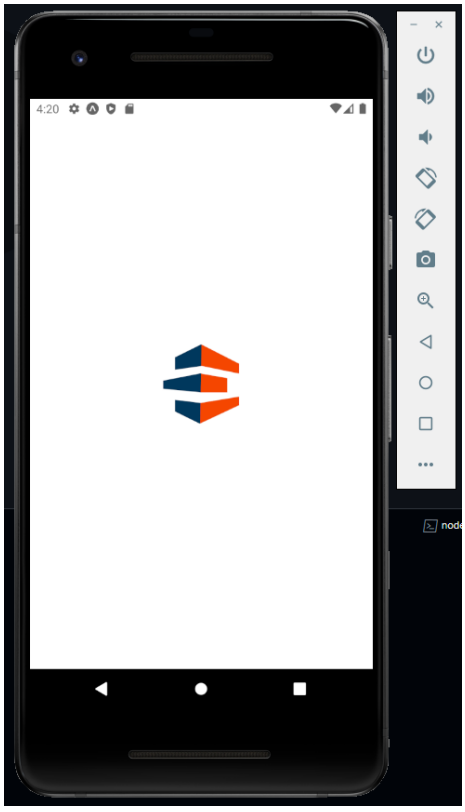


Figura 40 - Splash Screen em Android

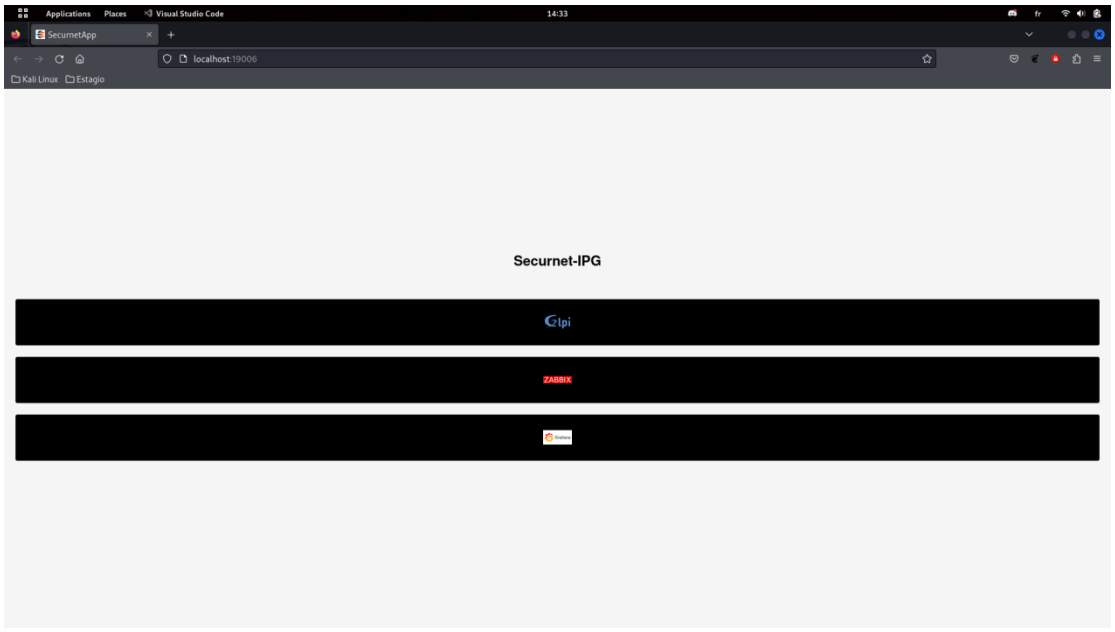


Figura 41 - App em web

Capítulo 6

6. Análise dos dados do Zabbix

À medida que o volume de dados nas organizações cresce exponencialmente, a necessidade de extrair informações significativas desses dados torna-se cada vez mais crítica. O mundo digital contemporâneo gera um fluxo contínuo de informações que, se bem explorado, pode fornecer ensinamentos valiosos, impulsionar a eficiência operacional e oferecer vantagens competitivas.

Dentro deste contexto, o *machine learning* emerge como uma ferramenta poderosa. Ao aplicar algoritmos que melhoram automaticamente através da experiência, o *machine learning* pode ajudar as organizações a identificar padrões e anomalias nos dados, otimizar processos e tomar decisões mais informadas.

Dada a quantidade vasta de eventos que um sistema Zabbix pode gerar, a aplicação de técnicas de *machine learning* torna-se não apenas atraente, mas quase essencial. Em ambientes complexos, distinguir entre eventos genuinamente críticos e ruído pode ser um desafio. Esta abordagem visa aplicar *o machine learning* para classificar e priorizar esses eventos, tornando a gestão e a resposta a incidentes mais eficientes.

Para ilustrar esse processo, a empresa forneceu um fluxograma que descreve como proceder quando aparece um alarme no Zabbix ilustrado na figura seguinte, Figura 42.

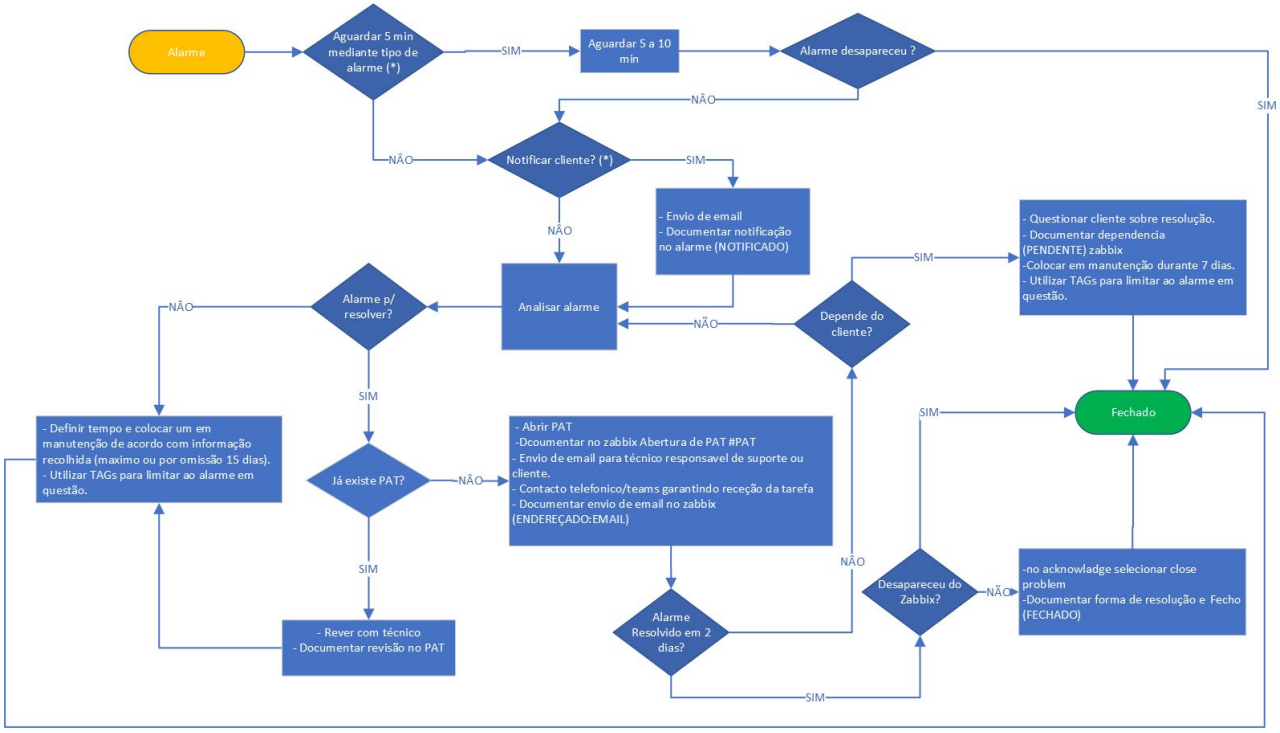


Figura 42 - Fluxograma de procedimento após alarme no Zabbix

Ao longo deste capítulo, será explorada a metodologia empregada, os algoritmos utilizados e os resultados obtidos, lançando luz sobre o potencial da combinação entre monitorização de sistemas e *machine learning*.

6.1. Escolha do algoritmo

A análise dos eventos registados no Zabbix, abrangendo uma compilação de métricas e indicadores diversificados, visa primordialmente à segmentação destes eventos de acordo com as características inerentes, como frequência de ocorrências dos eventos e da duração média dos mesmos. A opção pelo método *KMeans* fundamenta-se em múltiplos critérios. Primeiramente, a natureza da informação sobre análise exige uma segmentação baseada em características numéricas contínuas. O algoritmo *KMeans* destaca-se nesta categoria de dados devido à sua capacidade de minimizar a variância *intra-cluster*. Do ponto de vista interpretativo, o *KMeans* oferece uma claridade ímpar: cada

cluster é representado por um ponto central, o qual pode ser concebido como a figura "representativa" daquele conjunto específico. No Zabbix, tal peculiaridade facilita a compreensão imediata do comportamento padrão de um conjunto determinado de eventos.

Relativamente à eficiência computacional, ao lidar com conjuntos de dados de magnitude elevada, a eficiência emerge como uma consideração crucial. Este algoritmo é notório pela sua celeridade operacional, sobretudo quando comparado com técnicas de maior complexidade, como o agrupamento hierárquico. No eventual cenário de incorporação de atributos adicionais ao *dataframe*, o *KMeans* demonstra a sua adaptabilidade, permitindo uma execução simplificada com as informações adicionadas. Ademais, embora a prerrogativa de determinar o número de *clusters* possa inicialmente aparentar ser uma limitação, no contexto do Zabbix, essa especificidade é benéfica. A título ilustrativo, se existe uma previsibilidade quanto ao número de categorias desejado, é possível parametrizar o *KMeans* para segmentar conforme tal especificação. Importa referir que este *dataset* será obtido após um processo de extração de dados que será abordado posteriormente.

6.1.1. Algoritmo KMeans

O Kmeans é um dos algoritmos mais conhecidos e utilizados na área de agrupamento (ou clustering) de dados. O seu objetivo primordial é particionar um conjunto de dados em K grupos distintos, onde K é definido previamente pelo utilizador. Cada grupo é identificado pela média dos dados que pertencem a esse grupo, designada de centroide.

Funcionamento do algoritmo:

Inicialização: Escolher K pontos do conjunto de dados de forma aleatória (ou através de alguma técnica especifica) para servirem como centroides centrais.

Atribuição: Atribuir cada ponto do conjunto de dados ao centroide mais próximo, com base numa métrica de distância (frequentemente, a distância euclidiana é utilizada).

Atualização: Recalcular os centroides como a média aritmética de todos os pontos atribuídos a cada centroide.

Convergência: Repetir os passos dois e três até que os centroides não se alterem significativamente entre iterações consecutivas ou até que se atinja um número máximo predefinido de iterações.

Considerações técnicas:

A qualidade do resultado do KMeans pode ser sensível à inicialização dos centroides. Para contornar isto, é comum realizar várias inicializações e escolher a solução com a menor soma das distâncias quadradas entre cada ponto e o centroide correspondente.

O valor de K precisa de ser especificado a priori, o que pode ser considerado uma desvantagem. Métodos como o método do cotovelo (*elbow method*) são frequentemente utilizados para ajudar a determinar um valor apropriado de K.

O KMeans assume que os grupos são esféricos e de tamanhos semelhantes, o que pode não ser verdade em muitos conjuntos de dados reais. Quando esta suposição não é válida o algoritmo pode não produzir os melhores agrupamentos possíveis.

O algoritmo é sensível a *outliers*, o que pode afetar a localização dos centroides e, consequentemente, a qualidade dos agrupamentos.

Distância Euclidiana:

A distância euclidiana é uma métrica para medir a distância entre dois pontos num espaço n-dimensional. Dados dois pontos P=(p1,p2,...,pn) e Q=(q1,q2,...,qn), a distância euclidiana dd entre eles é definida pela fórmula:

$$d(P,Q) = \sqrt{\sum_{i=1}^{n} (qi - pi)^2}$$

Esta métrica, nomeada em homenagem ao matemático grego Euclides, é amplamente utilizada em ciência da computação, matemática e engenharia para calcular a proximidade entre pontos num espaço euclidiano. [5]

6.2. Implementação

.

Nesta implementação é abordada a tarefa de extrair, transformar e modelar dados do Zabbix usando técnicas avançadas de *machine learning*. O objetivo central é segmentar os eventos em categorias de relevância, fornecendo uma visão clara e acionável sobre quais eventos necessitam de maior atenção e quais podem ser considerados ruído e sendo assim ser necessário adicionar *thresholds* aos mesmos.

6.2.1. Extração dos dados do MariaDB

O código começa com a importação das bibliotecas necessárias e estabelece uma conexão com a base de dados MariaDB. A seguir, executa uma consulta SQL para extrair dados relacionados ao Zabbix. Esta consulta:

- Extrai informações relacionadas a eventos, *itens, triggers e hosts* do Zabbix
- Combina essas informações através de diversas operações 'JOIN'
- Filtra eventos com 'object = 0' e 'value = 1'
- Calcula a contagem de eventos e a duração média dos mesmos
- Classifica os *triggers* com base na sua prioridade
- Agrupa os resultados por *host*, descrição do *trigger* e prioridade

Após a extração, os dados são transformados num *dataframe*. No que diz respeito às transformações efetuadas neste *dataframe:*

- A coluna 'avg_duration' (duração média) é transformada de segundos para minutos
- Os valores da duração média são arredondados para duas casas decimais
- O *dataframe* é ordenado pela contagem de eventos em ordem decrescente

Finalmente, após as devidas transformações, o resultado é salvo num ficheiro em formato CSV.

Um "excerto" desse *dataframe* pode ser visualizado na figura subsequente, Figura 43.

	host	description	priority	event_count	avg_duration
1	pcRuben	Interface Intel(R) Dual Band Wireless-AC 7260(Wi-Fi): Ethernet has changed to lower speed	Information	2379	15.27
2	pcAfonso	Interface Intel(R) Dual Band Wireless-AC 8265(Wi-Fi): Ethernet has changed to lower speed	Information	231	1.01
3	pcFrancisco	Windows: The Memory Pages/sec is too high	Warning	111	145.39
4	pcFrancisco	Windows: CPU privileged time is too high	Warning	102	214.08
5	pcRuben	Windows: The Memory Pages/sec is too high	Warning	91	53.75
6	pcFrancisco	Interface Intel(R) Dual Band Wireless-AC 8260(Wi-Fi): Ethernet has changed to lower speed	Information	81	3.56
7	pcAfonso	Windows: The Memory Pages/sec is too high	Warning	74	234.11
8	ZabbixServer	Interface eth0: Link down	Average	69	426.17
9	ZabbixServer	Interface wlan0: Link down	Average	60	200.72
10	Firewall	Fortinet {HOST.NAME} Rebooted	Average	56	4.49
11	windowsHP	"dmwappushservice" (Device Management Wireless Application Protocol (WAP) Push mess	Average	55	1060.49
12	FirewallLAN	Fortinet {HOST.NAME} Rebooted	Average	54	9.97

Figura 43 - Excerto do dataframe

O código utilizado para a extração de dados da base de dados do Zabbix encontra-se em Anexo 19.1.

O *dataframe* utilizado também se encontra devidamente documentado no Anexo 20.

6.2.2. Elbow method

Como referido anteriormente o *elbow method* é um recurso amplamente utilizado para auxiliar na determinação do valor de K num algoritmo como o KMeans. Para aplicar este método executa-se o agrupamento várias vezes com diferentes valores de K e calcula-se a soma das distâncias quadradas (ou inércia) entre os pontos de dados e os centroides dos seus respetivos clusters.

Ao traçar a soma das distâncias quadradas para cada valor de K, observa-se que a inércia decresce á medida que K aumenta. O valor do K no qual essa diminuição se torna novamente menor e começa a estabilizar é chamado de "cotovelo", já que o gráfico tende a ter uma forma semelhante ao de um braço de um ser humano.

O ponto "cotovelo" é considerado uma indicação do número ótimo de clusters, pois representa um equilíbrio entre posição e complexidade computacional.

Para uma compreensão mais detalhada do método, o código utilizado para determinar o método do cotovelo para os dados extraídos anteriormente encontra-se documentado em Anexo 19.2. O gráfico resultante deste código encontra-se ilustrado na figura subsequente, Figura 43.

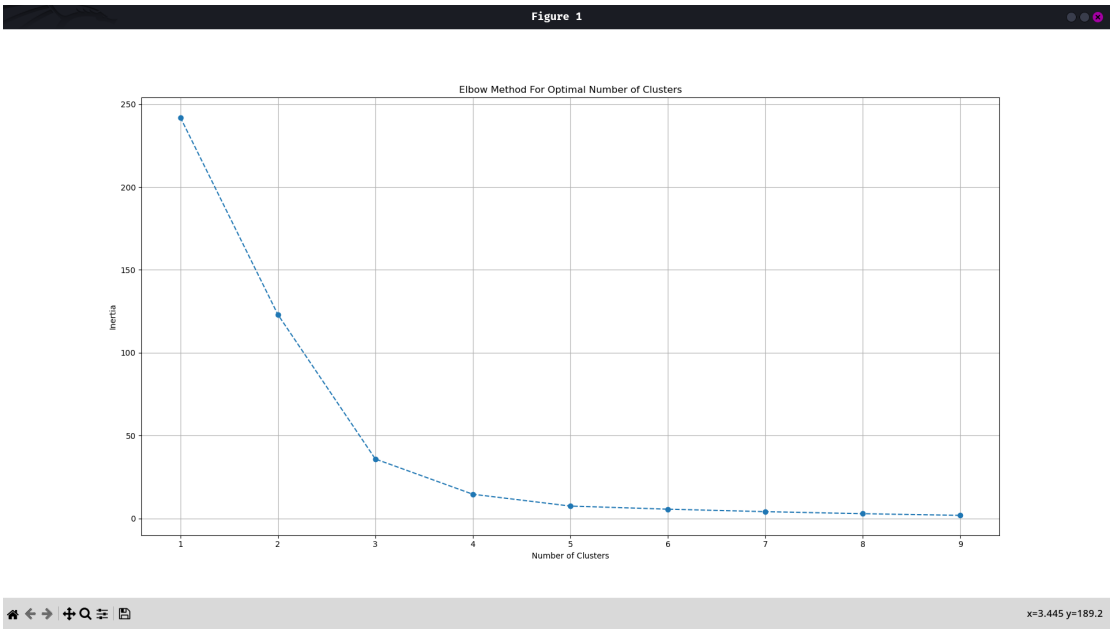


Figura 44 - Elbow method implementado nos dados extraídos

Saliente-se que, à luz do método adotado e da interpretação gráfica, constata-se que o ponto de interesse se situa num intervalo cujos valores oscilam entre três e quatro. Consequentemente, o valor de KK definido para esta análise foi de 3.

6.2.3. Processamento e modelagem

Após a extração e transformação dos dados, o código lê novamente o ficheiro CSV, normaliza algumas colunas, aplica o algoritmo *KMeans*, documentado em *Anexo 19.3*, para agrupar os eventos e, em seguida, adiciona o rótulo dos clusters ao *dataframe*.

Para rotular os eventos, uma função ´label_events´ é definida para categorizar os dados em três categorias distintas com base na contagem e duração média:

- "Mais irrelevante": Eventos com alta contagem e duração média curta.
- "Irrelevante": Eventos com alta contagem e duração média a elevada.
- "Relevante": Os restantes eventos, geralmente caracterizados por terem uma contagem baixa.

As categorias mencionadas representam os resultados desejados da clusterização feita pelo algoritmo. Assim, esta classificação manual facilita a subsequente comparação entre os resultados obtidos pela função e aqueles produzidos pelo algoritmo.

Finalmente, o *dataframe* atualizado é guardado num outro ficheiro CSV. E, para avaliar o desempenho do agrupamento, é calculada o coeficiente de silhueta, que é impressa no final.

6.3. Avaliação do modelo

A avaliação do desempenho de um modelo é um passo incontornável em qualquer projeto de *machine learning*. Esta fase garante que o modelo desenvolvido está funcionando como o esperado e oferece *insights* sobre áreas de possível aprimoramento.

6.3.1. Coeficiente de Silhueta

A utilização do coeficiente de silhueta como métrica de avaliação é fundamental na análise da qualidade dos agrupamentos gerados. Esta métrica oferece um equilíbrio entre a coesão (quão próximos os membros de um *cluster* estão entre si) e a separação (quão bem separados estão os diferentes *clusters*).

Para melhor interpretar o coeficiente de silhueta:

- Valores perto de +1 sugerem que a amostra está bem agrupada e distante de outros *clusters*.
- Valores em torno de 0 indicam que a amostra está próxima da fronteira de decisão entre dois *clusters* adjacentes.
- Valores perto de -1 podem indicar que as amostras foram agrupadas no cluster errado.

A adjacência intra-*clusters* geralmente refere-se à proximidade entre eles, particularmente nas regiões de fronteira. No contexto do coeficiente de silhueta e de técnicas de agrupamento como o KMeans a adjacência intra-*clusters* denota a proximidade ou fronteira entre dois grupos distintos no espaço de características. Esta adjacência pode ser visualmente percebida nas regiões onde os limites dos clusters se encontram ou sobrepõem.

Explicação Técnica no Contexto do Coeficiente de Silhueta:

Coesão: Refere-se à média da distância entre uma amostra e todas as outras amostras no mesmo cluster. Representa quão bem uma amostra é agrupada com as amostras do seu próprio cluster. Uma coesão menor indica que as amostras estão mais próximas umas das outras no mesmo cluster.

Separação: É a distância média da amostra ao cluster mais próximo que ela não pertence. Uma maior separação indica que um cluster está distante dos seus clusters vizinhos.

Adjacência através do Coeficiente de Silhueta: O coeficiente de silhueta é calculado usando a fórmula $s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))}$. Esta fórmula é uma métrica amplamente usada para avaliar a qualidade de clusters. A mesma foi introduzida por Peter J. Rousseeuw em 1986 num artigo intitulado ″. No artigo Rousseeuw apresenta um coeficiente de silhueta como uma ferramenta gráfica para interpretar e validar os resultados da análise de clusters. Ele descreve um coeficiente de silhueta para uma única amostra, bem como a média de todos os coeficientes para avaliar a qualidade geral do agrupamento. [6]

A fórmula em si é derivada das medidas de coesão e separação, como mencionado:

- a(i): é a média de distância da i-ésima amostra para as outras amostras no mesmo cluster (coesão)
- b(i): é a menor média da distância da i-ésima amostra para amostras num cluster diferente, minimizada sobre clusters (separação)

Neste contexto, um coeficiente de silhueta de 0.8491 indica uma boa segmentação dos dados, com *clusters* bem definidos e distintos entre si. Este valor sugere que os eventos

do Zabbix foram agrupados de forma significativa e que o algoritmo foi capaz de discernir com clareza entre os diferentes tipos de eventos. Os detalhes desta análise e o coeficiente de silhueta obtido estão ilustrados na Figura 44.

```
                host  ...                                              relevance
0           pcRuben  ...  Most irrelevant ( HIGH COUNT | LOW AVG DURATIO...
13      ZabbixServer  ...  Most irrelevant ( HIGH COUNT | LOW AVG DURATIO...
12       pcFrancisco  ...  Most irrelevant ( HIGH COUNT | LOW AVG DURATIO...
11       FirewallLAN  ...  Most irrelevant ( HIGH COUNT | LOW AVG DURATIO...
9           Firewall  ...  Most irrelevant ( HIGH COUNT | LOW AVG DURATIO...
..               ...  ...                                              ...
51      ZabbixServer  ...          Relevant (DONT APPEAR OFTEN | LOW COUNT)
50   DESKTOP-44ORGLB  ...          Relevant (DONT APPEAR OFTEN | LOW COUNT)
49   DESKTOP-44ORGLB  ...          Relevant (DONT APPEAR OFTEN | LOW COUNT)
47         windowsHP  ...          Relevant (DONT APPEAR OFTEN | LOW COUNT)
120         pcAfonso  ...          Relevant (DONT APPEAR OFTEN | LOW COUNT)

[121 rows x 5 columns]
Silhouette_Score: 0.8490861059066114
```

Figura 45 - Output do programa

O output mostra uma tabela (parte do *dataframe)* contendo as colunas:

- '*host*': O nome ou identificação do host
- '*description*': A descrição associada ao evento
- 'event_count': A quantidade de vezes que o evento ocorreu
- 'avg_duration': A duração média do evento
- 'cluster': Cada evento foi atribuído a um cluster e a sua afiliação ao cluster é registado nesta coluna
- 'relevance': A categoria de relevância associada ao evento, determinada com base na função 'label_events' mencionada anteriormente

Este *dataframe* com os eventos já agrupados, após a aplicação do algoritmo ao dataframe referido anteriormente, encontra-se documentado no Anexo 21.

Para uma representação mais intuitiva e visual da eficácia deste método de agrupamento foi implementado um código que permite ilustrar de forma gráfica o coeficiente de silhueta de cada amostra, bem como a distribuição de eventos em termos de contagem e duração média. Através destes gráficos, torna-se mais fácil perceber as interações entre as amostras e a sua pertinência aos clusters a que foram atribuídas.

No primeiro gráfico gerado, a silhueta das amostras é distribuída por cluster, permitindo não só observar a consistência interna de cada agrupamento, mas também avaliar o coeficiente médio de silhueta de toda a análise. A linha vertical vermelha no gráfico

indica precisamente este valor médio, servindo de referência para avaliar a qualidade de cada cluster individualmente.

O segundo gráfico foca-se na visualização dos clusters, representando cada evento em função da sua contagem e duração média. Os centroides de cada cluster, ou seja, os pontos médios, são assinalados com marcas vermelhas. Esta representação permite perceber rapidamente a distribuição dos eventos e a forma como os diferentes algoritmos de agrupamento conseguem separá-los com base nestas duas variáveis.

Estas visualizações, quando conjugadas com as métricas quantitativas anteriormente mencionadas, proporcionam uma visão abrangente e detalhada da eficácia do método de agrupamento escolhido, facilitando assim a tomada de decisões sobre possíveis melhorias ou ajustes na análise.

O código associado a esta visualização de dados encontra-se documentado em Anexo 19.4. Os gráficos resultantes do mesmo são respetivamente ilustrados nas figuras subsequentes, Figura 45 e 46.

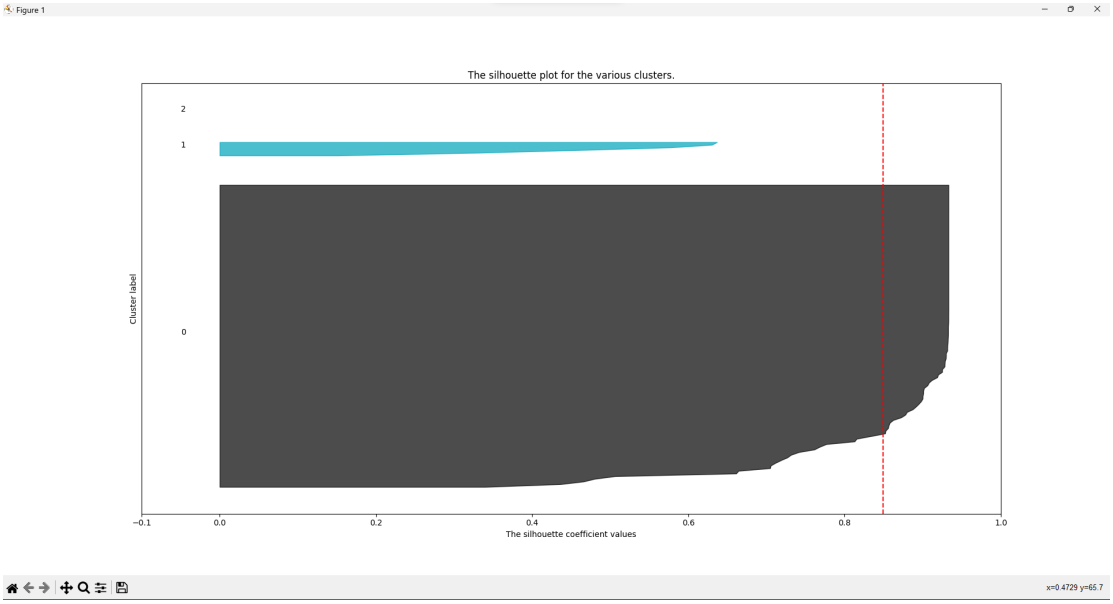


Figura 46 - Coeficiente de silhueta por cluster

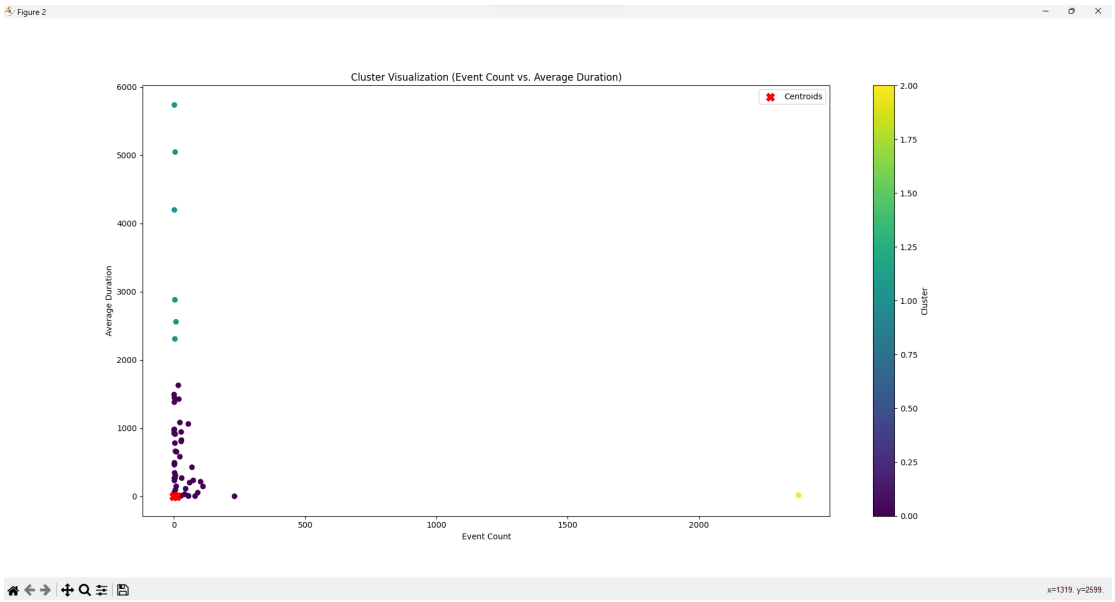


Figura 47 - Representação gráfica dos clusters

Podemos observar que na Figura 46:

- O cluster 0 apresenta uma altura considerável, indicando que uma grande quantidade de eventos foi associada a este cluster.
- O cluster 1 tem uma altura muito menor em comparação com o cluster 0, indicando que muito menos eventos foram categorizados neste cluster
- O cluster 2 é quase inexistente, visto que, como se pode observar na Figura 47 apenas um evento foi categorizado neste cluster

É crucial compreender que os algoritmos de *clustering*, como o KMeans, tendem a apresentar resultados mais robustos e elucidativos à medida que o tamanho do conjunto de dados aumenta. Por outras palavras, com uma amostra maior, o algoritmo teria mais informações para discernir padrões e características distintas dos eventos. A presença de mais dados pode também mitigar algumas das anomalias observadas, como a formação de um cluster com um único evento.

Assim, ao considerar a aplicação deste método a uma base de dados mais extensa, que seria o pretendido para este projeto, pode-se esperar alguns benefícios potenciais:

1. **Melhor segmentação**: A definição dos clusters poderá ser mais precisa e representativa das verdadeiras distribuições e tendências dos eventos.

2. **Redução de anomalias**: Com mais dados, os clusters tendem a ser mais equilibrados, e anomalias ou *outliers* podem ser melhor agrupados ou identificados.
3. **Melhor representação de padrões**: Uma base de dados mais extensa pode revelar padrões ou categorias de eventos que não são evidentes numa amostra menor.

Olhando para o futuro, há várias direções possíveis para aprimorar ainda mais este trabalho. Pode-se considerar a aplicação de outros algoritmos de *clustering* para comparar a eficiência ou a incorporação de mais atributos ao *dataset*, como por exemplo, a severidade dos eventos, para uma segmentação mais detalhada. Além disso, a integração deste modelo diretamente na interface do Zabbix pode proporcionar alertas e *insights* em tempo real, otimizando ainda mais a gestão de eventos.

Em resumo, a combinação da monitorização de sistemas com Machine Learning apresenta um caminho promissor para as organizações na era digital. O estudo aqui apresentado é um passo nessa direção, e a jornada de inovação continua.

Capítulo 7

7.Conclusão

Ao longo deste documento, abordámos os desafios e oportunidades associados à implementação e monitorização de redes informáticas num ambiente empresarial. O projeto em foco demonstrou que, com as ferramentas e metodologias corretas, é possível otimizar a gestão de redes, garantindo eficiência operacional e segurança.

No universo das redes informáticas, aprendemos sobre a importância da engenharia de redes, e como sua implementação e gestão adequadas são fundamentais para a sustentabilidade e segurança das operações de uma empresa. Através da configuração de uma rede simulada, abordámos práticas essenciais para a implementação de *firewalls*, *switches* e servidores, destacando a importância de uma DMZ para proteger os ativos de TI.

A introdução do servidor Zabbix provou ser um passo vital para a monitorização eficaz desta rede. A sua integração com outras ferramentas, como Grafana e GLPi, ampliou as capacidades de visualização e gestão, tornando mais fácil identificar e responder a possíveis problemas.

O uso do *Machine Learning*, particularmente na análise de dados do Zabbix, sublinhou o potencial desta tecnologia na otimização da gestão de redes.

Esta fusão da monitorização de redes com o Machine Learning facilita a filtragem de eventos que, num ambiente empresarial, seriam considerados de ruído. A adição de *thresholds* a esses eventos permite minimizar esse mesmo ruído, destacando apenas os eventos relevantes que exigem intervenção técnica. Esta abordagem melhora significativamente a eficiência da monitorização.

A criação da aplicação em *React Native* evidencia a importância de ter uma interface centralizada, permitindo um acesso facilitado e integrado aos diversos serviços da rede.

Para o futuro, várias direções podem ser exploradas. A contínua evolução das técnicas de *machine learning* e Inteligência Artificial pode abrir portas para análises mais profundas e automações avançadas. A expansão da aplicação criada em *React Native*, com a integração de mais funcionalidades e ferramentas, também representa uma possibilidade promissora.

Em suma, este projeto reforçou a intersecção entre a engenharia de redes e as tecnologias emergentes, como *machine learning*. Sublinha-se a ideia de que, com a implementação e gestão corretas, as redes informáticas podem ser poderosos pilares de apoio às operações empresariais, contribuindo para um ambiente mais seguro e eficiente.

Bibliografia

[1] Lda, Reload - Consultoria Informática, “Securnet,” [Online]. Available: https://www.securnet.pt/.

[2] Fortinet, “Fortinet - Document Library,” [Online]. Available: https://docs.fortinet.com/product/fortigate/hardware.

[3] LLC, Zabbix, "Download and Install Zabbix," [Online]. Available: https://www.zabbix.com/download?zabbix=6.4&os_distribution=debian&os_version=12&components=server_frontend_agent&db=mysql&ws=apache.

[4] B. v. B. Nathan Liefting, Zabbix 6 IT Infrastructure Monitoring Cookbook: Explore the new features of Zabbix 6 for designing, building, and maintaining your Zabbix setup, Packt Publishing Ltd., 2022.

[5] I. Dabbura, “K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks,” 17 Setembro 2018. [Online]. Available: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a.

Anexos

Anexo 1 - Criação das VM's

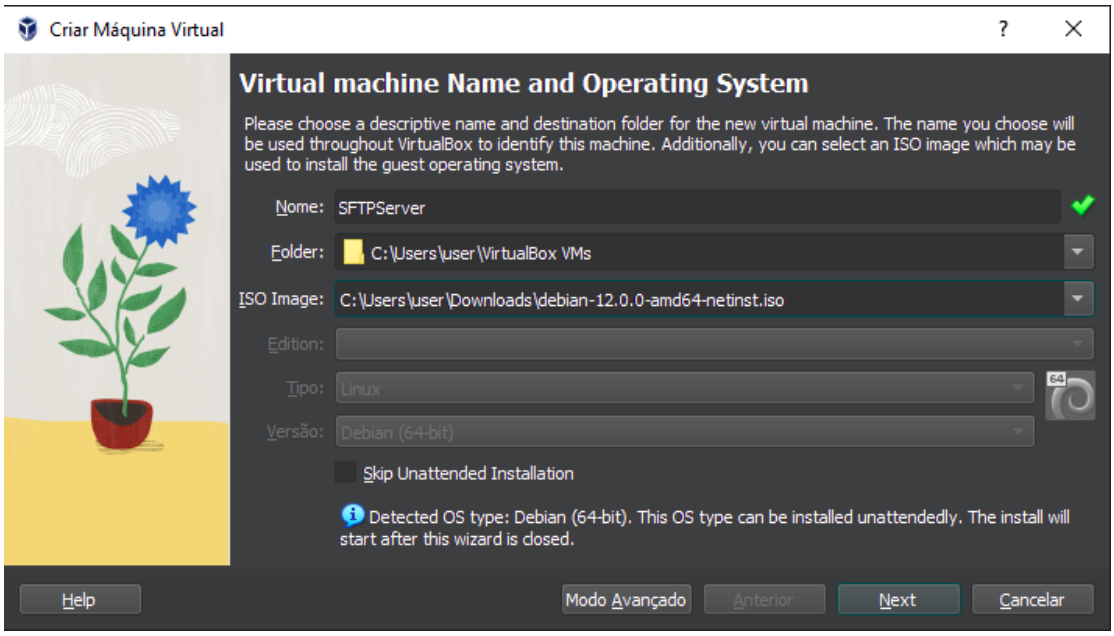


Figura 48 - Passo Inicial da criação de uma VM

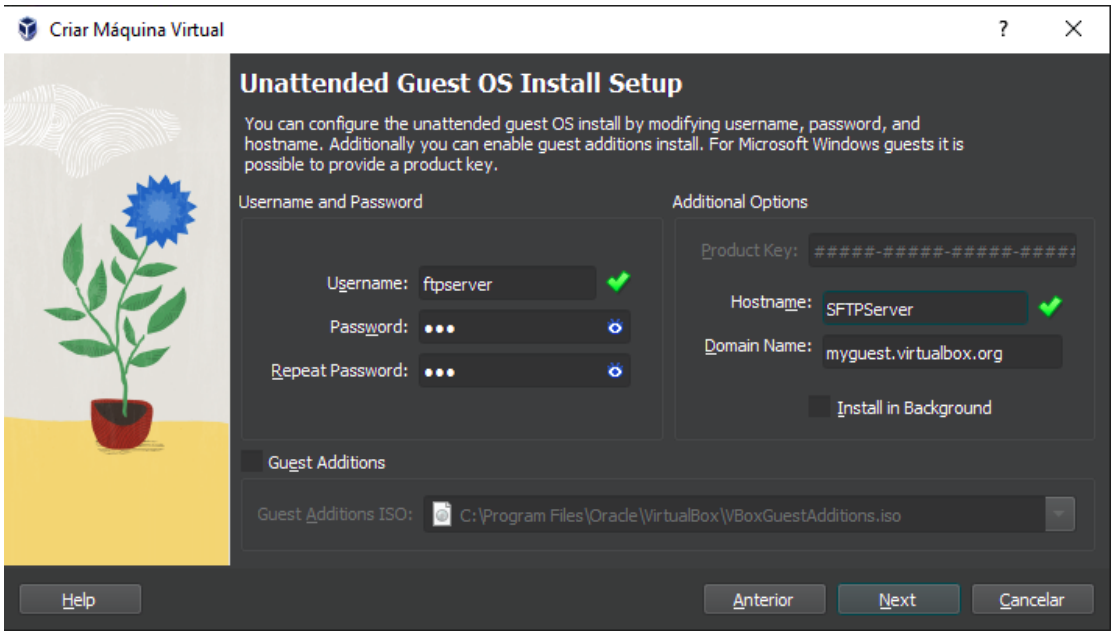


Figura 49 - Escolha de utilizador e password

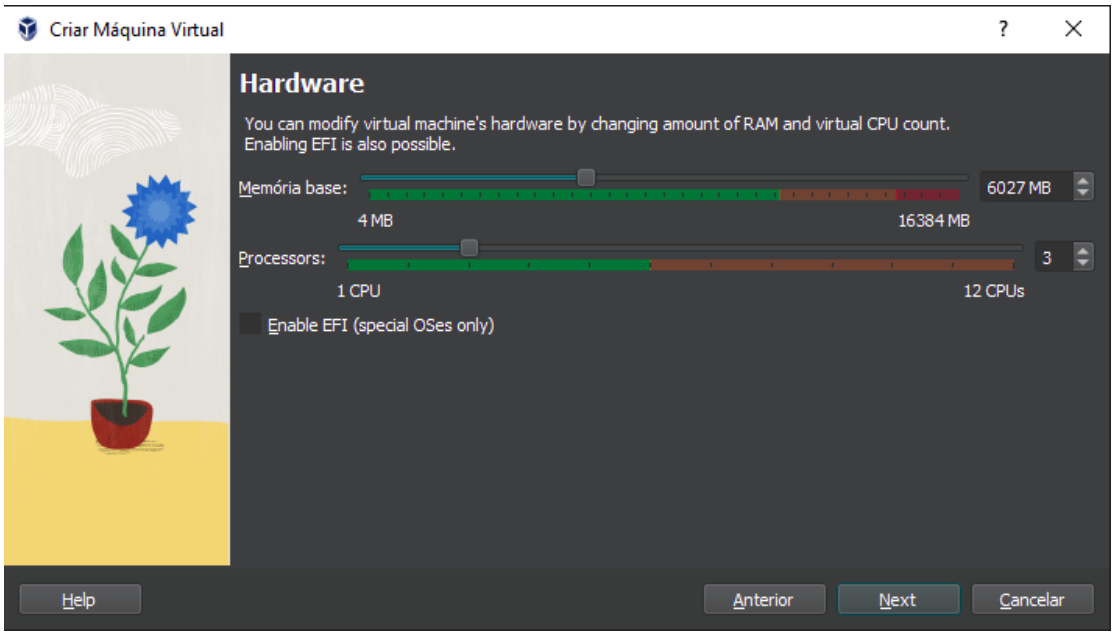


Figura 50 - Hardware dedicado à VM

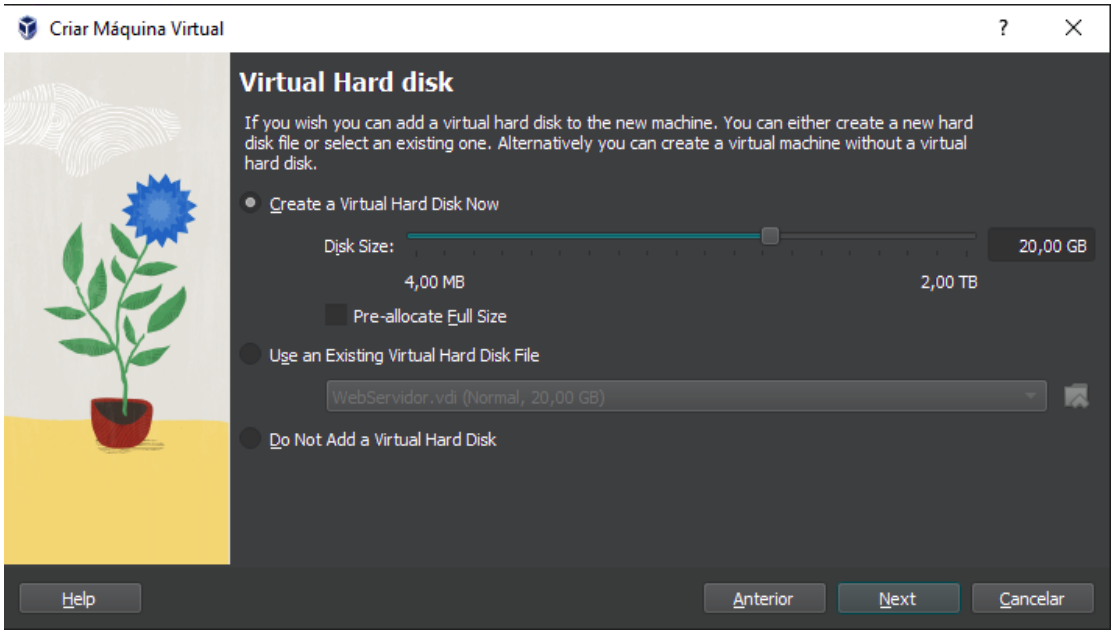


Figura 51 - Tamanho do disco da VM

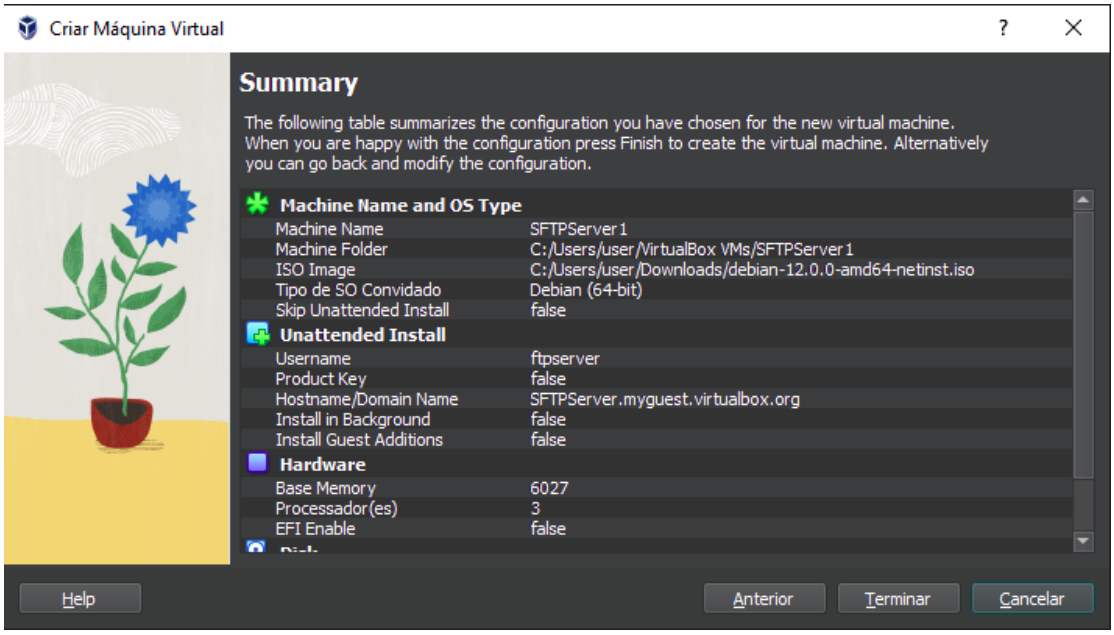


Figura 52 - Sumário da configuração

Anexo 2 - Configuração servidor SFTP

```
sudo apt-get install openssh-server
sudo adduser sftpuser
sudo nano /etc/ssh/sshd_config
```

A Figura seguinte mostra o conteúdo do ficheiro da configuração do serviço SSH com a configuração feita, de modo que o servidor SFTP funcione como previsto.

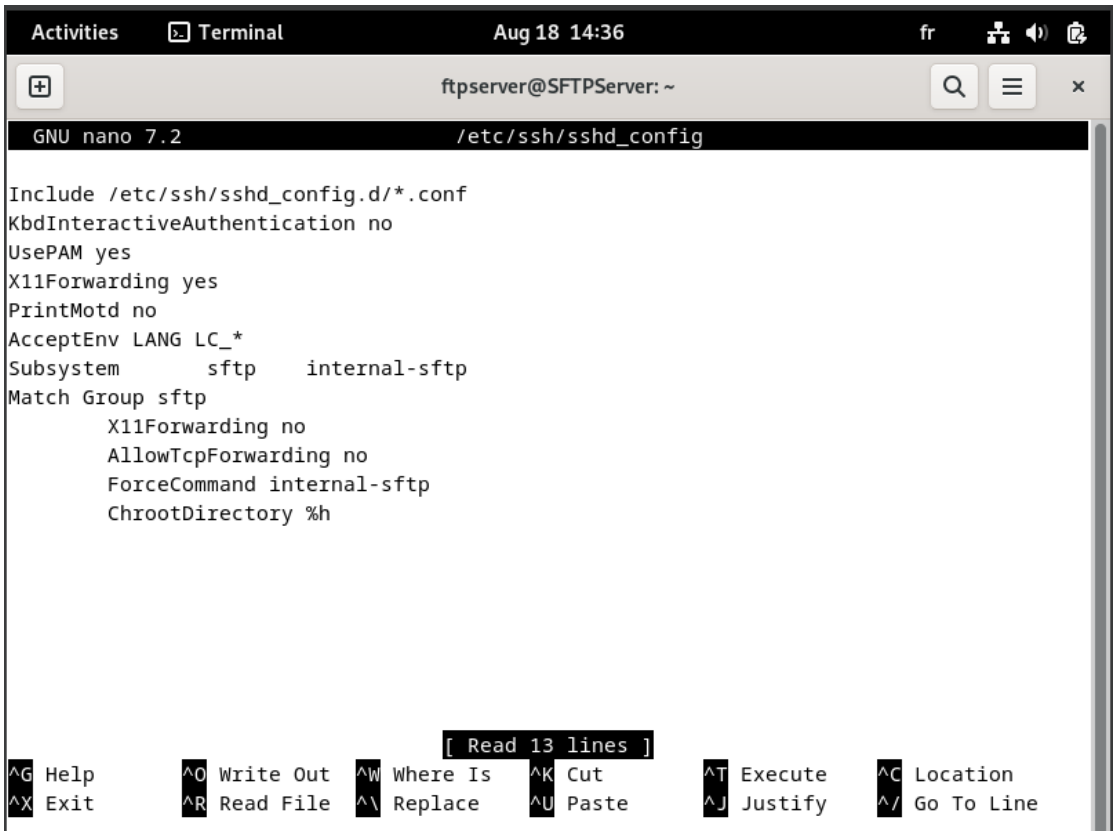


Figura 53 - Conteúdo do ficheiro sshd_config

```
sudo chown root:root /home/sftpuser
sudo chmod 755 /home/sftpuser
sudo mkdir /home/sftpuser/files
sudo chown sftpuser:sftpuser /home/sftpuser/files
```

Anexo 3 - Código fonte do servidor Web

Como referenciado no texto o código não foi desenvolvido durante o estágio, tendo sido desenvolvido anteriormente fora do contexto de estágio. Mesmo assim, foi utilizado de maneira a não ficar simplesmente com a página predefinida quando feita a instalação do Apache2. Este mesmo código encontra-se num repositório GitHub com o seguinte link:

https://github.com/luisantoniio1998/luisantoniio1998.github.io

O seu resultado também se encontra online no seguinte link: https://luisantoniio1998.github.io/

Anexo 4 - Configuração do servidor e-mail

```
wget
https://github.com/roundcube/roundcubemail/releases/download/1.6.0/roundcube
mail-1.6.0-complete.tar.gz
tar xvf roundcubemail-1.6.0-complete.tar.gz
sudo mkdir -p /var/www/
sudo mv roundcubemail-1.6.0 /var/www/roundcube
cd /var/www/roundcube
sudo chown www-data:www-data temp/ logs/ -R
sudo apt install software-properties-common
sudo add-apt-repository ppa:ondrej/php
sudo apt update
sudo apt install php-net-ldap2 php-net-ldap3 php-imagick php8.2-fpm php8.2-
common php8.2-gd php8.2-imap php8.2-mysql php8.2-curl php8.2-zip php8.2-xml
php8.2-mbstring php8.2-bz2 php8.2-intl php8.2-gmp php8.2-redis
sudo mysql -u root
CREATE DATABASE roundcube DEFAULT CHARACTER SET utf8 COLLATE
utf8_general_ci;
CREATE USER roundcube@localhost IDENTIFIED BY 'password'; GRANT ALL
PRIVILEGES ON roundcube.* TO roundcube@localhost;
flush privileges;
exit;
sudo nano /etc/apache2/sites-available/roundcube.conf
```

A figura seguinte demonstra o conteúdo do ficheiro criado e alterado para que seja possível o funcionamento do software Roundcube.

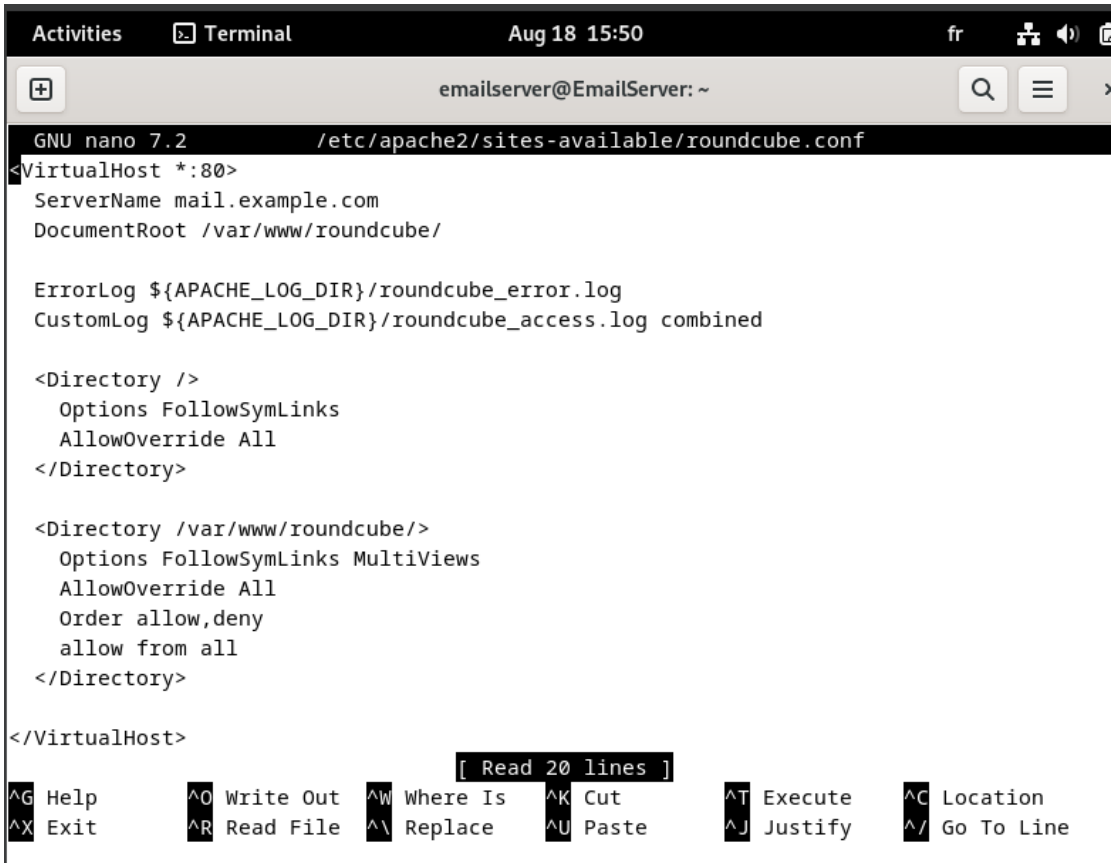


Figura 54 - Conteúdo do ficheiro roundcube.conf

```
sudo a2ensite roundcube.conf
sudo systemctl reload apache2
```

Após esta configuração inicial é necessário aceder ao URL: http://localhost/roundcube/installer de maneira a aceder ao instalador deste software, durante os processos o instalador vai verificar os requisitos para executar o Roundcube , incluindo a versão do PHP, extensões do PHP e outras definições de servidor. De seguida é necessário efetuar a configuração da ligação à base de dados, definições IMAP e SMTP, e outras preferências. Após estas configurações o instalador vai gerar ficheiros de configuração baseados nos inputs feitos.

Anexo 5 - Configuração da FortiGate 61E

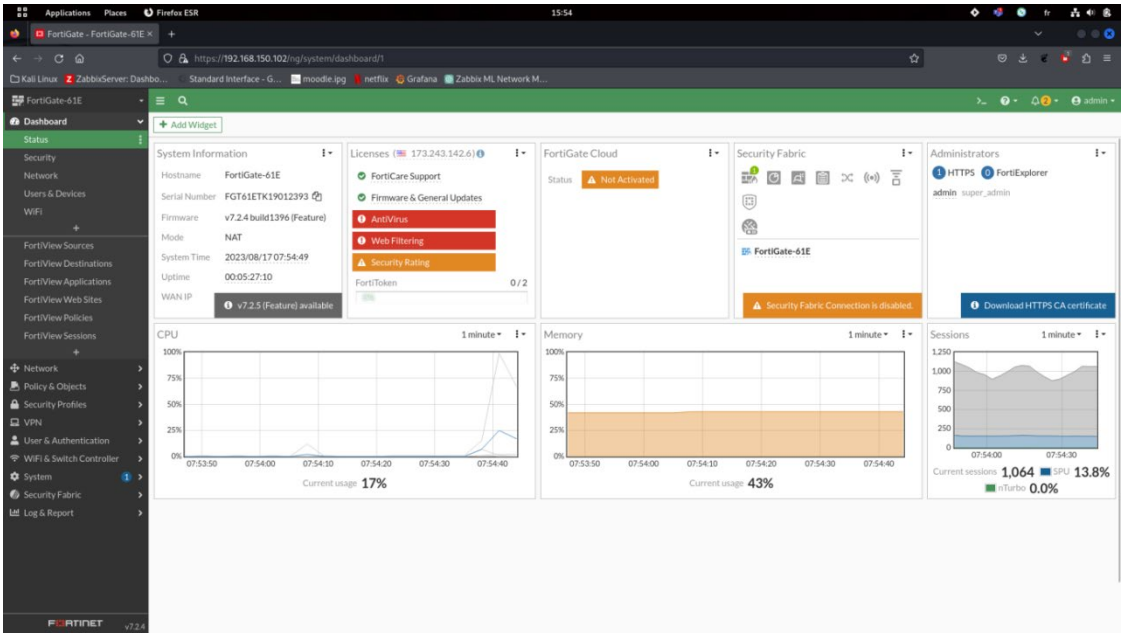


Figura 55 - Dashboard da FortiGate

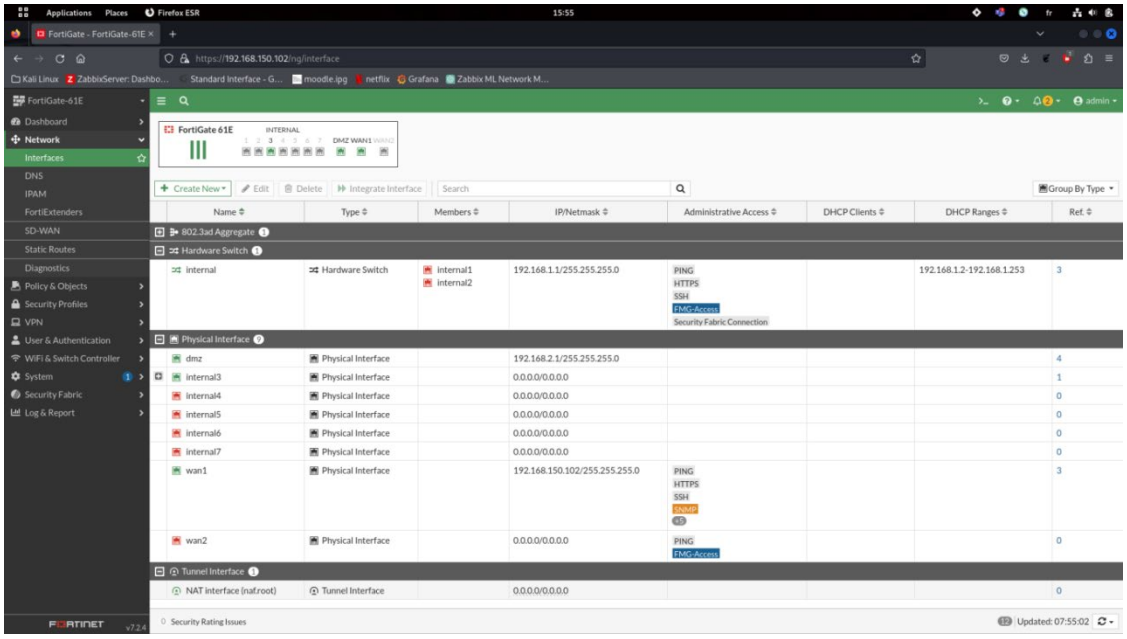


Figura 56 - Interfaces da FortiGate

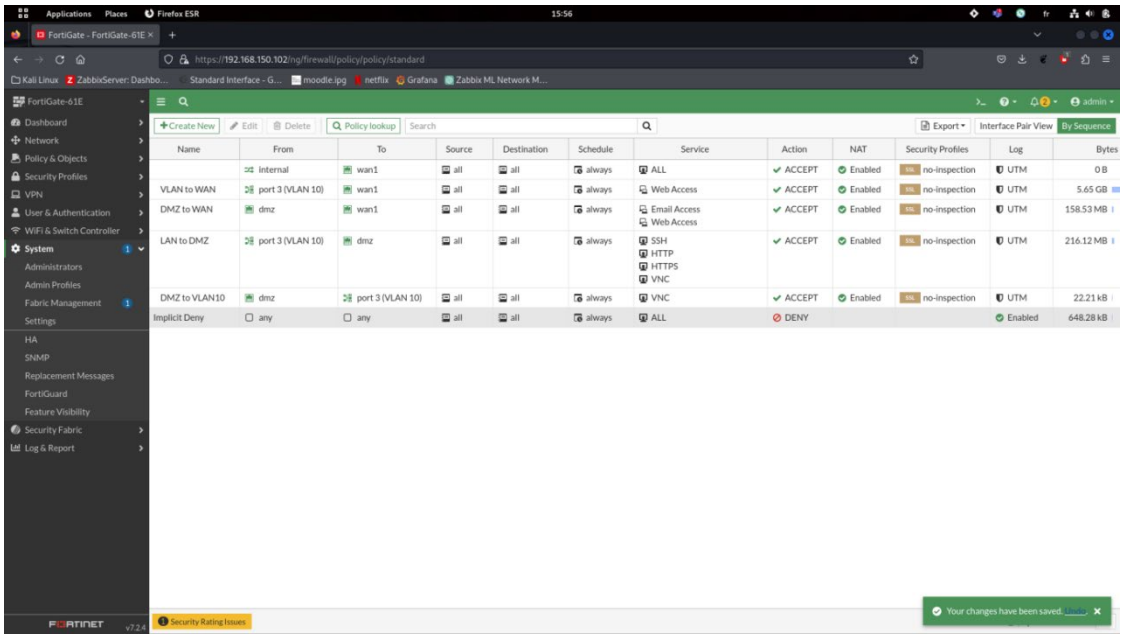


Figura 57 - Politicas implementadas

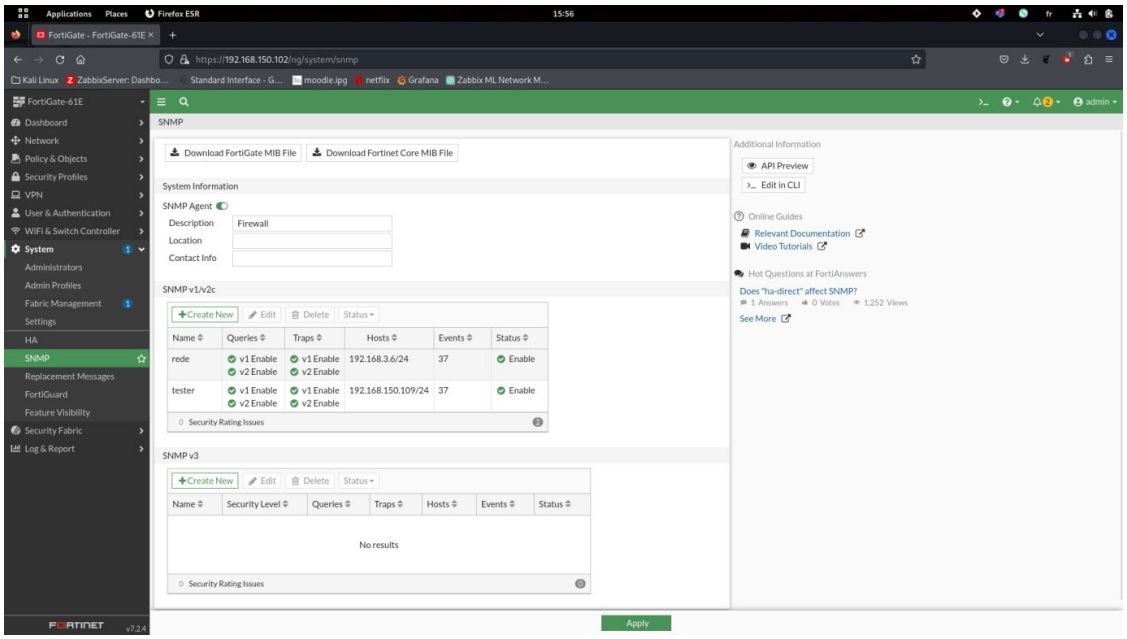


Figura 58 - Configuração do SNMP

Anexo 6 - Configuração do FortiSwitch 108E-POE

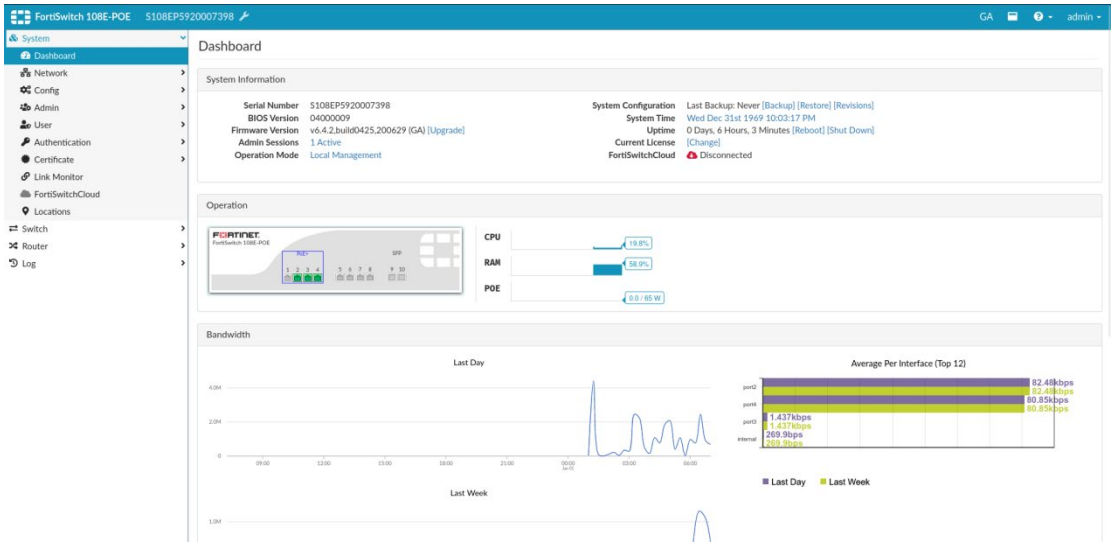


Figura 59 - Dashboard do FortiSwitch

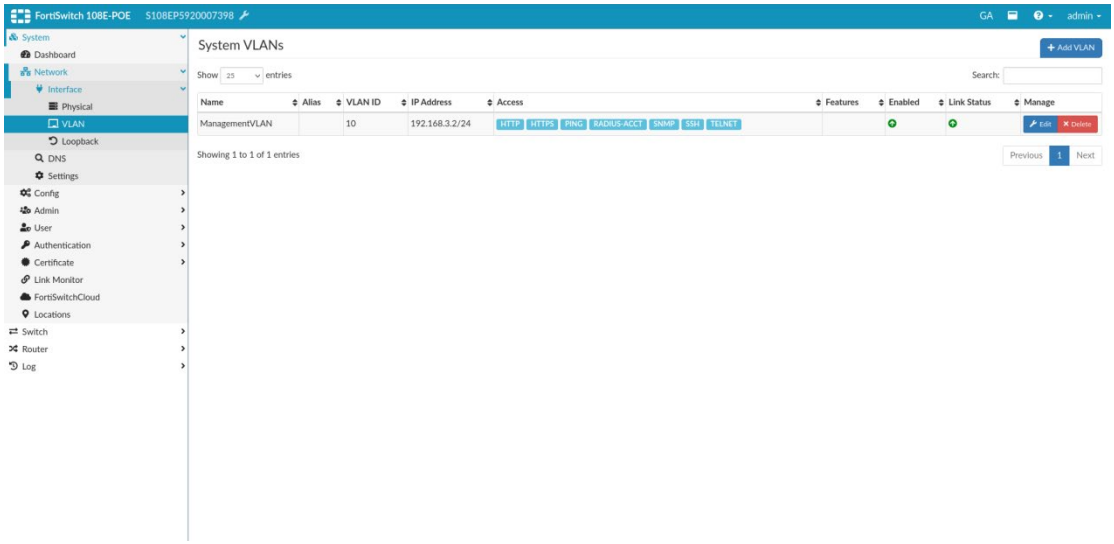


Figura 60 - VLAN´s criadas

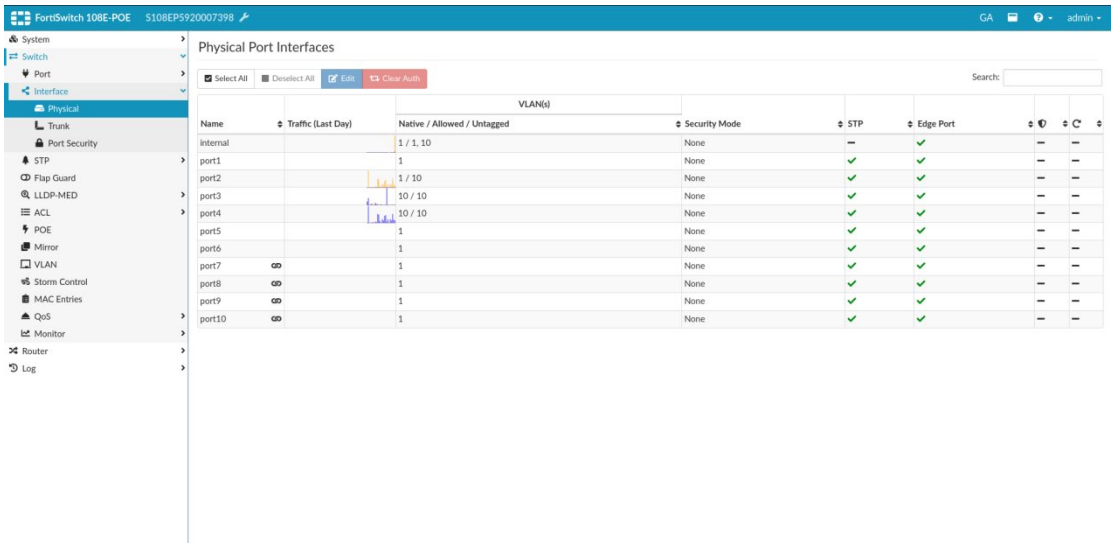


Figura 61 - Interfaces do FortiSwitch

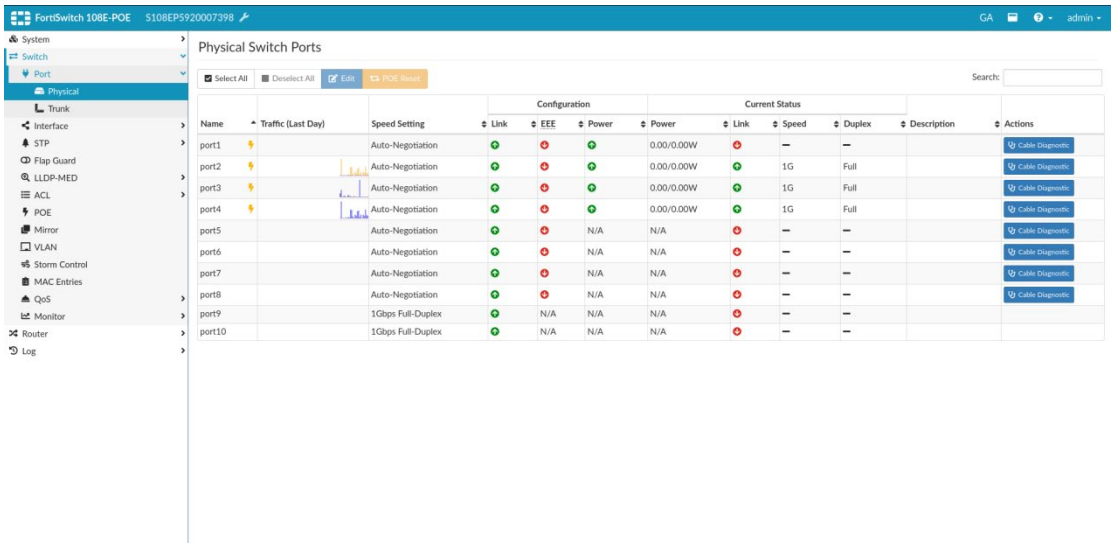


Figura 62 - Portas fisicas do FortiSwitch

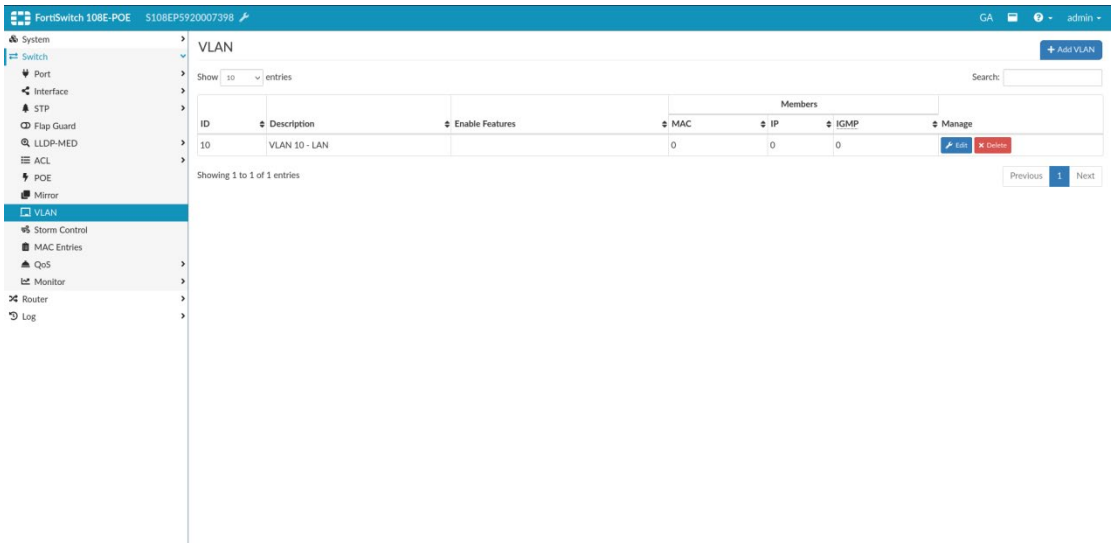


Figura 63 - VLAN´s em Switch

Anexo 7 - Configuração de um agente Zabbix usando SNMPv2 (DMZ - Windows host)

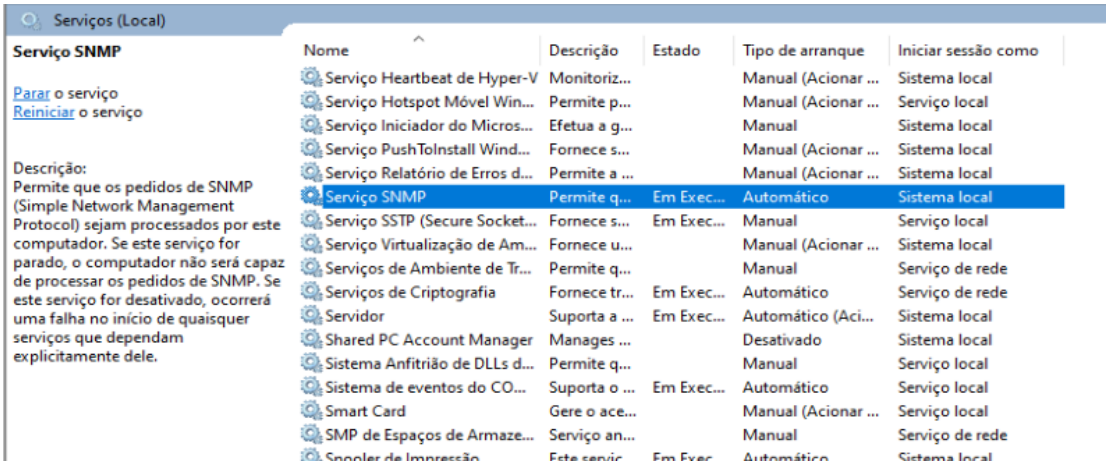


Figura 64 - Ativação do serviço SNMP no agente

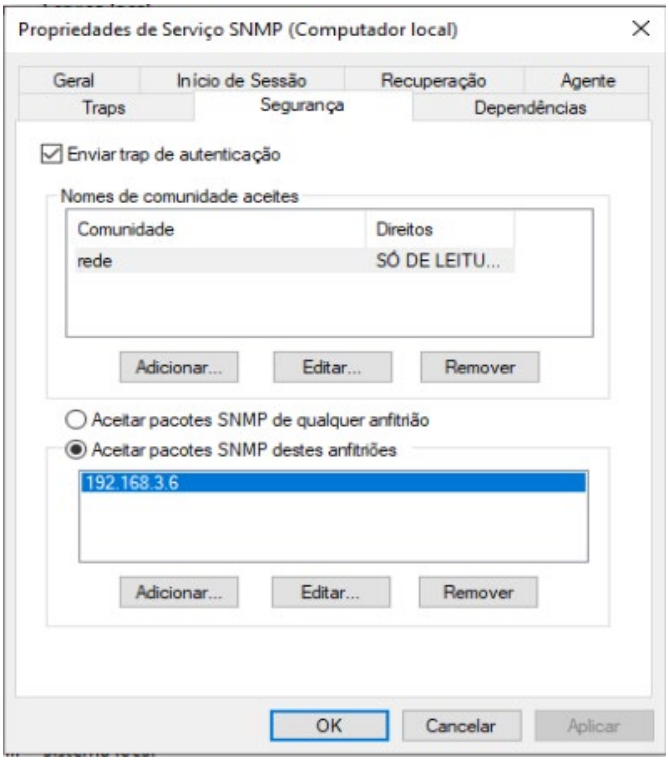


Figura 65 - Configuração do serviço SNMP no agente

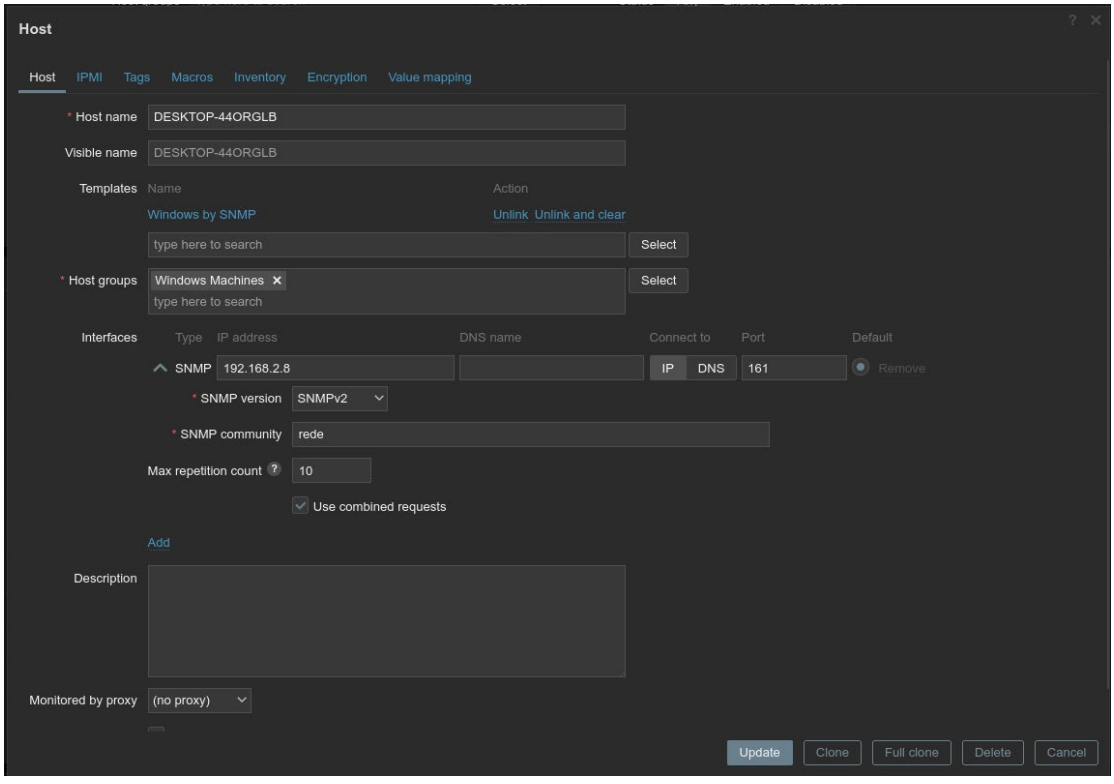


Figura 66 - Configuração do host no servidor Zabbix

Anexo 8 - Configuração de um agente Zabbix usando SNMPv3 (SFTP Server - Linux)

```
sudo apt install snmpd
```

```
sudo nano /etc/snmp/snmpd.conf
```

```
sysLocation Sitting on the Dock of the Bay
sysContact Me <me@example.org>
sysServices 72
master agentx
agentaddress udp:192.168.2.3:161
view systemonly included .1.3.6.1.2.1.1
view systemonly included .1.3.6.1.2.1.25.1
rocommunity public default -V systemonly
rocommunity6 public default -V systemonly
createuser myuser SHA-512 myauthphrase AES myprivphrase
rouser myuser authpriv -V systemonly
includeDir /etc/snmp/snmpd.conf.d
```

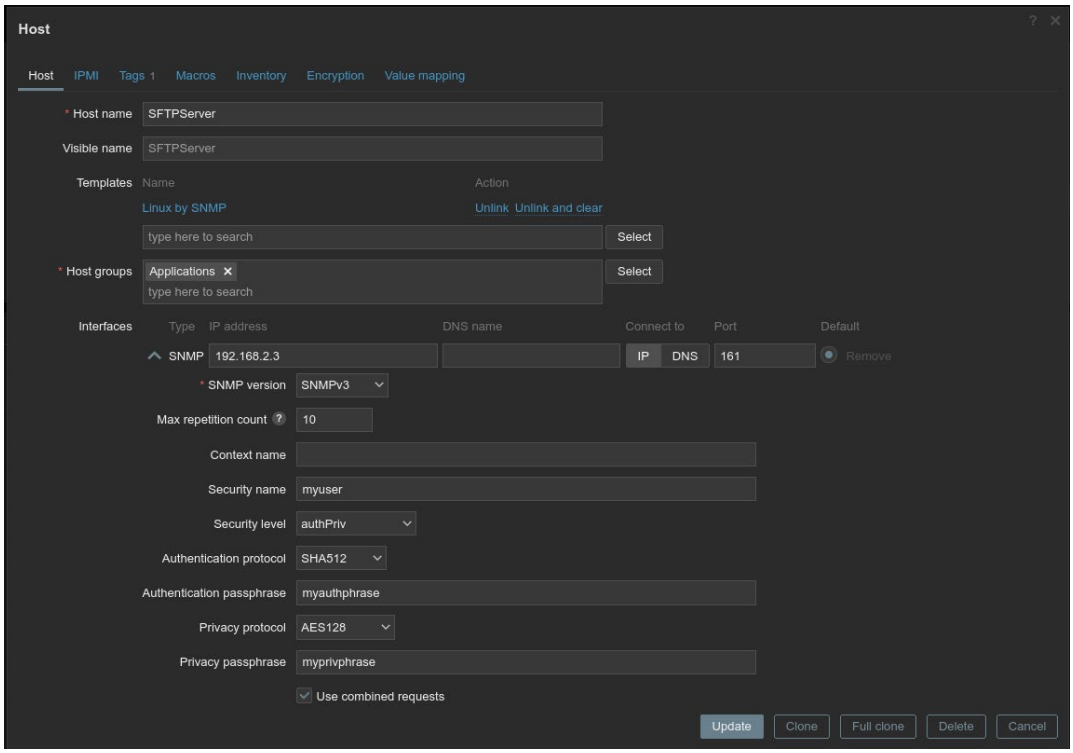


Figura 67 - Configuração de um Zabbix host com SNMPv3

Anexo 9 - Configuração de um agente Zabbix versão 2 (Web Server - Linux)

```
wget https://repo.zabbix.com/zabbix/6.4/debian/pool/main/z/zabbix-release/zabbix-release_6.4-1+debian12_all.deb

dpkg -i zabbix-release_6.4-1+debian12_all.deb

sudo apt update

sudo apt install zabbix-agent2 zabbix-agent2-plugin-*

sudo nano /etc/zabbix/zabbix_agent2.conf


LogFile=/var/log/zabbix/zabbix_agent2.log

LogFileSize=0

Server=192.168.2.1

ServerActive=192.168.2.1

Hostname=SFTPServer

PersistentBufferPeriod=1h

Include=/etc/zabbix/zabbix_agent2.d/*.conf

Include=/usr/local/etc/zabbix_agent2.conf.d/*.conf

ControlSocket=/tmp/agent.sock

Include=./zabbix_agent2.d/plugins.d/*.conf
```

De referir que o conteúdo deste ficheiro não é mostrado na totalidade devido à sua extensividade. Por essa mesma razão mostrei apenas os parâmetros que necessitam de ser alterados de acordo com o desejado de maneira a obter-se o correto funcionamento da comunicação entre o host e o servidor.

```
sudo systemctl restart zabbix-agent2

sudo systemctl enable zabbix-agent2
```

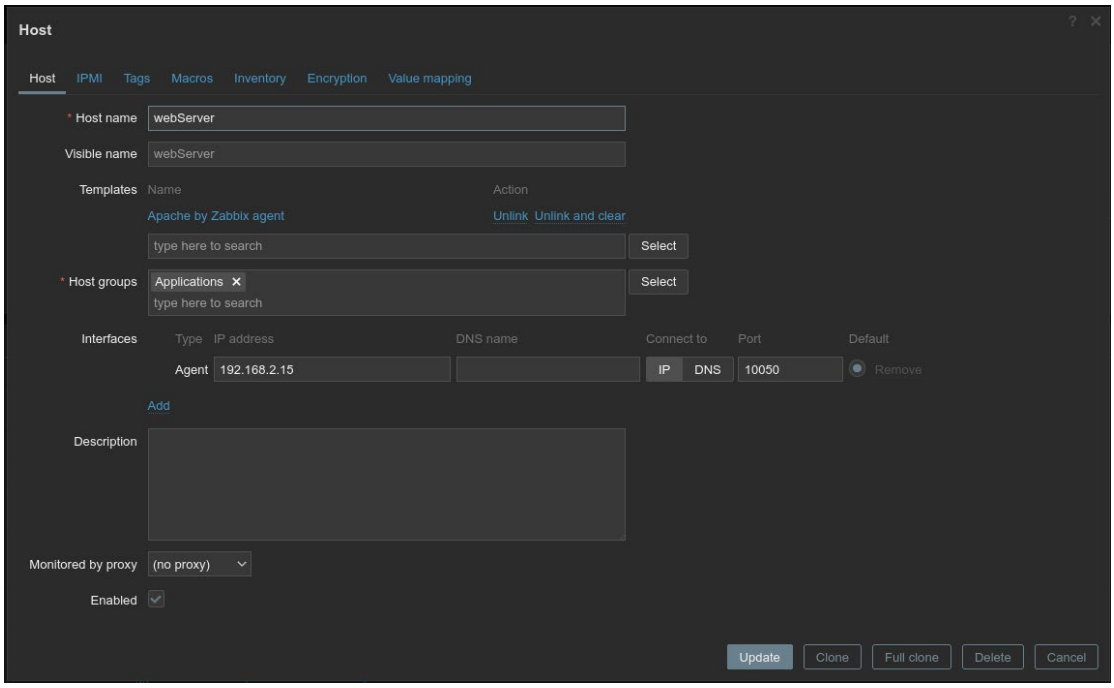


Figura 68 - Configuração de um Zabbix host usando Zabbix-Agent 2

Anexo 10 - Configuração de um agente Zabbix usando SNMPv2 (FortiGate 61E)

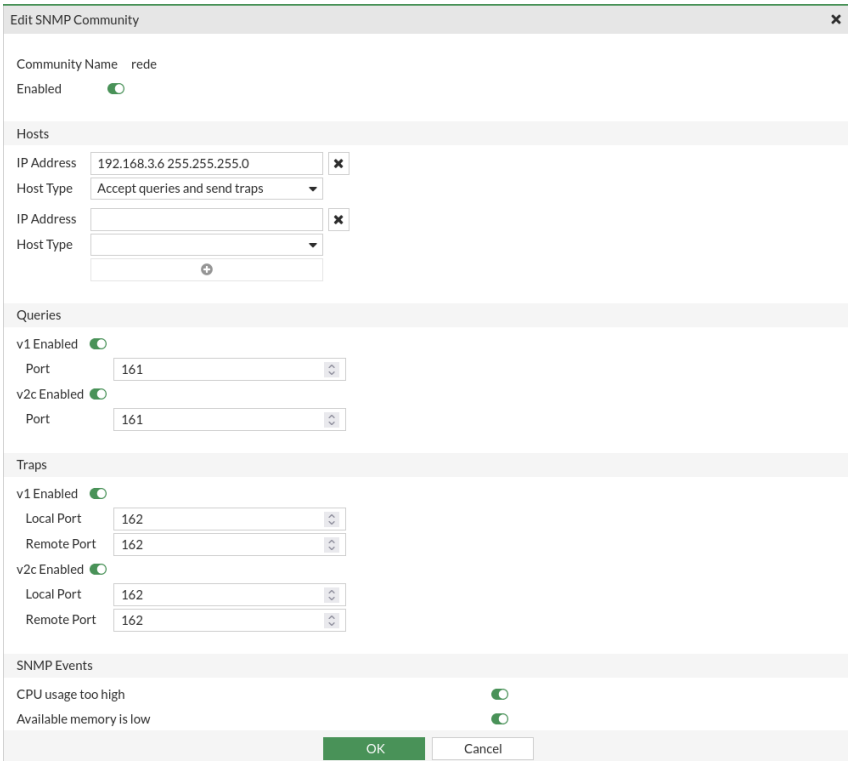


Figura 69 - Configuração SNMPv2 na FortiGate

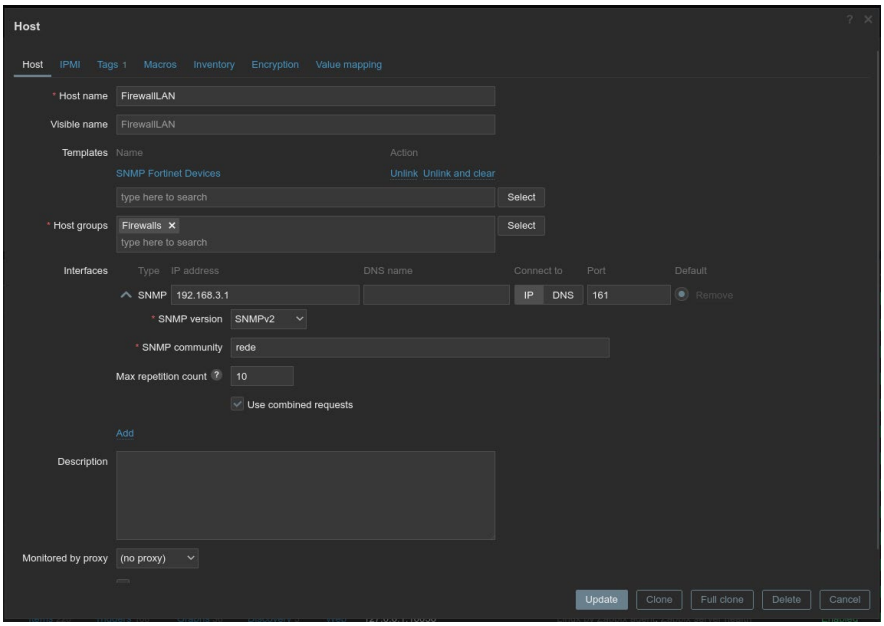


Figura 70 - Criação de um Zabbix host usando SNMPv2

Anexo 11 - Configuração de um agente Zabbix usando SNMPv3 (FortiSwitch 108E-POE)

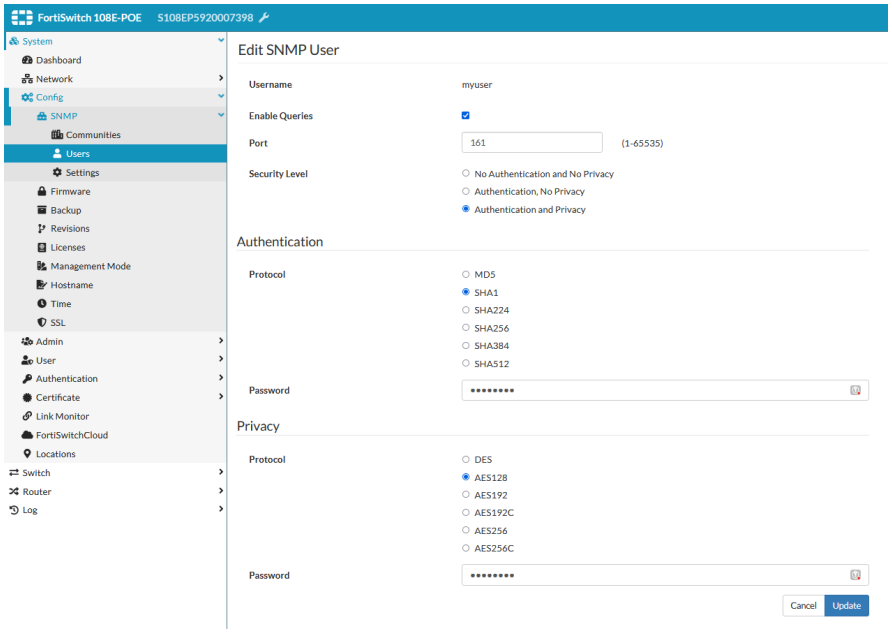


Figura 71 - Configuração do SNMPv3 no FortiSwitch

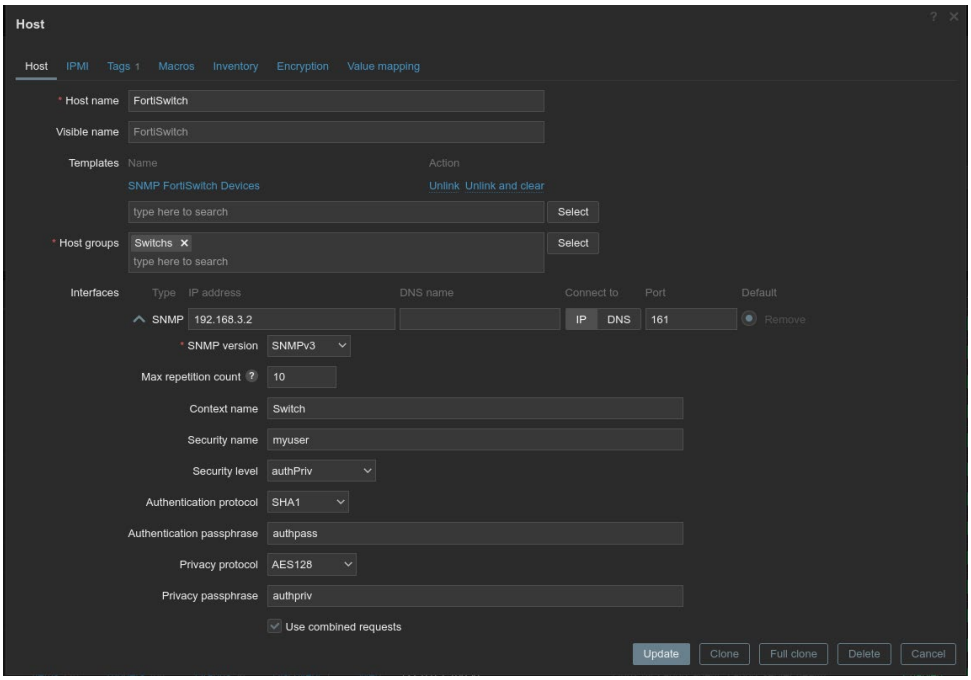


Figura 72 - Criação do host relativo ao FortiSwitch com SNMPv3

Anexo 12 - Importação template Zabbix da comunidade

Dentro do ecossistema oficial do Zabbix, existe um repositório denominado "community-templates" que agrega uma variedade de templates criados pela comunidade. Para atender às necessidades específicas de monitorização dos dispositivos alvo, selecionamos um template apropriado deste repositório. A referência completa para este repositório pode ser encontrada no seguinte endereço URL: https://github.com/zabbix/community-templates.

Uma vez selecionado o template apropriado, a sua integração no ambiente frontend do Zabbix é um processo direto. Dentro da interface de utilizador, deve-se navegar até ao menu "Data Collection" e optar pela opção 'Templates'. Em seguida, inicia-se a operação de importação do template escolhido. Posteriormente, é fundamental associar esse template ao host específico em questão. Ressalta-se que, dada a natureza intuitiva da interface do Zabbix, este procedimento é simplificado e otimizado para uma experiência amigável ao utilizador.Precisamente por esta razão, não se julgou necessário complementar a descrição com ilustrações do procedimento.

Anexo 13 - Configuração de itens

13.1. Configuração de monitorização de base de dados

```
sudo apt install odbc-mariadb unixodbc unixodbc-dev odbcinst
odbcinst -j
unixODBC 2.3.12
DRIVERS............: /etc/odbcinst.ini
SYSTEM DATA SOURCES: /etc/odbc.ini
FILE DATA SOURCES..: /etc/ODBCDataSources
USER DATA SOURCES..: /home/kali/.odbc.ini
SQLULEN Size.......: 8
SQLLEN Size........: 8
SQLSETPOSIROW Size.: 8
```

```
nano /etc/obdc.ini
GNU nano 7.2 /etc/odbc.ini
[book]
Description = MySQL book test database
Driver = MariaDB Unicode
Server = 127.0.0.1
Port = 3306
Database = zabbix
User = zabbix
Password = password
┌──(kali㉿kali)-[~]
└─$ isql -v book
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| echo [string]                         |
| quit                                  |
|                                       |
+---------------------------------------+
SQL>
```

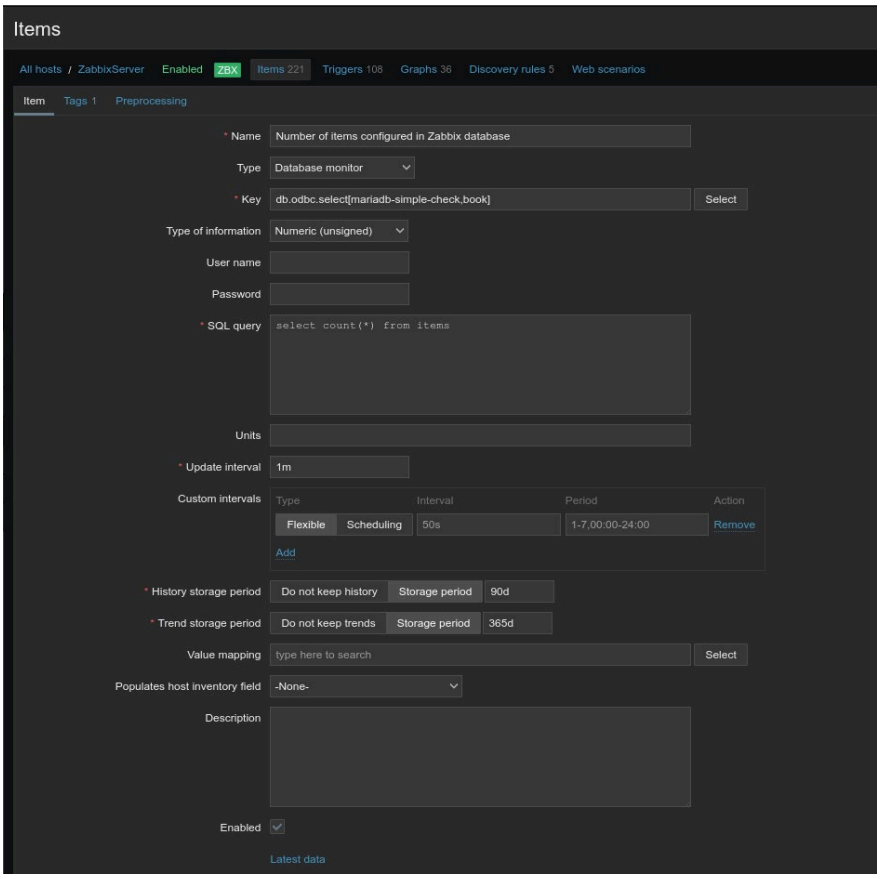


Figura 73 - Criação de um item para monitorizar o serviço da base de dados

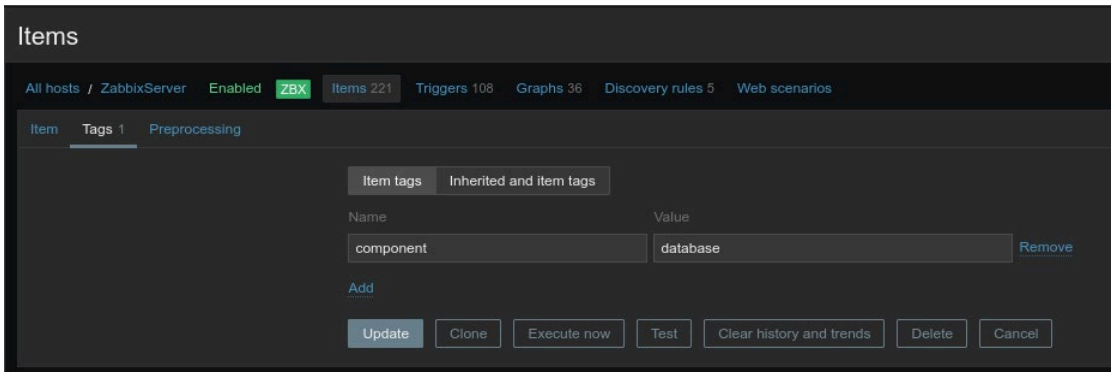


Figura 74 - Tags usadas para o item criado

Figura 75 - Funcionamento do item criado

13.2. Criação de item para mostrar tarefas rotineiras

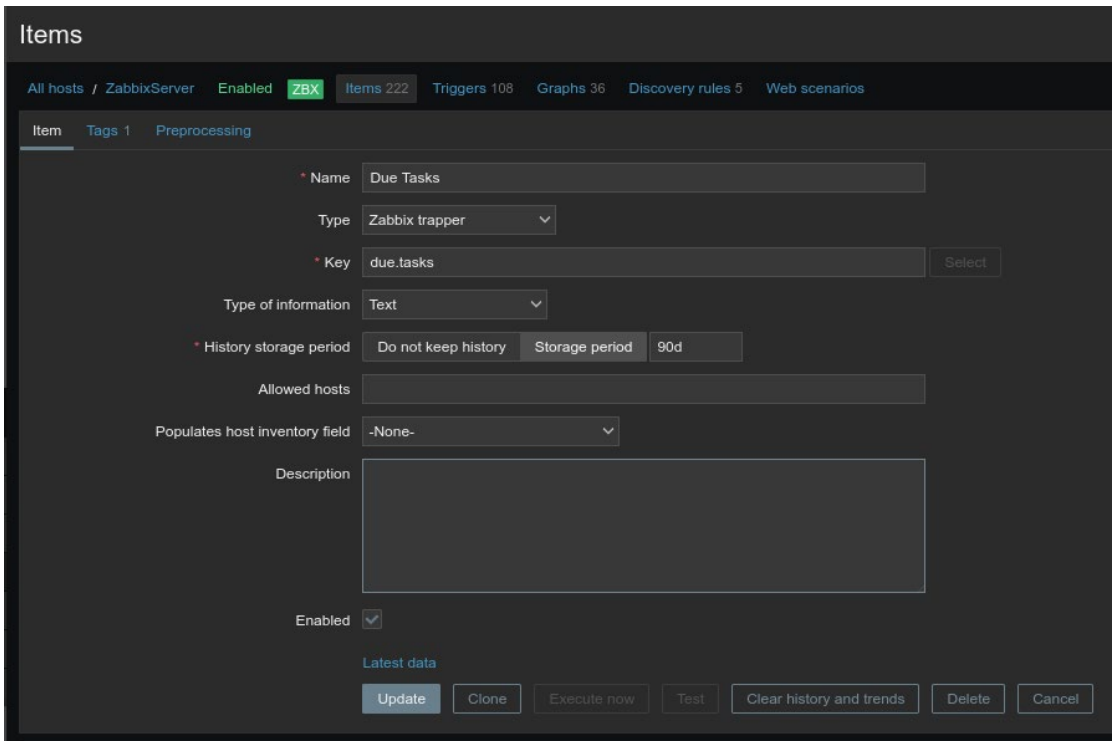


Figura 76 - Criação de um item para mostrar conteúdo de um ficheiro

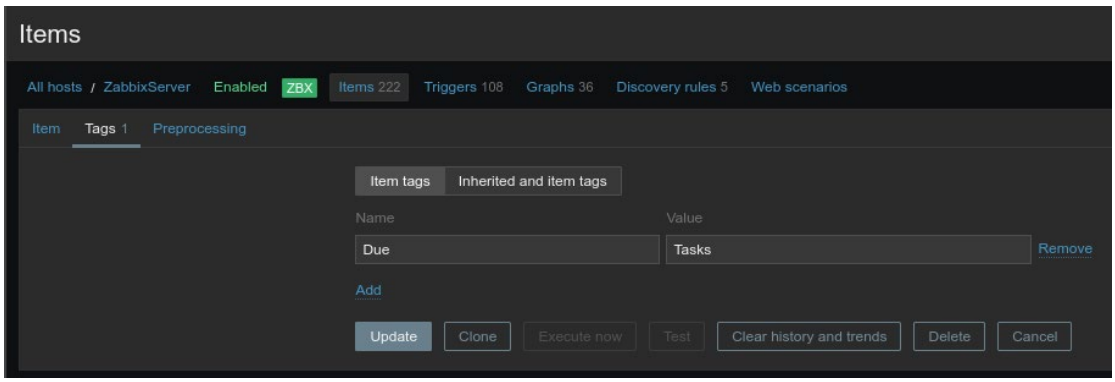


Figura 77 - Tags usadas para o item criado

13.3. Uso da função forecast

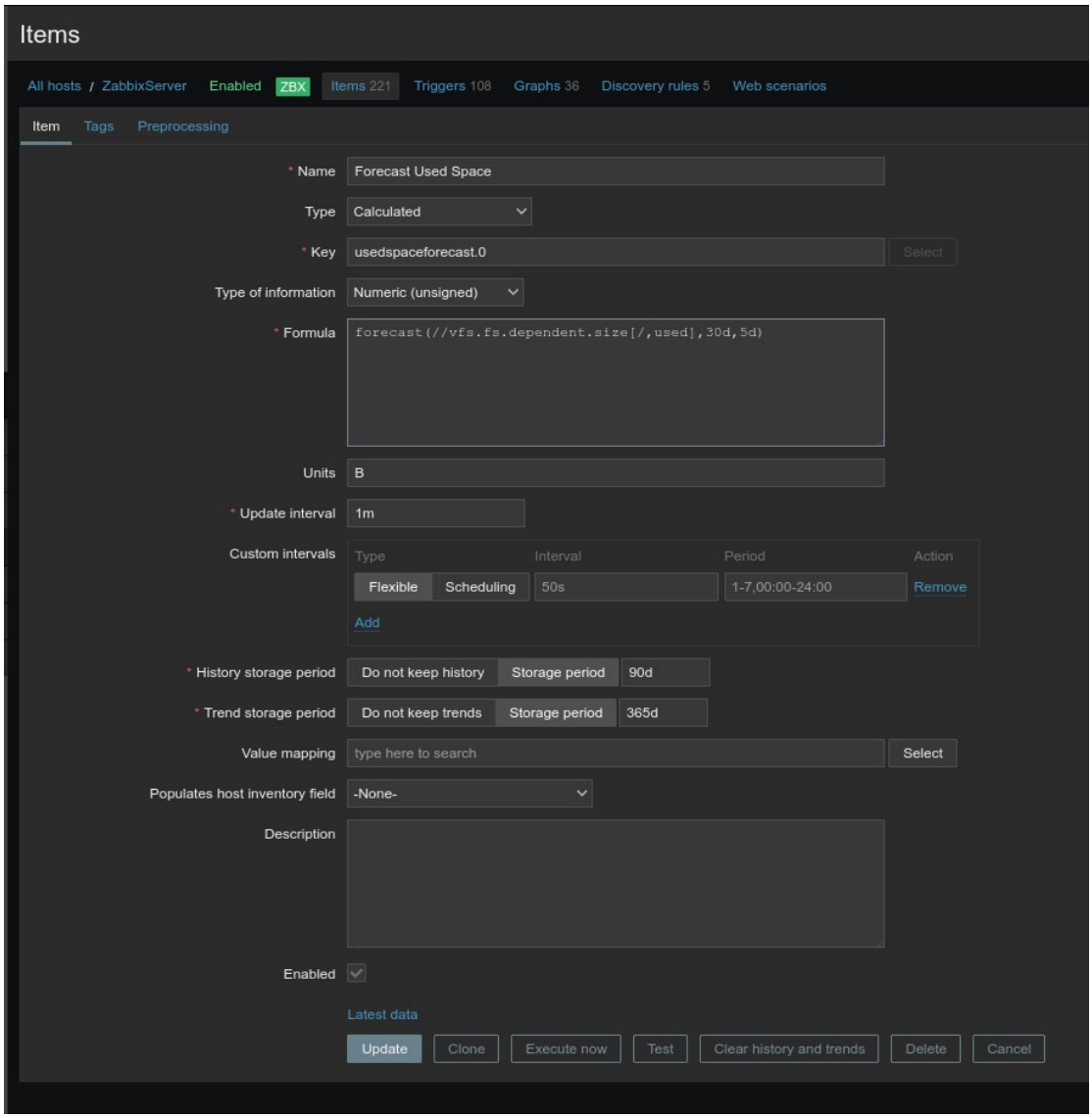


Figura 78 - Criação de um item usando uma função de previsão (forecast)

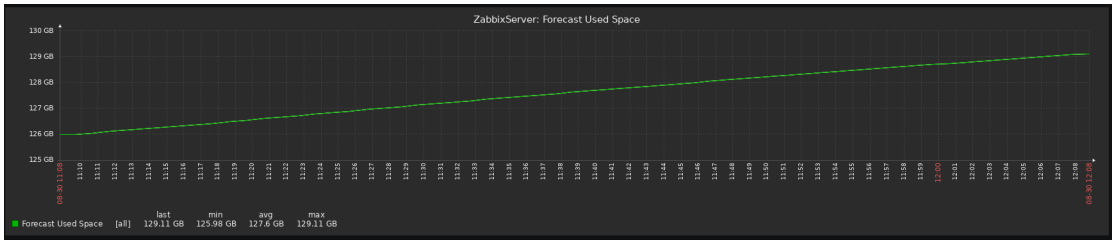


Figura 79 - Gráfico resultante do item criado

13.4. Uso da função timeleft

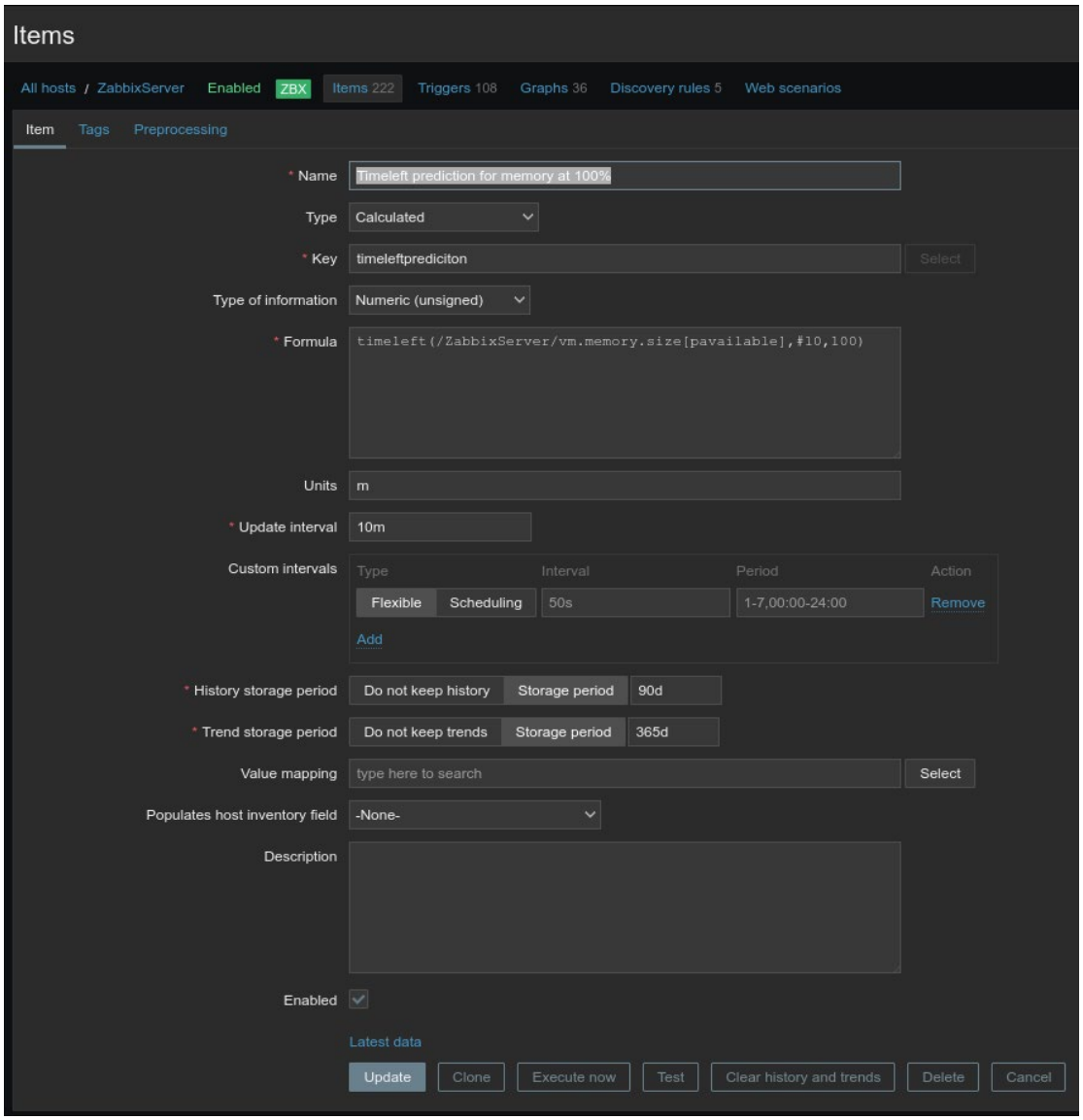


Figura 80 - Ciração de um item usando a função timeleft

Anexo 14 - Configuração de triggers

14.1. Configuração de trigger para demonstrar tarefas rotineiras juntamente com o item em 13.2.

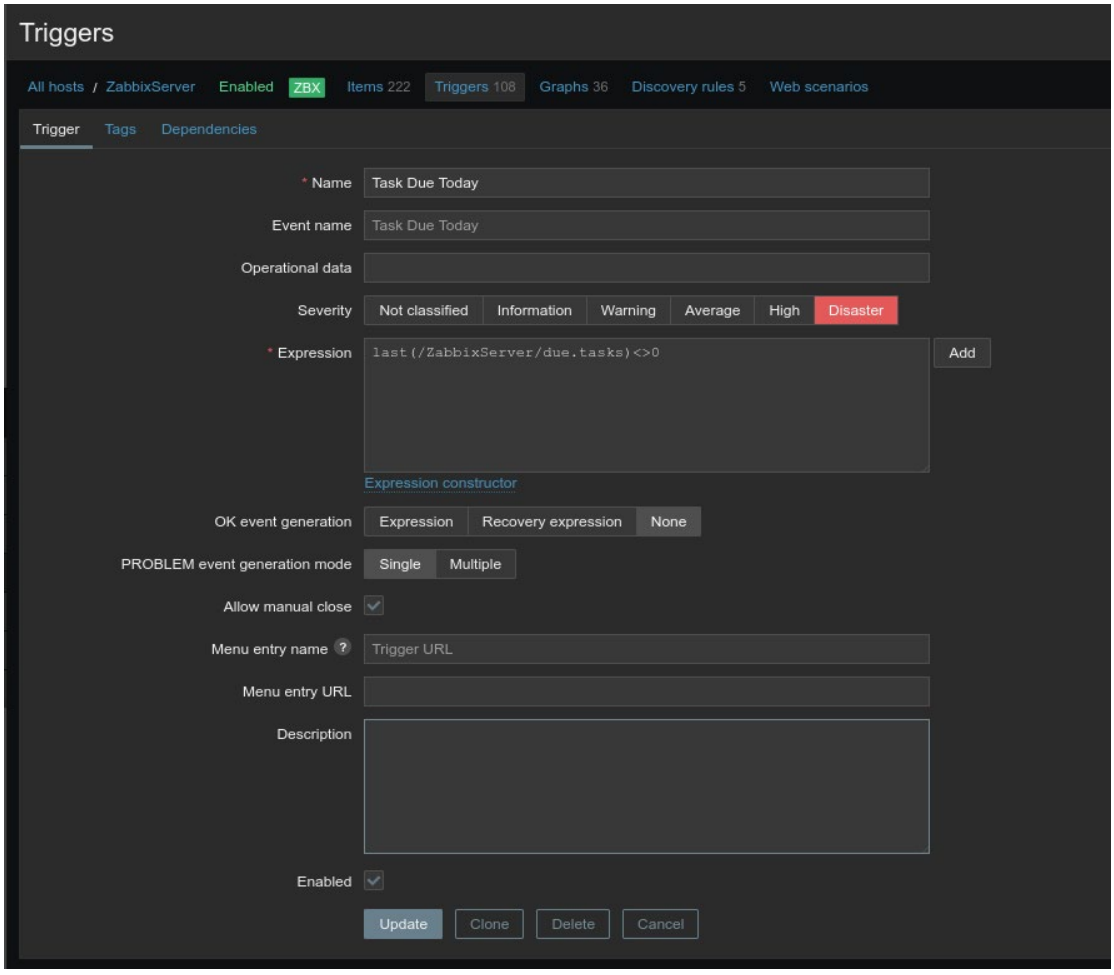


Figura 81 - Criação de um trigger associado ao item em 13.2

Anexo 15 - Configuração e criação de scripts

15.1. Configuração e demonstração script de automação para reiniciar serviço

```
webserver@WebServidor:~$ sudo nano /etc/zabbix/zabbix_agent2.conf
Plugins.SystemRun.LogRemoteCommands=0
webserver@WebServidor:~$ sudo nano /etc/sudoers
root ALL=(ALL:ALL) ALL
zabbix ALL=(ALL) NOPASSWD: /etc/init.d/rsyslog restart
```

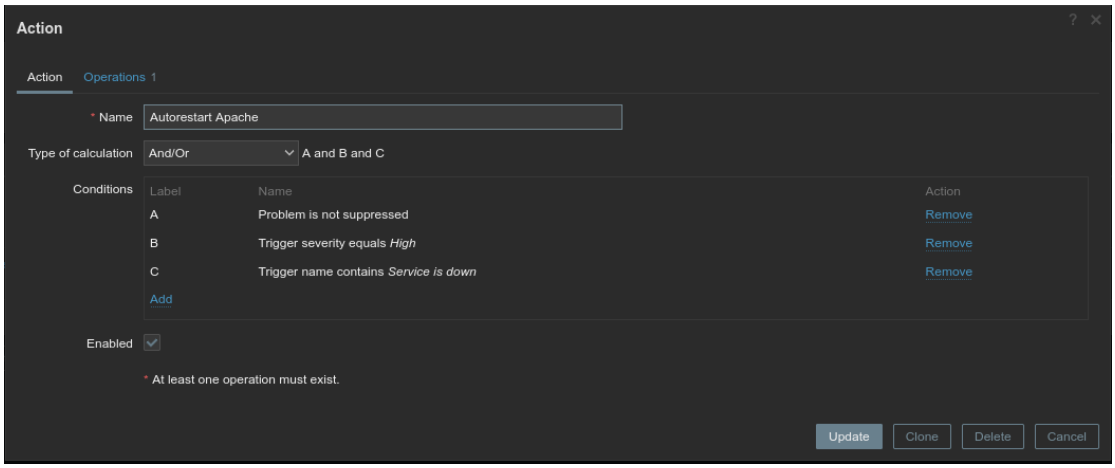


Figura 82 - Configuração de uma ação e das condições que acionam a mesma

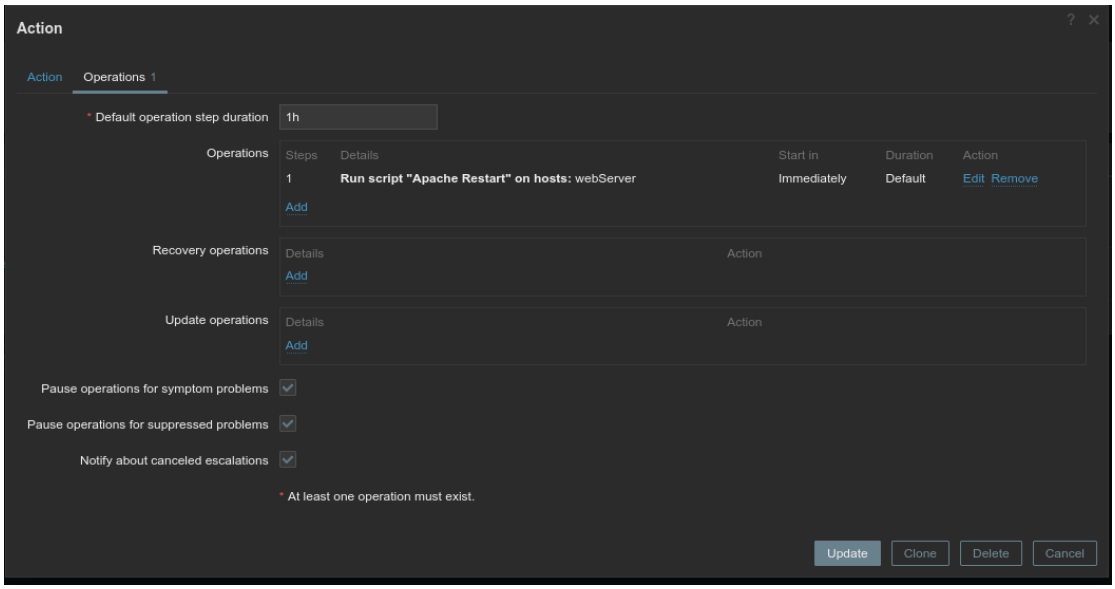


Figura 83 - Operações associadas à ação

```
webserver@WebServidor:~$ sudo systemctl stop apache2
```

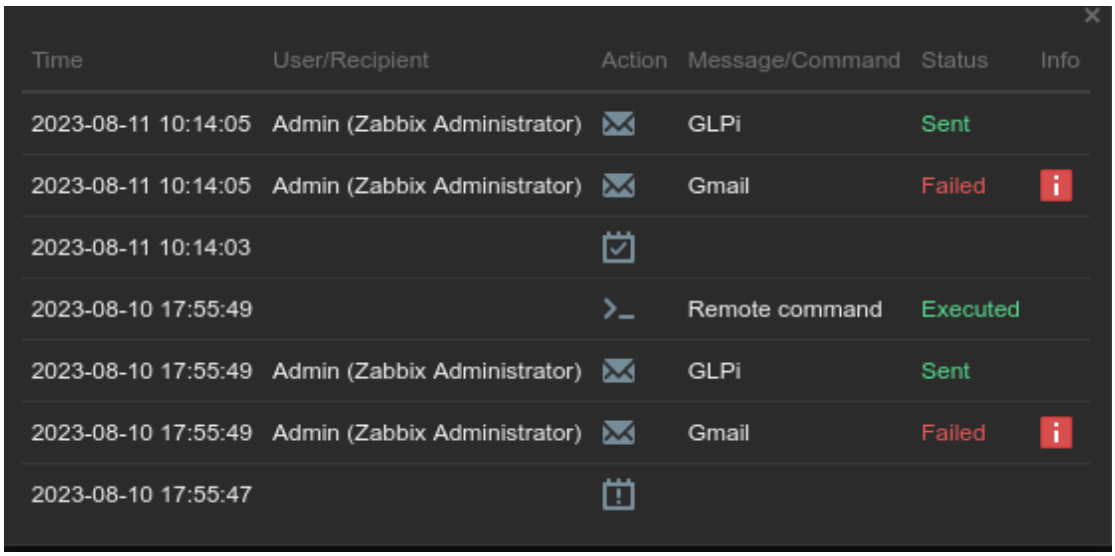


Figura 84 - Demonstração da ação efetuada com sucesso

```
webserver@WebServidor:~$ sudo systemctl status apache2
apache2.service - The Apache HTTP Server
Loaded: loaded (/lib/systemd/system/apache2.service; disabled; preset: dis>
Active: active (running)
Docs: https://httpd.apache.org/docs/2.4/
Process: 23628 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S>
Main PID: 23722 (apache2)
Tasks: 12 (limit: 18890)
Memory: 134.5M
CPU: 11.515s
```

Anexo 16 - Integração com GLPi

16.1. Instalação do GLPi

```
cd /var/www/html
sudo wget https://github.com/glpi-
project/glpi/releases/download/10.0.9/glpi-10.0.9.tgz
sudo tar zxvf glpi-10.0.9.tgz
sudo chown -R www-data:www-data glpi/
sudo chmod -R 755 glpi/
sudo mysql -u root -p
password
CREATE DATABASE glpi;
CREATE USER 'glpiuser'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON glpi.* TO 'glpiuser'@'localhost';
FLUSH PRIVILEGES;
EXIT;
sudo nano /etc/apache2/sites-available/glpi.conf
<VirtualHost *:80>
ServerAdmin webmaster@yourdomain.com
DocumentRoot /var/www/html/glpi
ServerName glpi.yourdomain.com
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
<Directory /var/www/html/glpi>
Options +FollowSymLinks
AllowOverride All
Require all granted
</Directory>
</VirtualHost>
```

Após este processo, é necessário aceder ao URL : http://localhost/glpi em que irá mostrar o instalador do software, sendo necessário a configuração da ligação á base de dados criada e escolher algumas preferências como a linguagem, o tema , entre outras.

16.2. Configuração do Zabbix

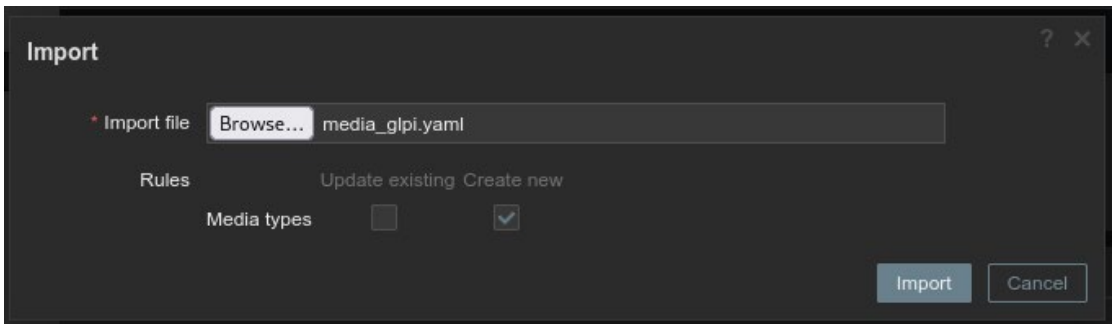


Figura 85 - Ficheiro importado da documentação oficial do Zabbix

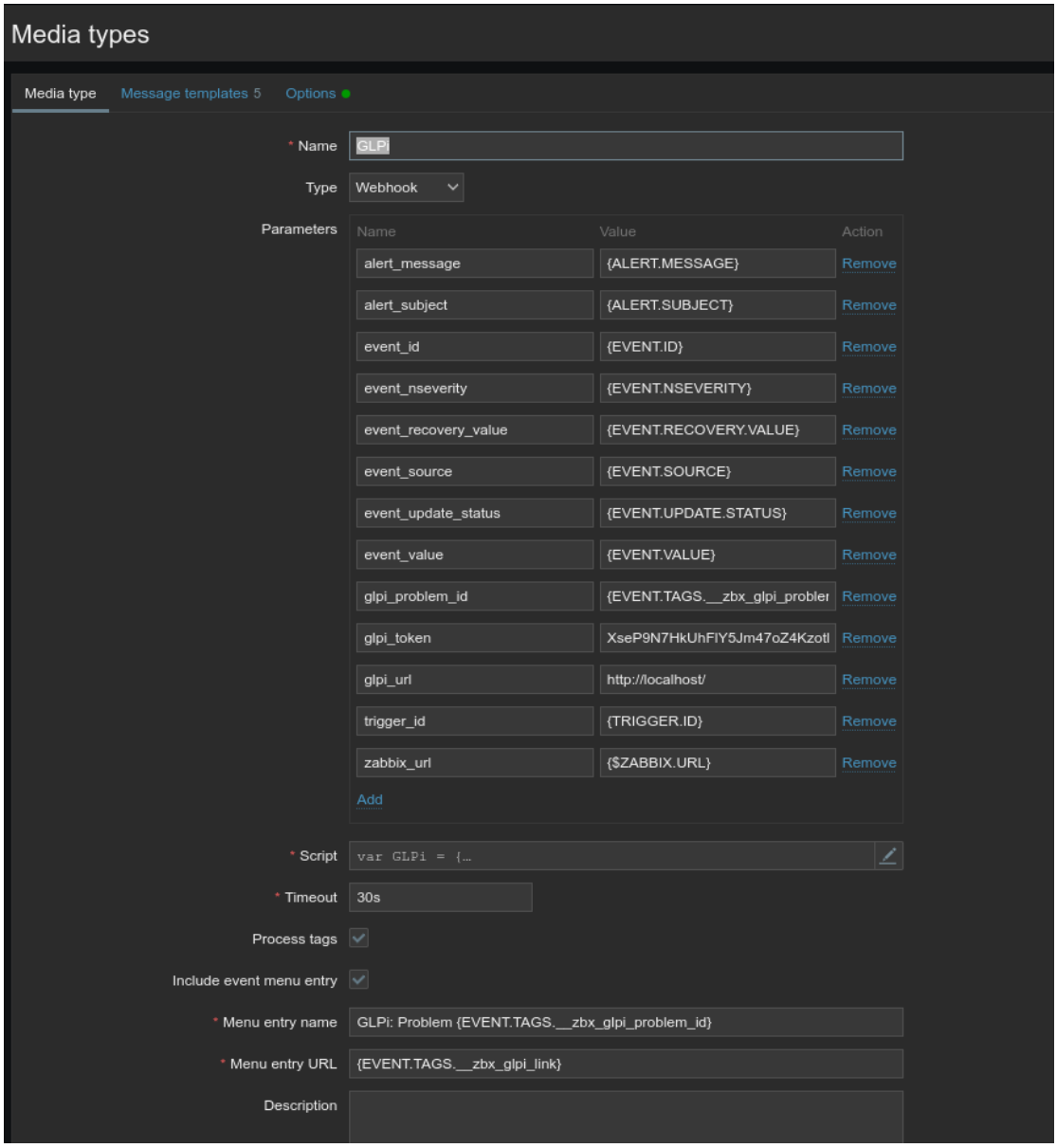


Figura 86 - Configuração dos parametros da integração

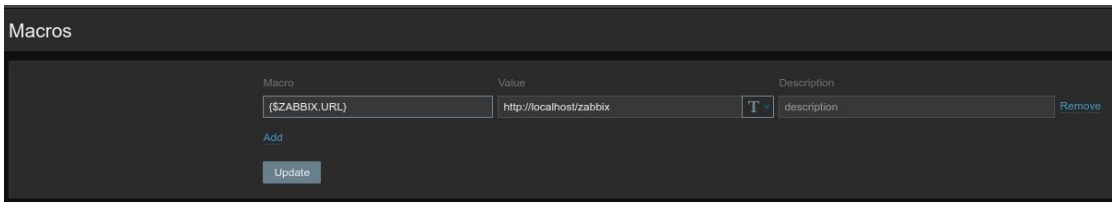


Figura 87 - Criação de um macro global para o correto funcionamento da integração

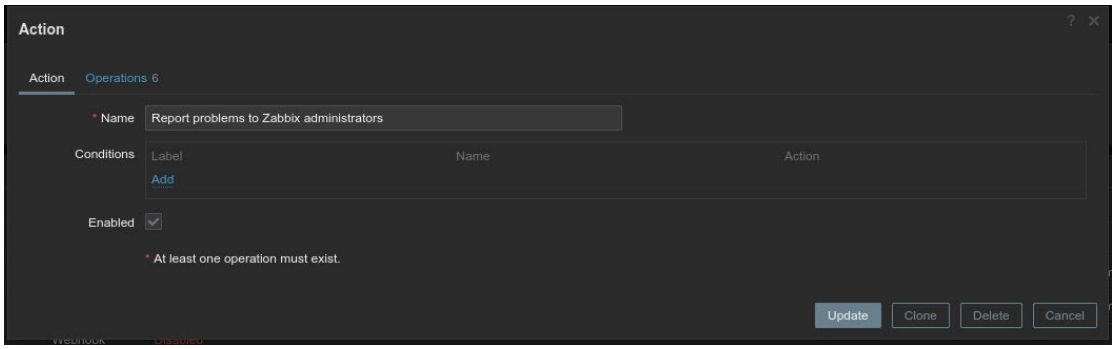


Figura 88 - Criação de uma ação para o correto funcionamento da integração

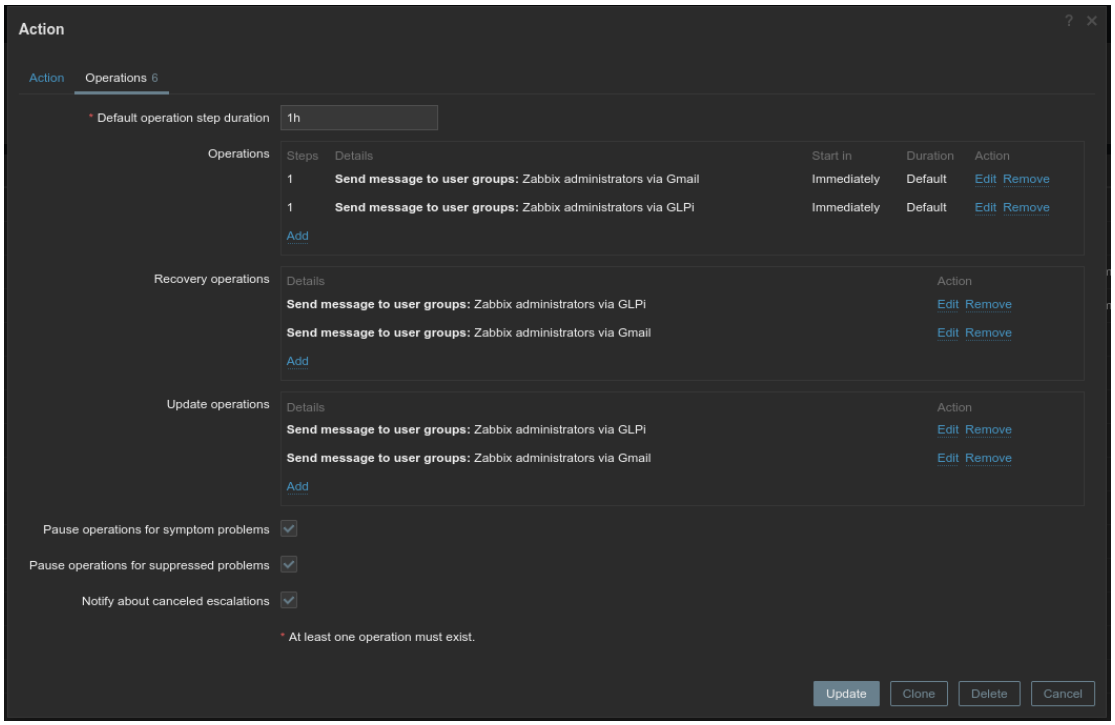


Figura 89 - Operações associadas à ação criada

16.3. Demonstração da integração

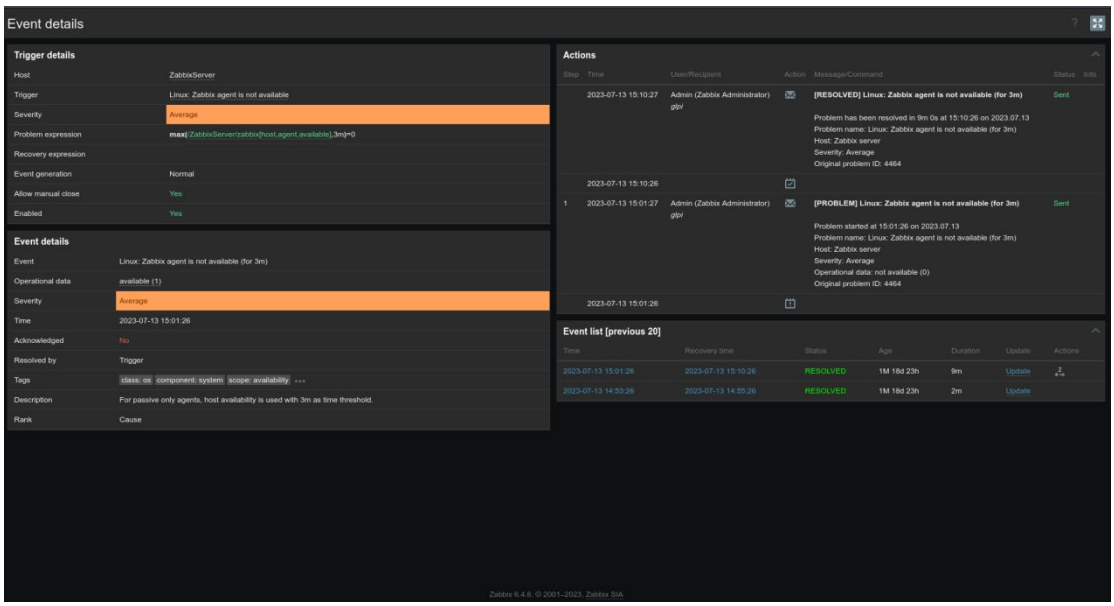


Figura 90 - Evento no Zabbix

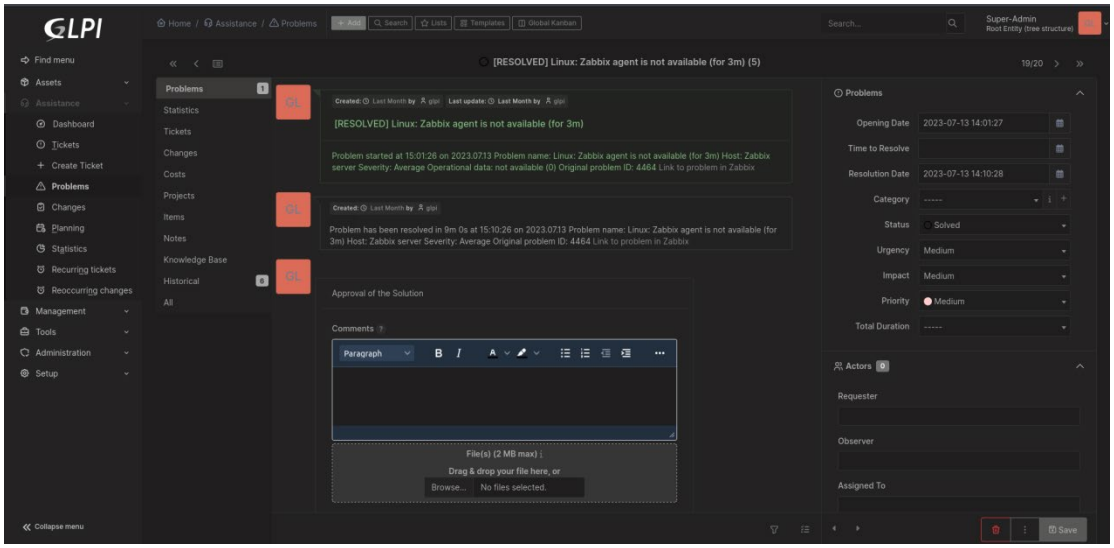


Figura 91 - Evento do Zabbix a aparecer como problema no GLPi de forma automatizada

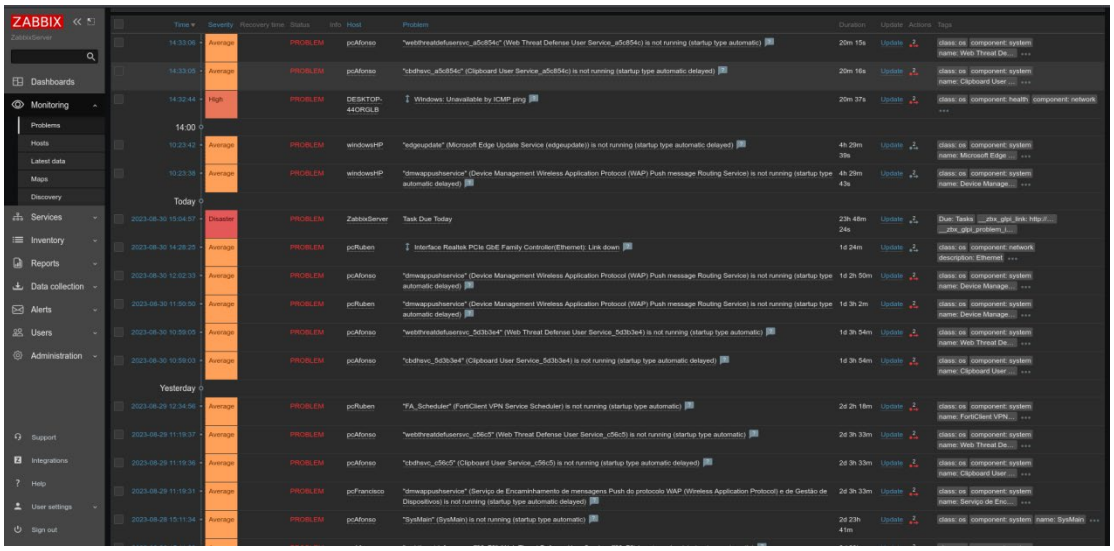


Figura 92 - Eventos do Zabbix

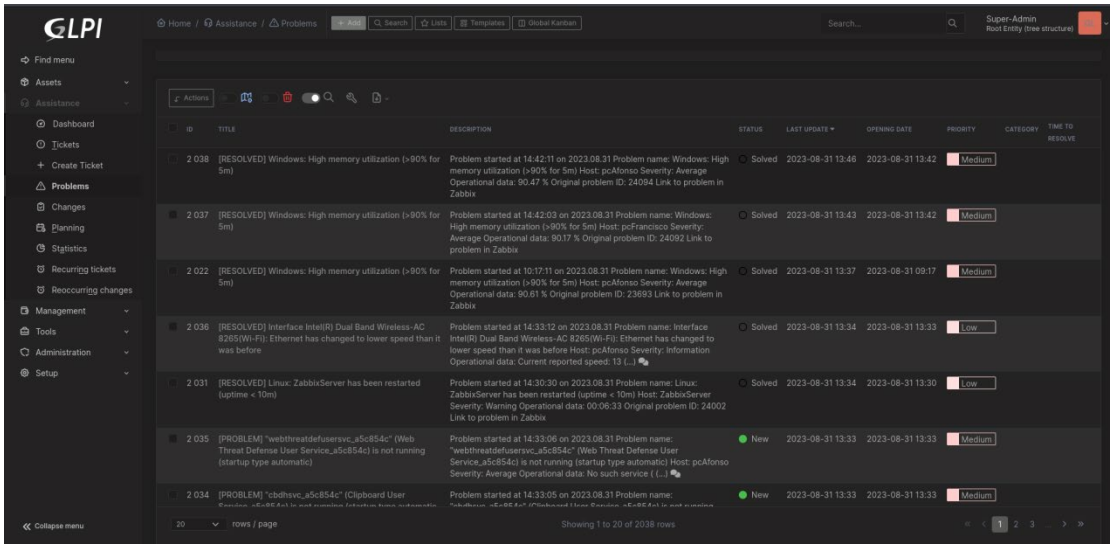


Figura 93 - Problemas no GLPi relativos aos eventos do Zabbix

Anexo 17 - Integração com Grafana

17.1. Instalação do Grafana

```
sudo apt-get install -y apt-transport-https

sudo apt-get install -y software-properties-common wget

sudo wget -q -O /usr/share/keyrings/grafana.key
https://apt.grafana.com/gpg.key

echo "deb [signed-by=/usr/share/keyrings/grafana.key]
https://apt.grafana.com stable main" | sudo tee -a
/etc/apt/sources.list.d/grafana.list

sudo apt-get update

sudo apt-get install grafana
```

Após este processo basta aceder ao URL: http://localhost:3000 e tem-se logo acesso ao software sem a necessidade de configuração do mesmo.

17.2. Implementação no Grafana

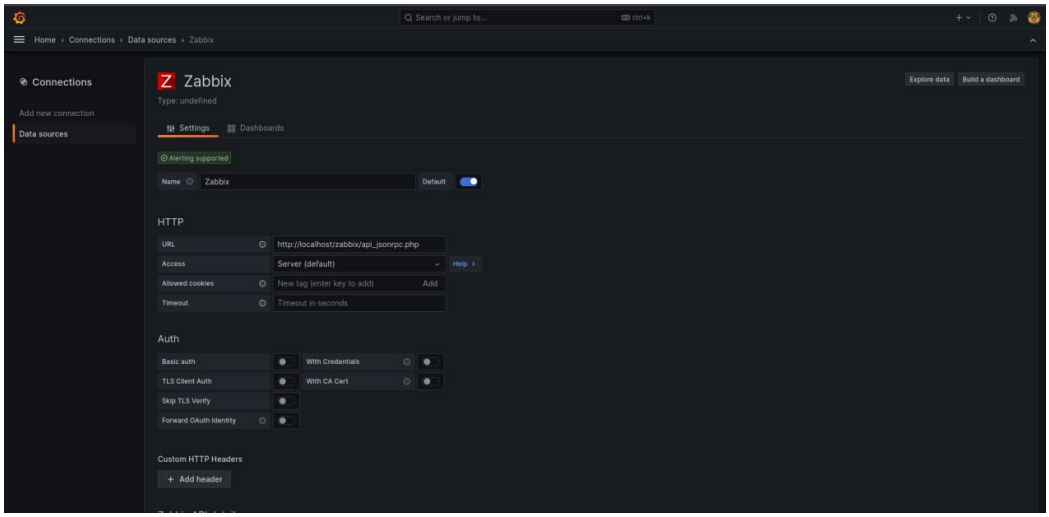


Figura 94 - Configuração do Grafana de maneira a ser integrado com Zabbix

17.3. Demonstração do Grafana

17.3.1. Dashboard criado

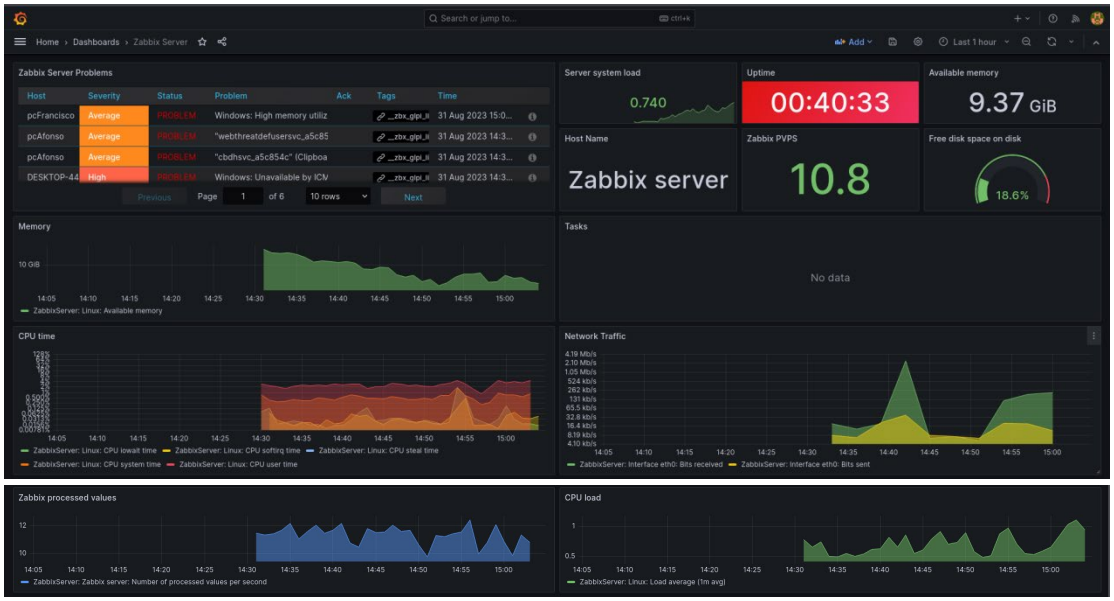


Figura 95 - Dashboard criado de raiz no Grafana

17.3.2. Dashboard importado

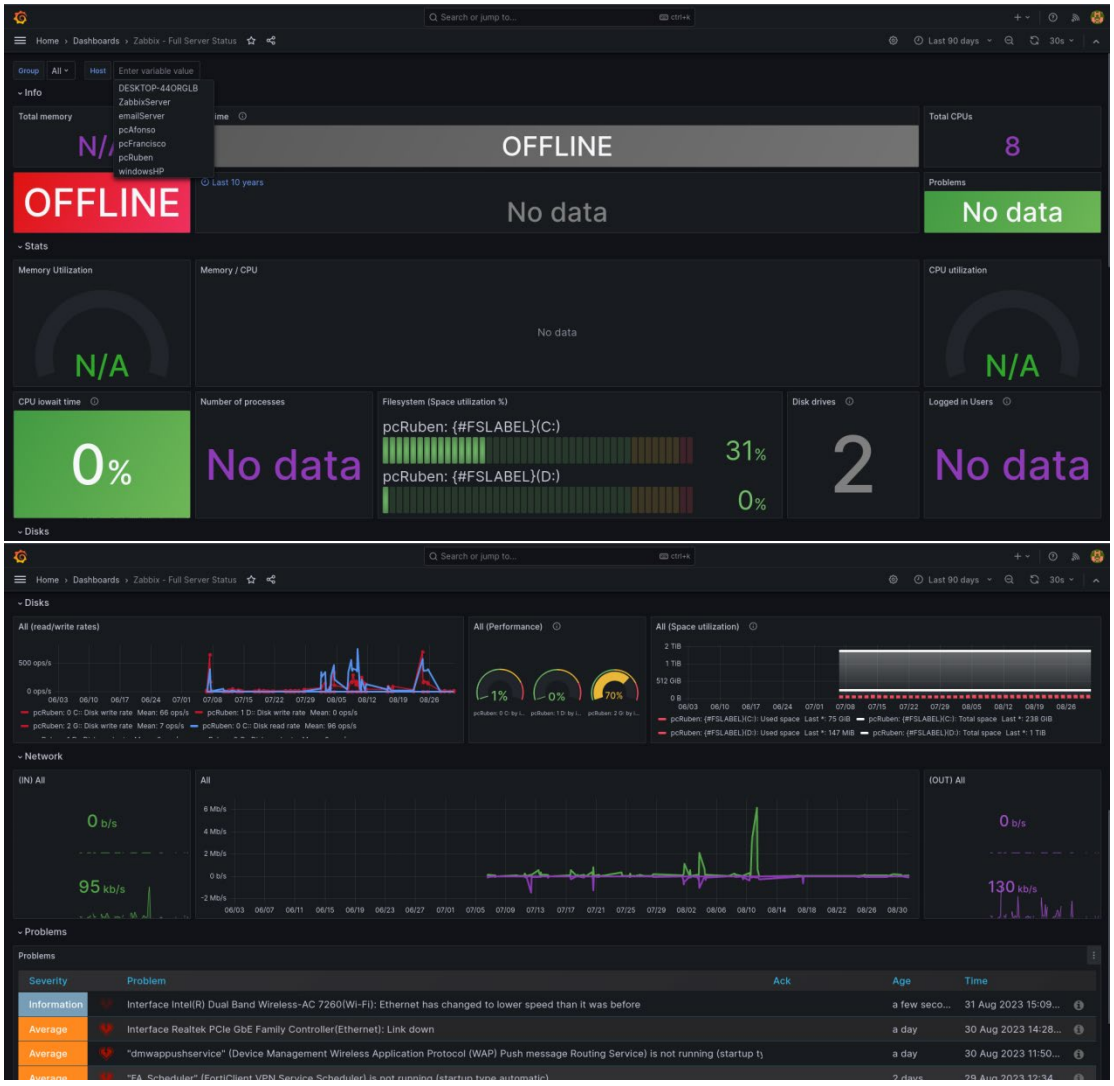


Figura 96 - Dashboard importado Grafana

Anexo 18 - Aplicação React Native

18.1. Código-fonte

```
import React from 'react';
import { StyleSheet, View, Text, Image, Linking } from 'react-native';
import { Button } from 'react-native-elements';
import logoGLPI from './imgs/logo-glpi.png';
import logoZabbix from './imgs/zabbix_logo.png';
import logoGrafana from './imgs/grafana-logo.png';

const App = () => {
  return (
    <View style={styles.container}>
      <Text style={styles.title}>Securnet-IPG</Text>
      <Button
        containerStyle={styles.buttonContainer}
        buttonStyle={styles.button}
        icon={<Image source={logoGLPI} style={styles.image}
resizeMode="contain" />}
        onPress={() => Linking.openURL('http://192.168.150.109')}
      />
      <Button
        containerStyle={styles.buttonContainer}
        buttonStyle={styles.button}
        icon={<Image source={logoZabbix} style={styles.image}
resizeMode="contain" />}
        onPress={() => Linking.openURL('http://192.168.150.109/zabbix')}
      />
      <Button
        containerStyle={styles.buttonContainer}
        buttonStyle={styles.button}
        icon={<Image source={logoGrafana} style={styles.image}
resizeMode="contain" />}
        onPress={() => Linking.openURL('http://192.168.150.109:3000')}
      />
    </View>
  );
};
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    backgroundColor: '#f5f5f5',
    padding: 20,
  },
  title: {
    fontSize: 24,
    fontWeight: 'bold',
    textAlign: 'center',
    marginBottom: 40,
  },
```

```
  buttonContainer: {
    marginVertical: 10,
  },
  button: {
    padding: 15,
    backgroundColor: '#000000',
  },
  image: {
    width: 50,
    height: 50,
  },
});
export default App;
```

18.2. Documentação de icons e splash

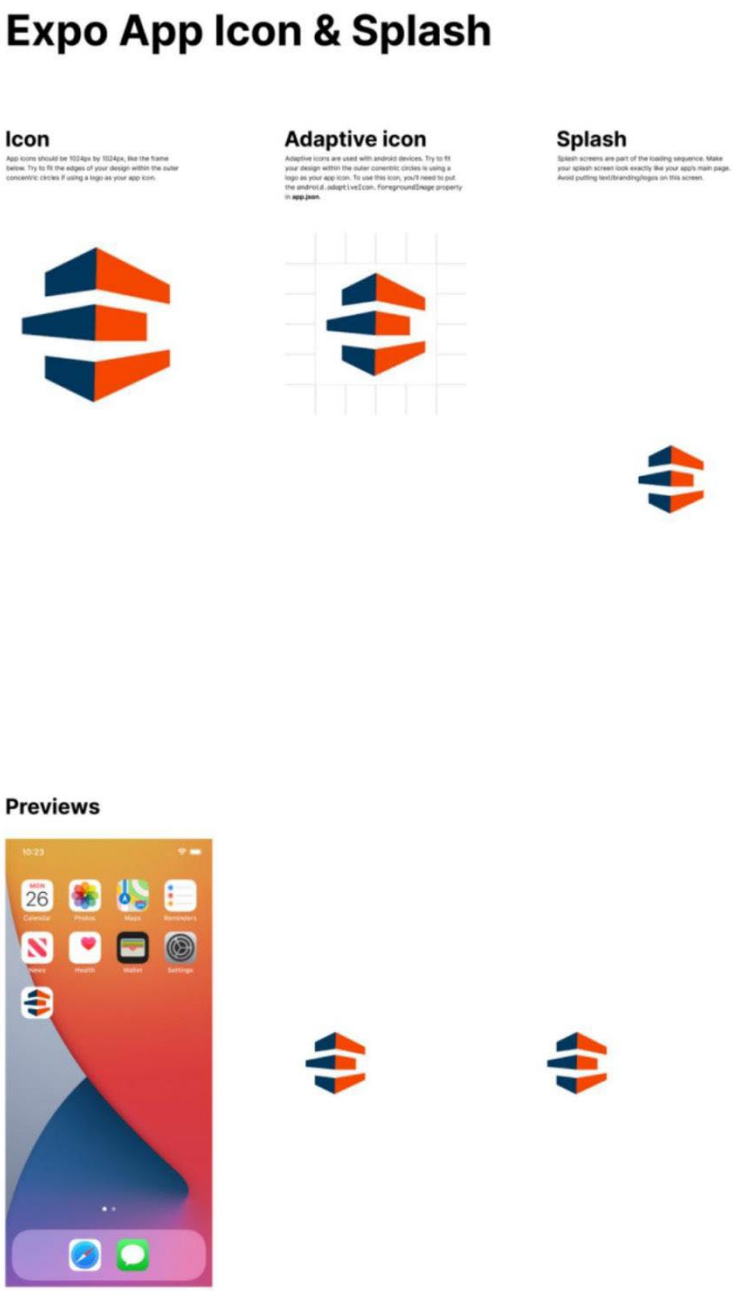


Figura 97 - Template expo splash e icon

Anexo 19 - Machine Learning

19.1. Extração de dados

```
import mariadb
import pandas as pd
try:
    conn = mariadb.connect(
        user="zabbix",
        password="password",
        host="localhost",
        port=3306,
        database="zabbix"
    )
    cur = conn.cursor()
    cur.execute("""SELECT
    sub_h.host,
    sub_t.description,
    sub_t.priority,
    COUNT(*) as event_count,
    AVG(sub_e.duration) as avg_duration
FROM
    (SELECT
         e1.objectid AS triggerid,
         e1.object,
         e1.value,
         COALESCE(e2.clock - e1.clock, 0) AS duration
     FROM
         events e1
     LEFT JOIN
         events e2 ON e1.objectid = e2.objectid AND e2.eventid = (SELECT
MIN(eventid) FROM events WHERE objectid = e1.objectid AND eventid >
e1.eventid)
    ) AS sub_e
JOIN
    (SELECT
        f.triggerid,
        f.itemid
    FROM
        functions f) AS sub_f ON sub_e.triggerid = sub_f.triggerid
JOIN
    (SELECT
        i.itemid,
        i.hostid
    FROM
        items i) AS sub_i ON sub_f.itemid = sub_i.itemid
JOIN
    (SELECT
        t.triggerid,
```

```
        t.description,
        (CASE
            WHEN t.priority = 0 THEN 'Not classified'
            WHEN t.priority = 1 THEN 'Information'
            WHEN t.priority = 2 THEN 'Warning'
            WHEN t.priority = 3 THEN 'Average'
            WHEN t.priority = 4 THEN 'High'
            ELSE 'Disaster'
        END) AS priority
    FROM
        triggers t) AS sub_t ON sub_f.triggerid = sub_t.triggerid
JOIN
    (SELECT
        h.hostid,
        h.host
    FROM
        hosts h) AS sub_h ON sub_i.hostid = sub_h.hostid
WHERE
    sub_e.object = 0 AND sub_e.value = 1
GROUP BY
    sub_h.host, sub_t.description, sub_t.priority
""")
    rows = cur.fetchall()
    conn.close()
except mariadb.Error as e:
    print(f"Error connecting to MariaDB Platform: {e}")
    sys.exit(1)
df = pd.DataFrame(rows, columns=["host", "description", "priority",
"event_count", "avg_duration"])
df['avg_duration'] = df['avg_duration'].astype(float)
df['avg_duration'] = (df['avg_duration'] / 60).round(2)
df = df.sort_values(by='event_count', ascending=False)
print(df)
df.to_csv('zabbix_events2.csv', index=False)
```

19.2. Elbow method

```
import pandas as pd

import numpy as np

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt


df = pd.read_csv('zabbix_events2.csv')


scaler = StandardScaler()

scaled_data = scaler.fit_transform(df[['event_count', 'avg_duration']])
```

```
cluster_range = range(1, 10)

inertias = []

for k in cluster_range:

kmeans = KMeans(n_clusters=k, n_init=10, random_state=0).fit(scaled_data)

inertias.append(kmeans.inertia_)


plt.figure(figsize=(10,6))

plt.plot(cluster_range, inertias, marker='o', linestyle='--')

plt.xlabel('Number of Clusters')

plt.ylabel('Inertia')

plt.title('Elbow Method For Optimal Number of Clusters')

plt.grid(True)

plt.show()
```

19.3. Processamento e modelagem

```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score

df = pd.read_csv('zabbix_events2.csv')

scaler = StandardScaler()
scaled_data = scaler.fit_transform(df[['event_count', 'avg_duration']])

kmeans = KMeans(n_clusters=3, random_state=0).fit(scaled_data)

df['cluster'] = kmeans.labels_

def label_events(row):
    if row['event_count'] > df['event_count'].quantile(0.8) and
row['avg_duration'] < df['avg_duration'].quantile(0.70):
        return 'Most irrelevant ( HIGH COUNT | LOW AVG DURATION | Most
likely needs a threshold)'
    elif row['event_count'] > df['event_count'].quantile(0.8):
        return 'Irrelevant (HIGH COUNT | MEDIUM TO HIGH AVG DURATION | Needs
analyse)'
    else:
        return 'Relevant (DONT APPEAR OFTEN | LOW COUNT)'

df['relevance'] = df.apply(label_events, axis=1)

relevance_order = [
    'Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a
threshold)',
```

```
    'Irrelevant (HIGH COUNT | MEDIUM TO HIGH AVG DURATION | Needs analyse)',
    'Relevant (DONT APPEAR OFTEN | LOW COUNT)'
]
df['relevance_rank'] =
df['relevance'].astype('category').cat.set_categories(relevance_order,
ordered=True)
df.sort_values(by='relevance_rank', inplace=True)
df.drop(columns=['relevance_rank'], inplace=True)

df.to_csv('zabbix_events_3clusters_classified.csv', index=False)

print(df[['host', 'description', 'event_count', 'avg_duration',
'relevance']])
score = silhouette_score(scaled_data, kmeans.labels_)
print('Silhouette Score:', score)
```

19.4. Análise de dados

```
fig, ax1 = plt.subplots(1, 1)
fig.set_size_inches(8, 6)

ax1.set_xlim([-0.1, 1])
ax1.set_ylim([0, len(scaled_data) + (kmeans.n_clusters + 1) * 10])

silhouette_avg = silhouette_score(scaled_data, kmeans.labels_)
print("The average silhouette_score is :", silhouette_avg)

sample_silhouette_values = silhouette_samples(scaled_data, kmeans.labels_)

y_lower = 10
for i in range(kmeans.n_clusters):
    ith_cluster_silhouette_values = \
        sample_silhouette_values[kmeans.labels_ == i]

    ith_cluster_silhouette_values.sort()

    size_cluster_i = ith_cluster_silhouette_values.shape[0]
    y_upper = y_lower + size_cluster_i

    color = plt.cm.nipy_spectral(float(i) / kmeans.n_clusters)
    ax1.fill_betweenx(np.arange(y_lower, y_upper),
                      0, ith_cluster_silhouette_values,
                      facecolor=color, edgecolor=color, alpha=0.7)

    ax1.text(-0.05, y_lower + 0.5 * size_cluster_i, str(i))

    y_lower = y_upper + 10

ax1.set_title("The silhouette plot for the various clusters.")
ax1.set_xlabel("The silhouette coefficient values")
ax1.set_ylabel("Cluster label")

ax1.axvline(x=silhouette_avg, color="red", linestyle="--")
```

```
ax1.set_yticks([])  # Clear the y-axis labels / ticks
ax1.set_xticks([-0.1, 0, 0.2, 0.4, 0.6, 0.8, 1])

fig, ax2 = plt.subplots(1, 1)
fig.set_size_inches(8, 6)

scatter = ax2.scatter(df['event_count'], df['avg_duration'], c=df['clus-
ter'], cmap='viridis')
ax2.set_xlabel('Event Count')
ax2.set_ylabel('Average Duration')
ax2.set_title('Cluster Visualization (Event Count vs. Average Duration)')
colorbar = plt.colorbar(scatter, ax=ax2, label='Cluster')

centroids = kmeans.cluster_centers_
ax2.scatter(centroids[:, 0], centroids[:, 1], c='red', marker='X', s=100,
label='Centroids')
ax2.legend()

plt.show()
```

Anexo 20 – Dataframe

	A	B	C	D	E
1	host	description	priority	event_count	avg_duration
2	pcRuben	Interface Intel(R) Dual Band Wireless-AC 7260(Wi-Fi): Ethernet has char	Information	2379	15.27
3	pcAfonso	Interface Intel(R) Dual Band Wireless-AC 8265(Wi-Fi): Ethernet has char	Information	231	1.01
4	pcFrancisco	Windows: The Memory Pages/sec is too high	Warning	111	145.39
5	pcFrancisco	Windows: CPU privileged time is too high	Warning	102	214.08
6	pcRuben	Windows: The Memory Pages/sec is too high	Warning	91	53.75
7	pcFrancisco	Interface Intel(R) Dual Band Wireless-AC 8260(Wi-Fi): Ethernet has char	Information	81	3.56
8	pcAfonso	Windows: The Memory Pages/sec is too high	Warning	74	234.11
9	ZabbixServer	Interface eth0: Link down	Average	69	426.17
10	ZabbixServer	Interface wlan0: Link down	Average	60	200.72
11	Firewall	Fortinet {HOST.NAME} Rebooted	Average	56	4.49
12	windowsHP	"dmwappushservice" (Device Management Wireless Application Protoc	Average	55	1060.49
13	FirewallLAN	Fortinet {HOST.NAME} Rebooted	Average	54	9.97
14	pcFrancisco	Windows: High memory utilization	Average	45	111.78
15	ZabbixServer	Linux: {HOST.NAME} has been restarted	Warning	41	31.39
16	pcFrancisco	Windows: CPU queue length is too high	Warning	30	268.26
17	DESKTOP-44ORGLB	Windows: Unavailable by ICMP ping	High	29	824.62
18	SFTPServer	Linux: Host has been restarted	Warning	28	8.41
19	emailServer	Linux: Zabbix agent is not available	Average	28	943.57
20	pcFrancisco	Windows: Zabbix agent is not available	Average	28	804.83
21	windowsHP	Windows: Zabbix agent is not available	Average	23	580.68
22	SFTPServer	Linux: Unavailable by ICMP ping	High	23	1081.83
23	ZabbixServer	Zabbix server: Utilization of discoverer processes is high	Average	20	12.35
24	pcRuben	Windows: Zabbix agent is not available	Average	19	1424.34
25	pcAfonso	Windows: Zabbix agent is not available	Average	17	1626.78
26	pcFrancisco	Windows: High CPU utilization	Warning	14	4.64
27	emailServer	Linux: {HOST.NAME} has been restarted	Warning	14	9.24
28	emailServer	Linux: System time is out of sync	Warning	11	12.84
29	ZabbixServer	Linux: High swap space usage	Warning	10	651.97
30	windowsHP	Windows: CPU queue length is too high	Warning	10	1.44
31	pcFrancisco	Windows: Host has been restarted	Warning	9	147.73
32	ZabbixServer	Linux: Number of installed packages has been changed	Warning	8	2557.5
33	pcFrancisco	"SplunkForwarder" (SplunkForwarder) is not running	Average	7	8.86
34	ZabbixServer	Linux: Zabbix agent is not available	Average	7	8.71
35	ZabbixServer	Linux: /etc/passwd has been changed	Information	6	95

36	DESKTOP-44ORGLB	Windows: No SNMP data collection	Warning	6	305.33
37	pcFrancisco	Interface Intel(R) Ethernet Connection I219-V(Ethernet): Link down	Average	6	660.5
38	windowsHP	Windows: The Memory Pages/sec is too high	Warning	6	1.99
39	pcRuben	"FA_Scheduler" (FortiClient VPN Service Scheduler) is not running	Average	5	5046
40	ZabbixServer	Linux: High CPU utilization	Warning	5	1.36
41	ZabbixServer	Task Due Today	Disaster	5	912.26
42	SFTPServer	Linux: No SNMP data collection	Warning	5	4.87
43	windowsHP	Windows: CPU privileged time is too high	Warning	4	1.99
44	pcFrancisco	Windows: Operating system description has changed	Information	4	780
45	pcRuben	Windows: Operating system description has changed	Information	4	274.98
46	windowsHP	Windows: Operating system description has changed	Information	4	2880
47	pcFrancisco	"AnyDesk" (AnyDesk Service) is not running	Average	4	2307.25
48	pcAfonso	Windows: Operating system description has changed	Information	4	269.14
49	windowsHP	Windows: Host has been restarted	Warning	3	343.63
50	DESKTOP-44ORGLB	Interface wireless_8(Local Area Connection* 9-VirtualBox NDIS Light-W	Average	3	0
51	DESKTOP-44ORGLB	Interface wireless_10(Local Area Connection* 9-WFP 802.3 MAC Layer Li	Average	3	0
52	DESKTOP-44ORGLB	Interface wireless_9(Local Area Connection* 9-QoS Packet Scheduler-0(	Average	3	0
53	ZabbixServer	Zabbix server: Version has changed	Information	3	483
54	ZabbixServer	Zabbix server: Utilization of unreachable poller processes is high	Average	2	268
55	pcFrancisco	"stisvc" (AquisiÃ§Ã£o de Imagem do Windows (WIA)) is not running	Average	2	47.5
56	pcAfonso	Windows: CPU privileged time is too high	Warning	2	1.11
57	pcRuben	"SgrmBroker" (System Guard Runtime Monitor Broker) is not running	Average	2	1.25
58	pcRuben	Windows: Host has been restarted	Warning	2	980.75
59	emailServer	Linux: Operating system description has changed	Information	2	1380
60	emailServer	sda: Disk read/write request responses are too high	Warning	2	1446
61	ZabbixServer	Zabbix server: More than 100 items having missing data for more than 1	Warning	2	233.34
62	pcFrancisco	"edgeupdate" (Microsoft Edge Update Service (edgeupdate)) is not run	Average	2	4197.51
63	windowsHP	"edgeupdate" (Microsoft Edge Update Service (edgeupdate)) is not run	Average	2	5735.5
64	ZabbixServer	Linux: Operating system description has changed	Information	2	60
65	SFTPServer	Linux: High ICMP ping loss	Warning	2	492
66	windowsHP	Windows: High CPU utilization	Warning	2	1.5
67	DESKTOP-44ORGLB	Windows: High ICMP ping loss	Warning	2	1
68	pcAfonso	Windows: Host has been restarted	Warning	2	463.75
69	pcRuben	"edgeupdate" (Microsoft Edge Update Service (edgeupdate)) is not run	Average	1	0
70	pcAfonso	"UsoSvc" (Update Orchestrator Service) is not running	Average	1	924
71	pcAfonso	"cbdhsvc_1215bcf5" (Clipboard User Service_1215bcf5) is not running	Average	1	0
72	pcRuben	"SplunkForwarder" (SplunkForwarder) is not running	Average	1	1
73	pcRuben	"UsoSvc" (Update Orchestrator Service) is not running	Average	1	963.05
74	pcRuben	"WSearch" (Windows Search) is not running	Average	1	1
75	pcAfonso	"cbdhsvc_f8cf1b8" (Clipboard User Service_f8cf1b8) is not running	Average	1	0
76	pcRuben	"uhssvc" (Microsoft Update Health Service) is not running	Average	1	1.47
77	pcRuben	"wscsvc" (Security Center) is not running	Average	1	0.75
78	pcRuben	CPU load too high	Not classified	1	0.13
79	pcAfonso	"cbdhsvc_1d87ade0" (Clipboard User Service_1d87ade0) is not running	Average	1	0
80	pcRuben	Windows: CPU privileged time is too high	Warning	1	1
81	pcRuben	Windows: High CPU utilization	Warning	1	0.13
82	ZabbixServer	Zabbix server: Utilization of preprocessing manager processes is high	Average	1	1
83	ZabbixServer	Zabbix server: Utilization of poller processes is high	Average	1	0.92
84	webServer	Apache: Host has been restarted	Information	1	9.72
85	webServer	Service is down	High	1	978.27
86	ZabbixServer	TimeLeftTest	High	1	1
87	FirewallLAN	Fortinet {HOSTNAME} - Usage of CPU over 95%	Average	1	1
88	pcAfonso	"cbdhsvc_18257ce6" (Clipboard User Service_18257ce6) is not running	Average	1	0
89	pcAfonso	"cbdhsvc_25c2837e" (Clipboard User Service_25c2837e) is not running	Average	1	0
90	pcAfonso	"cbdhsvc_200fb99f" (Clipboard User Service_200fb99f) is not running	Average	1	0
91	pcAfonso	"webthreatdefusersvc_b8343" (Web Threat Defense User Service_b834	Average	1	0
92	pcAfonso	"jhi_service" (Intel(R) Dynamic Application Loader Host Interface Servic	Average	1	1.15
93	pcAfonso	"webthreatdefusersvc_1215bcf5" (Web Threat Defense User Service_12	Average	1	0
94	pcAfonso	"webthreatdefusersvc_18257ce6" (Web Threat Defense User Service_1	Average	1	0
95	pcAfonso	"webthreatdefusersvc_1d87ade0" (Web Threat Defense User Service_1	Average	1	0
96	pcAfonso	"webthreatdefusersvc_200fb99f" (Web Threat Defense User Service_20	Average	1	0
97	pcAfonso	"webthreatdefusersvc_21fb6817" (Web Threat Defense User Service_21	Average	1	0
98	pcAfonso	"webthreatdefusersvc_25c2837e" (Web Threat Defense User Service_2	Average	1	0
99	pcAfonso	"webthreatdefusersvc_25cdd49b" (Web Threat Defense User Service_2	Average	1	0
100	pcAfonso	"cbdhsvc_b8343" (Clipboard User Service_b8343) is not running	Average	1	0
101	pcAfonso	"webthreatdefusersvc_31e60805" (Web Threat Defense User Service_3	Average	1	0
102	pcAfonso	"webthreatdefusersvc_7080176" (Web Threat Defense User Service_70	Average	1	0
103	pcAfonso	"webthreatdefusersvc_aa6915e" (Web Threat Defense User Service_aa	Average	1	0
104	pcAfonso	"webthreatdefusersvc_aa746" (Web Threat Defense User Service_aa74	Average	1	0
105	pcAfonso	"webthreatdefusersvc_ad5c5" (Web Threat Defense User Service_ad5c	Average	1	0

106	pcAfonso	"webthreatdefusersvc_f8cf1b8" (Web Threat Defense User Service_f8cf	Average	1	0
107	pcAfonso	"cbdhsvc_21fb6817" (Clipboard User Service_21fb6817) is not running	Average	1	0
108	pcAfonso	"wscsvc" (Security Center) is not running	Average	1	2
109	pcAfonso	0 C:: Disk write request responses are too high	Warning	1	0.97
110	pcAfonso	"cbdhsvc_ad5c5" (Clipboard User Service_ad5c5) is not running	Average	1	0
111	pcAfonso	"cbdhsvc_aa746" (Clipboard User Service_aa746) is not running	Average	1	0
112	pcAfonso	Windows: System time is out of sync	Warning	1	0
113	pcAfonso	"cbdhsvc_aa6915e" (Clipboard User Service_aa6915e) is not running	Average	1	0
114	pcAfonso	"cbdhsvc_7080176" (Clipboard User Service_7080176) is not running	Average	1	0
115	pcAfonso	"cbdhsvc_31e60805" (Clipboard User Service_31e60805) is not running	Average	1	0
116	pcFrancisco	"MicrosoftSearchInBing" (Microsoft Search in Bing) is not running	Average	1	0
117	pcAfonso	"cbdhsvc_2d529fc1" (Clipboard User Service_2d529fc1) is not running	Average	1	0
118	pcFrancisco	"SysMain" (SysMain) is not running	Average	1	1492
119	pcFrancisco	"UsoSvc" (Atualizar serviÃ§o Orchestrator) is not running	Average	1	6
120	pcAfonso	"cbdhsvc_25cdd49b" (Clipboard User Service_25cdd49b) is not running	Average	1	0
121	pcAfonso	"edgeupdate" (Microsoft Edge Update Service (edgeupdate)) is not run	Average	1	0
122	pcAfonso	"webthreatdefusersvc_2d529fc1" (Web Threat Defense User Service_2c	Average	1	0

Anexo 21 – Dataframe clusterizado

	host	description	priority	event_co	avg_duration	cluster	relevance
2	pcRuben	Interface Intel(R) Dual Band Wirele	Informatic	2379	15.27	2	Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a threshold)
3	ZabbixServer	Linux: {HOST.NAME} has been resta	Warning	41	31.39	0	Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a threshold)
4	pcFrancisco	Windows: High memory utilization	Average	45	111.78	0	Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a threshold)
5	FirewallLAN	Fortinet {HOST.NAME} Rebooted	Average	54	9.97	0	Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a threshold)
6	Firewall	Fortinet {HOST.NAME} Rebooted	Average	56	4.49	0	Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a threshold)
7	ZabbixServer	Zabbix server: Utilization of discove	Average	20	12.35	0	Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a threshold)
8	ZabbixServer	Interface wlan0: Link down	Average	60	200.72	0	Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a threshold)
9	pcFrancisco	Interface Intel(R) Dual Band Wirele	Informatic	81	3.56	0	Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a threshold)
10	pcRuben	Windows: The Memory Pages/sec i	Warning	91	53.75	0	Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a threshold)
11	pcFrancisco	Windows: CPU privileged time is to	Warning	102	214.08	0	Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a threshold)
12	pcFrancisco	Windows: The Memory Pages/sec i	Warning	111	145.39	0	Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a threshold)
13	pcAfonso	Interface Intel(R) Dual Band Wirele	Informatic	231	1.01	0	Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a threshold)
14	SFTPServer	Linux: Host has been restarted	Warning	28	8.41	0	Most irrelevant (HIGH COUNT | LOW AVG DURATION | Most likely needs a threshold)
15	pcFrancisco	Windows: Zabbix agent is not availa	Average	28	804.83	0	Irrelevant (HIGH COUNT | MEDIUM TO HIGH AVG DURATION | Needs analyse)
16	SFTPServer	Linux: Unavailable by ICMP ping	High	23	1081.83	0	Irrelevant (HIGH COUNT | MEDIUM TO HIGH AVG DURATION | Needs analyse)
17	windowsHP	Windows: Zabbix agent is not availa	Average	23	580.68	0	Irrelevant (HIGH COUNT | MEDIUM TO HIGH AVG DURATION | Needs analyse)
18	pcAfonso	Windows: Zabbix agent is not availa	Average	17	1626.78	0	Irrelevant (HIGH COUNT | MEDIUM TO HIGH AVG DURATION | Needs analyse)
19	pcRuben	Windows: Zabbix agent is not availa	Average	19	1424.34	0	Irrelevant (HIGH COUNT | MEDIUM TO HIGH AVG DURATION | Needs analyse)
20	DESKTOP-44O	Windows: Unavailable by ICMP pin	High	29	824.62	0	Irrelevant (HIGH COUNT | MEDIUM TO HIGH AVG DURATION | Needs analyse)
21	pcFrancisco	Windows: CPU queue length is too	Warning	30	268.26	0	Irrelevant (HIGH COUNT | MEDIUM TO HIGH AVG DURATION | Needs analyse)
22	windowsHP	"dmwappushservice" (Device Mana	Average	55	1060.49	0	Irrelevant (HIGH COUNT | MEDIUM TO HIGH AVG DURATION | Needs analyse)
23	ZabbixServer	Interface eth0: Link down	Average	69	426.17	0	Irrelevant (HIGH COUNT | MEDIUM TO HIGH AVG DURATION | Needs analyse)
24	pcAfonso	Windows: The Memory Pages/sec i	Warning	74	234.11	0	Irrelevant (HIGH COUNT | MEDIUM TO HIGH AVG DURATION | Needs analyse)
25	emailServer	Linux: Zabbix agent is not available	Average	28	943.57	0	Irrelevant (HIGH COUNT | MEDIUM TO HIGH AVG DURATION | Needs analyse)
26	pcAfonso	"webthreatdefusersvc_1d87ade0" (	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
27	pcAfonso	"webthreatdefusersvc_18257ce6" (	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
28	pcAfonso	"webthreatdefusersvc_1215bcf5" (\	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
29	pcAfonso	"jhi_service" (Intel(R) Dynamic App	Average	1	1.15	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
30	pcAfonso	"webthreatdefusersvc_b8343" (We	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
31	pcAfonso	"cbdhsvc_200fb99f" (Clipboard Use	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
32	pcAfonso	"cbdhsvc_25c2837e" (Clipboard Use	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
33	pcAfonso	"cbdhsvc_18257ce6" (Clipboard Use	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
34	FirewallLAN	Fortinet {HOSTNAME} - Usage of CP	Average	1	1	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
35	ZabbixServer	TimeLeftTest	High	1	1	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)

36	ZabbixServer	Zabbix server: Utilization of poller	Average	1	0.92	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
37	webServer	Apache: Host has been restarted	Informatic	1	9.72	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
38	ZabbixServer	Zabbix server: Utilization of prepro	Average	1	1	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
39	pcRuben	Windows: High CPU utilization	Warning	1	0.13	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
40	pcRuben	Windows: CPU privileged time is to	Warning	1	1	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
41	pcAfonso	"cbdhsvc_1d87ade0" (Clipboard Use	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
42	pcRuben	CPU load too high	Not classi	1	0.13	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
43	pcRuben	"wscsvc" (Security Center) is not ru	Average	1	0.75	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
44	pcRuben	"uhssvc" (Microsoft Update Health	Average	1	1.47	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
45	pcAfonso	"webthreatdefusersvc_200fb99f" (\	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
46	webServer	Service is down	High	1	978.27	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
47	pcAfonso	"webthreatdefusersvc_21fb6817" ('	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
48	pcAfonso	0 C:: Disk write request responses a	Warning	1	0.97	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
49	pcAfonso	"webthreatdefusersvc_25cdd49b" (	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
50	pcAfonso	"cbdhsvc_25cdd49b" (Clipboard Use	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
51	pcFrancisco	"UsoSvc" (Atualizar serviÃ§o Orche	Average	1	6	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
52	pcFrancisco	"SysMain" (SysMain) is not running	Average	1	1492	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
53	pcAfonso	"cbdhsvc_2d529fc1" (Clipboard Use	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
54	pcFrancisco	"MicrosoftSearchInBing" (Microsoft	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
55	pcAfonso	"cbdhsvc_31e60805" (Clipboard Use	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
56	pcAfonso	"cbdhsvc_7080176" (Clipboard User	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
57	pcAfonso	"cbdhsvc_aa6915e" (Clipboard User	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
58	pcAfonso	Windows: System time is out of syr	Warning	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
59	pcAfonso	"cbdhsvc_aa746" (Clipboard User Se	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
60	pcAfonso	"cbdhsvc_ad5c5" (Clipboard User Se	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
61	pcAfonso	"cbdhsvc_f8cf1b8" (Clipboard User	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
62	pcAfonso	"wscsvc" (Security Center) is not ru	Average	1	2	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
63	pcAfonso	"cbdhsvc_21fb6817" (Clipboard Use	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
64	pcAfonso	"webthreatdefusersvc_f8cf1b8" (W	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
65	pcAfonso	"webthreatdefusersvc_ad5c5" (We	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
66	pcAfonso	"webthreatdefusersvc_aa746" (We	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
67	pcAfonso	"webthreatdefusersvc_aa6915e" (V	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
68	pcAfonso	"webthreatdefusersvc_7080176" (V	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
69	pcAfonso	"webthreatdefusersvc_31e60805" (	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
70	pcAfonso	"cbdhsvc_b8343" (Clipboard User Se	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
71	pcAfonso	"webthreatdefusersvc_25c2837e" (	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
72	pcRuben	"WSearch" (Windows Search) is not	Average	1	1	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
73	pcFrancisco	"edgeupdate" (Microsoft Edge Upd	Average	2	4197.51	1	Relevant (DONT APPEAR OFTEN | LOW COUNT)
74	pcRuben	"SplunkForwarder" (SplunkForwarc	Average	1	1	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
75	windowsHP	Windows: Operating system descri	Informatic	4	2880	1	Relevant (DONT APPEAR OFTEN | LOW COUNT)
76	pcRuben	Windows: Operating system descri	Informatic	4	274.98	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
77	pcFrancisco	Windows: Operating system descri	Informatic	4	780	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
78	windowsHP	Windows: CPU privileged time is to	Warning	4	1.99	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
79	SFTPServer	Linux: No SNMP data collection	Warning	5	4.87	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
80	ZabbixServer	Task Due Today	Disaster	5	912.26	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
81	ZabbixServer	Linux: High CPU utilization	Warning	5	1.36	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
82	pcRuben	"FA_Scheduler" (FortiClient VPN Se	Average	5	5046	1	Relevant (DONT APPEAR OFTEN | LOW COUNT)
83	windowsHP	Windows: The Memory Pages/sec i	Warning	6	1.99	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
84	pcFrancisco	Interface Intel(R) Ethernet Connect	Average	6	660.5	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
85	DESKTOP-44O	Windows: No SNMP data collection	Warning	6	305.33	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
86	ZabbixServer	Linux: /etc/passwd has been chang	Informatic	6	95	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
87	ZabbixServer	Linux: Zabbix agent is not available	Average	7	8.71	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
88	pcFrancisco	"SplunkForwarder" (SplunkForwarc	Average	7	8.86	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
89	ZabbixServer	Linux: Number of installed package	Warning	8	2557.5	1	Relevant (DONT APPEAR OFTEN | LOW COUNT)
90	pcFrancisco	Windows: Host has been restarted	Warning	9	147.73	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
91	windowsHP	Windows: CPU queue length is too	Warning	10	1.44	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
92	ZabbixServer	Linux: High swap space usage	Warning	10	651.97	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
93	emailServer	Linux: System time is out of sync	Warning	11	12.84	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
94	emailServer	Linux: {HOST.NAME} has been resta	Warning	14	9.24	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
95	pcFrancisco	Windows: High CPU utilization	Warning	14	4.64	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
96	pcFrancisco	"AnyDesk" (AnyDesk Service) is not	Average	4	2307.25	1	Relevant (DONT APPEAR OFTEN | LOW COUNT)
97	pcRuben	"UsoSvc" (Update Orchestrator Ser	Average	1	963.05	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
98	pcAfonso	Windows: Operating system descri	Informatic	4	269.14	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
99	DESKTOP-44O	Interface wireless_8(Local Area Cor	Average	3	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
100	pcAfonso	"cbdhsvc_1215bcf5" (Clipboard Use	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
101	pcAfonso	"UsoSvc" (Update Orchestrator Ser	Average	1	924	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
102	pcRuben	"edgeupdate" (Microsoft Edge Upd	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
103	pcAfonso	Windows: Host has been restarted	Warning	2	463.75	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)

104	DESKTOP-44O	Windows: High ICMP ping loss	Warning	2	1	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
105	windowsHP	Windows: High CPU utilization	Warning	2	1.5	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
106	SFTPServer	Linux: High ICMP ping loss	Warning	2	492	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
107	ZabbixServer	Linux: Operating system descriptio	Informatic	2	60	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
108	windowsHP	"edgeupdate" (Microsoft Edge Upd	Average	2	5735.5	1	Relevant (DONT APPEAR OFTEN | LOW COUNT)
109	pcAfonso	"edgeupdate" (Microsoft Edge Upd	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
110	ZabbixServer	Zabbix server: More than 100 items	Warning	2	233.34	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
111	emailServer	sda: Disk read/write request respo	Warning	2	1446	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
112	emailServer	Linux: Operating system descriptio	Informatic	2	1380	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
113	pcRuben	Windows: Host has been restarted	Warning	2	980.75	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
114	pcRuben	"SgrmBroker" (System Guard Runti	Average	2	1.25	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
115	pcAfonso	Windows: CPU privileged time is to	Warning	2	1.11	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
116	pcFrancisco	"stisvc" (AquisiÃ§Ã£o de Imagem d	Average	2	47.5	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
117	ZabbixServer	Zabbix server: Utilization of unreac	Average	2	268	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
118	ZabbixServer	Zabbix server: Version has changed	Informatic	3	483	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
119	DESKTOP-44O	Interface wireless_9(Local Area Cor	Average	3	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
120	DESKTOP-44O	Interface wireless_10(Local Area Cc	Average	3	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
121	windowsHP	Windows: Host has been restarted	Warning	3	343.63	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)
122	pcAfonso	"webthreatdefusersvc_2d529fc1" (\	Average	1	0	0	Relevant (DONT APPEAR OFTEN | LOW COUNT)