INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA - CÂMPUS CAÇADOR
CURSO DE GRADUAÇÂO EM SISTEMAS DE INFORMAÇÃO

GUILHERME AUGUSTO LEVECKE

Monitoramento Avançado e Observabilidade de Ativos de TI Potencializados por Inteligência Artificial.

CAÇADOR, 2025.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA - CÂMPUS CAÇADOR
CURSO DE GRADUAÇÂO EM SISTEMAS DE INFORMAÇÃO

GUILHERME AUGUSTO LEVECKE

Monitoramento Avançado e Observabilidade de Ativos de TI Potencializados por Inteligência Artificial.

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Bacharel em Sistemas de Informação.

Orientador:
Prof. Ademir Goulart, Dr.

Coorientador:
Rui Batista dos Santos

CAÇADOR, 2025.

Levecke, Guilherme Augusto
L657m Monitoramento avançado e observabilidade de ativos de TI potencializados por Inteligência Artificial / Guilherme Augusto Levecke ; orientador: Ademir Goulart; coorientador: Rui Batista dos Santos. -- 2025.
68f

Trabalho de Conclusão de Curso (Graduação)-Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Caçador, 2025.
Inclui bibliografias

1. Monitoramento. 2. Observabilidade. 3. Zabbix. 4. Grafana. 5. Linux. I. Goulart, Ademir. II. Santos, Rui Batista dos. III. Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina - Graduação em Sistemas de Informação. IV. Título.

CDD 658

Ficha catalográfica elaborada pela Bibliotecária
Janice Moser Corrêa – CRB-14/1865

MONITORAMENTO AVANÇADO E OBSERVABILIDADE DE ATIVOS DE TI POTENCIALIZADOS POR INTELIGÊNCIA ARTIFICIAL.

GUILHERME AUGUSTO LEVECKE

Este Trabalho foi julgado adequado de forma parcial para obtenção do Título de Bacharel em Sistemas de Informação e aprovado na sua forma parcial pela banca examinadora do Curso de Sistemas de Informação do Instituto Federal de Educação Ciência, e Tecnologia de Santa Catarina.

CAÇADOR, 16 de agosto de 2025.

Banca Examinadora:

gov.br
Documento assinado digitalmente
ADEMIR GOULART
Data: 29/07/2025 18:15:33-0300
Verifique em https://validar.iti.gov.br

Ademir Goulart, Dr.

gov.br
Documento assinado digitalmente
RUI BATISTA DOS SANTOS
Data: 30/07/2025 19:10:55-0300
Verifique em https://validar.iti.gov.br

Rui Batista dos Santos.

gov.br
Documento assinado digitalmente
CRISTIANO MESQUITA GARCIA
Data: 30/07/2025 19:23:32-0300
Verifique em https://validar.iti.gov.br

Cristiano Mesquita Garcia, Msc.

gov.br
Documento assinado digitalmente
ANDREIA DE FATIMA MURARO
Data: 31/07/2025 07:21:35-0300
Verifique em https://validar.iti.gov.br

Andreia de Fátima Muraro.

RESUMO

O presente trabalho tem como foco principal abordar o monitoramento e a observabilidade de ativos de TI, utilizando ferramentas *open source* como *Zabbix* e *Grafana*. Em um cenário onde a alta disponibilidade e o desempenho eficiente dos sistemas são essenciais para o sucesso das organizações, torna-se fundamental implementar processos robustos e eficazes de monitoramento e observabilidade. Através dessas ferramentas, é possível obter uma visão detalhada e em tempo real da infraestrutura de TI, identificar possíveis falhas, otimizar recursos e garantir a continuidade dos serviços. Além disso, o estudo destaca a importância de uma abordagem proativa na gestão dos ativos de TI, visando minimizar tempos de inatividade e garantir a resiliência dos ambientes tecnológicos, proporcionando maior segurança, eficiência e valor agregado para as empresas. Nesse contexto, a integração com inteligência artificial (IA) surge como um diferencial estratégico, permitindo realizar a análise de falhas e a automação de resposta a incidentes. A IA potencializa a capacidade das ferramentas de monitoramento e observabilidade, tornando-as mais eficientes no controle de falhas e otimização de recursos, promovendo uma gestão ainda mais assertiva dos ativos de TI.

Palavras-chave: Monitoramento, Observabilidade, Zabbix, Grafana, Linux.

ABSTRACT

This study focuses on addressing the monitoring and observability of IT assets using open-source tools such as Zabbix and Grafana. In a scenario where high availability and efficient system performance are essential for organizational success, it becomes crucial to implement robust and effective monitoring and observability processes. Through these tools, it is possible to obtain a detailed and real-time view of the IT infrastructure, identify potential failures, optimize resources, and ensure service continuity. Furthermore, the study highlights the importance of a proactive approach to IT asset management, aiming to minimize downtime and ensure the resilience of technological environments, providing greater security, efficiency, and added value to organizations. In this context, the integration of artificial intelligence (AI) emerges as a strategic differentiator, enabling failure analysis and incident response automation. AI enhances the capabilities of monitoring and observability tools, making them more efficient in failure control and resource optimization, thus promoting an even more assertive management of IT assets.

Keywords: Monitoring, Observability, Zabbix, Grafana, Linux.

Lista de Figuras

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
CPU	*Central Processing Unit*
GNU	*GNU Privacy Guard*
HTML	*HyperText Markup Language*
IA	*Inteligência Artificial*
ICMP	*Internet Control Message Protocol*
IPMI	*Intelligent Platform Management Interface*
IoT	*Internet of Things*
JMX	*Java Management Extensions*
QoS	*Quality of Service*
SNMP	*Simple Network Management Protocol*
TSDB	*Time-Series Database*
Wi-Fi	*Wireless Fidelity*
TI	Tecnologia da Informação

SUMÁRIO

1 INTRODUÇÃO

A Tecnologia da Informação (TI) tornou-se um componente essencial nas organizações modernas, desempenhando um papel vital na automação de processos, melhoria da eficiência e promoção da inovação. O uso de TI permite a gestão eficaz de dados, facilita a tomada de decisões e aprimora a comunicação. Avanços como computação em nuvem e inteligência artificial têm impulsionado um crescimento exponencial da TI, possibilitando que as empresas escalem suas operações de forma ágil e se adaptem rapidamente às mudanças de mercado, promovendo a transformação digital e o aumento da produtividade. Segundo (Comer, 2016), a confiabilidade e o desempenho de redes e sistemas computacionais são fatores determinantes para a continuidade operacional, especialmente em ambientes empresariais. Com esse crescimento, a quantidade de ativos de TI em uma organização aumenta substancialmente, abrangendo dispositivos como computadores, servidores, impressoras, celulares, coletores, *tablets* e uma variedade de outros ativos essenciais para o funcionamento e suporte aos processos empresariais. Manter um controle rigoroso sobre esses ativos é fundamental para garantir a segurança e a continuidade dos negócios, pois a falta de controle pode resultar em diversas vulnerabilidades de segurança e interrupções no serviço. Segundo (Forouzan; Mosharraf, 2013), a falha de um único dispositivo pode comprometer a comunicação em toda uma infraestrutura interconectada. Diante desse contexto, esse trabalho propõe uma abordagem teórica para explorar o monitoramento e a observabilidade de ativos de TI, com ênfase no uso das ferramentas *Zabbix* e *Grafana* potencializados pela inteligência artificial. No Capítulo 1, são apresentados o contexto, a justificativa e os objetivos do projeto, enfatizando a importância do monitoramento e da observabilidade no gerenciamento de ativos de TI, especialmente em ambientes corporativos que demandam alta disponibilidade e segurança. O Capí- tulo 2 desenvolve uma fundamentação teórica abrangente, abordando os conceitos de gerenciamento de redes, monitoramento, observabilidade e inteligência artificial, além da apresentação detalhada das ferramentas Zabbix e Grafana. No Capítulo 3, descrevem-se os procedimentos metodológicos adotados, incluindo o levantamento bibliográfico, o planejamento da implementação prática e o estudo sobre a aplicação da IA por meio da API Gemini. O Capítulo 4 expõe a implementação prática, desde a instalação das ferramentas e a configuração do ambiente de monitoramento até a integração do Zabbix com o Gemini e a criação de *dashboards* personalizados no Grafana. Em seguida, o Capítulo 5 apresenta os resultados obtidos com a aplicação da solução em ambiente real, com evidências visuais e análises qualitativas dos benefícios. Por fim, o Capítulo 6 traz as considerações finais, sintetizando as contribuições do trabalho e reforçando sua relevância para a área de infraestrutura de TI.

1.1 Justificativa

A justificativa para este trabalho está fundamentada na crescente necessidade de monitoramento e observabilidade dos ativos de tecnologia da informação (TI) nas organizações modernas. À medida que a TI se torna um componente essencial para a automação de processos, melhoria da eficiência e inovação, o número de ativos, como computadores, servidores e dispositivos móveis, também aumenta substancialmente. Essa expansão traz desafios significativos em termos de controle e segurança, pois a falta de monitoramento eficaz pode resultar em vulnerabilidades e interrupções nos serviços. A implementação de um processo robusto de monitoramento e observabilidade, utilizando ferramentas como *Zabbix* e *Grafana*, é crucial para garantir a disponibilidade e o desempenho contínuo dos ativos de TI. Essas ferramentas permitem uma visão centralizada e em tempo real da infraestrutura, facilitando a identificação rápida de falhas e a otimização dos recursos. Além disso, o monitoramento eficaz ajuda as organizações a tomar decisões proativas, reduzindo riscos e promovendo a resiliência da infraestrutura tecnológica. Assim, o trabalho justifica-se pela necessidade de assegurar que os ativos de TI funcionem de maneira eficiente e segura, garantindo a continuidade dos negócios e a excelência operacional em um ambiente empresarial dinâmico e em constante evolução.

1.2 Definição do Problema

Como a ausência de observabilidade adequada e de automação nos processos de monitoramento de TI pode estar contribuindo para a detecção tardia de falhas, a resolução lenta de incidentes e os impactos negativos na continuidade operacional das empresas?

1.3 Objetivo Geral

O seguinte trabalho visa evidenciar importância do monitoramento de ativos de TI por meio da utilização das ferramentas *Zabbix* e *Grafana* com auxílio da inteligência artificial, mostrando como essas soluções contribuem para uma melhor observabilidade do escopo de ativos de TI, com a mitigação de falhas e vulnerabilidades e a garantia da funcionalidade plena do negócio. O trabalho visa evidenciar como o uso dessas ferramentas proporciona maior eficiência operacional, melhora o tempo de resposta a incidentes e possibilita uma gestão estratégica da infraestrutura tecnológica, promovendo uma visão centralizada, em tempo real, e proativa sobre os ativos de TI.

1.4 Objetivos Específicos

O seguinte trabalho pretende:

a) Explorar o conceito de monitoramento e observabilidade de ativos de TI

- Detalhar as diferenças entre monitoramento e observabilidade e como ambos se aplicam à gestão de ativos tecnológicos.
- Identificar os benefícios e desafios associados a esses processos.

b) Apresentar as funcionalidades das ferramentas *Zabbix* e *Grafana*

- Descrever as principais características do *Zabbix* como ferramenta de monitoramento.
- Explicar como o *Grafana* complementa o *Zabbix*, fornecendo visualizações e *insights* de dados.

c) Demonstrar a integração prática entre *Zabbix* e *Grafana*

- Configurar e apresentar um ambiente que mostre a aplicação dessas ferramentas de forma integrada

d) Integrar inteligência artificial ao monitoramento realizado pelo Zabbix

- Utilizar scripts para conectar a API do Zabbix à API do Gemini, permitindo a análise automatizada de incidentes.
- Demonstrar como a IA pode sugerir soluções diretamente nos alertas do Zabbix, tornando o processo de resposta mais rápido e eficiente.

Após este capítulo de introdução, na sequencia apresenta-se o capítulo com a fundamentação teórica.

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste trabalho tem como objetivo estabelecer os conceitos e fundamentos que embasam o monitoramento e a observabilidade de ativos de TI, além de apresentar as ferramentas *Zabbix* e *Grafana* como soluções práticas para essas atividades. Inicialmente, aborda-se o gerenciamento de redes, destacando sua importância para a manutenção da infraestrutura tecnológica e o papel essencial que desempenha na identificação, resolução e prevenção de falhas. O trabalho também apresenta o *Zabbix* como uma ferramenta robusta de monitoramento, destacando suas principais funcionalidades, como a coleta de métricas, geração de alertas e escalabilidade para grandes ambientes. Por fim, é introduzido o *Grafana*, uma plataforma de visualização de dados que complementa o *Zabbix*, permitindo a criação de *dashboards* interativos e fornecendo *insights* detalhados em tempo real, essencial para a tomada de decisões proativas na gestão de ativos de TI.

Além disso, este trabalho considera o papel crescente da inteligência artificial no cenário do monitoramento moderno. Tecnologias de IA têm sido utilizadas para automatizar processos de análise, identificar padrões em grandes volumes de dados e oferecer respostas rápidas a incidentes. Nesse contexto, discute-se a aplicação da IA integrada ao *Zabbix*, por meio do uso da *API* do modelo Gemini, como uma estratégia para enriquecer a resposta a eventos com diagnósticos automáticos e sugestões de correção em tempo real. Essa abordagem representa uma evolução na forma como ambientes de TI são monitorados, tornando-os mais autônomos, inteligentes e proativos.

Em seguida, são explorados os conceitos de monitoramento e observabilidade, enfatizando suas diferenças e complementaridades.

2.1 Gerenciamento de Redes

Para entender um pouco o conceito de redes, vejamos o que diz o autor (Forouzan; Mosharraf, 2013):

> Uma rede é a interligação de um conjunto de dispositivos capazes de se comunicar. Nesta definição, um dispositivo pode ser um host (ou um sistema final, como às vezes é chamado), tal como um grande computador, desktop, laptop, estação de trabalho, telefone celular ou sistema de segurança. Um dispositivo nessa definição também pode ser um dispositivo de conexão, tal como um roteador, que liga uma rede a outras redes, um switch (ou comutador) que liga dispositivos entre si, um modem (modulador-demodulador) que altera a forma dos dados, e assim por diante.

O conceito de redes vai além da simples conexão entre dispositivos. Ele também envolve como esses sistemas são gerenciados, protegidos e mantidos. Isso

é essencial em um mundo cada vez mais dependente da tecnologia, desde acessar um site na internet até fazer uma chamada de vídeo depende do bom funcionamento dessas redes. Portanto, redes não são apenas uma infraestrutura tecnológica, mas também a espinha dorsal de praticamente todas as operações modernas, garantindo que as pessoas, dispositivos e sistemas permaneçam conectados, independentemente das distâncias envolvidas (Tanenbaum; Feamster; Wetherall, 2021).

Entretanto, esse conceito não se limita à interconexão de dispositivos. Ele engloba o monitoramento do desempenho, o diagnóstico de problemas e a implementação de medidas para manter a disponibilidade e a integridade da infraestrutura. "O gerenciamento de redes desempenha um papel importante na Internet à medida que ela cresce cada vez mais. A falha de um único dispositivo pode interromper a comunicação de um ponto a outro da Internet (Forouzan; Mosharraf, 2013)". Esses processos são fundamentais em organizações que dependem de redes para suportar operações críticas, garantindo confiabilidade e segurança.

O gerenciamento de redes modernas enfrenta desafios constantes, como o aumento do volume de dados, a diversidade de dispositivos conectados e a constante evolução das ameaças cibernéticas (Souza *et al.*, 2021).

Figura 1 – Previsão de gastos mundiais com TI (milhões de dólares americanos)

Categoria	Gastos 2023	Crescimento 2023 (%)	Gastos 2024	Crescimento 2024 (%)
Sistemas de Data Center	$ 236.098,00	4	$ 293.091,00	24,1
Dispositivos	$ 692.784,00	-6,5	$ 730.125,00	5,4
Software	$ 974.089,00	11,5	$1.096.913,00	12,6
Serviços de TI	$1.503.698,00	4,9	$1.609.846,00	7,1
Serviços de Comunicação	$1.491.733,00	3,2	$1.537.188,00	3
Total de TI	$4.898.401,00	3,8	$5.267.163,00	7,5

Fonte: GARTNER(2024).

A Figura 1 apresenta a previsão de gastos mundiais com TI para 2023 e 2024, conforme dados da (Gartner, 2024) evidenciando um crescimento geral de 3,8 percentuais em 2023 e uma projeção de 7,5 percentuais em 2024, totalizando cerca de 5,27 trilhões de dólares. Um destaque importante é o segmento de Dispositivos, que, apesar de ter registrado uma queda de 6,5 percentuais em 2023, aponta para uma recuperação com crescimento previsto de 5,4 percentuais em 2024. Esse movimento reflete o aumento da demanda por dispositivos como servidores, computadores, smartphones, IoT e equipamentos de rede, todos fundamentais para a operação de ambientes de TI modernos. A grande diversidade e quantidade desses dispositivos reforçam a necessidade de soluções eficientes de monitoramento, garantindo o funcionamento adequado dos ativos e a continuidade dos serviços nas organizações.

Administrar uma rede de forma eficiente exige não apenas ferramentas avançadas, mas também habilidades especializadas para lidar com a complexidade e a dinâmica das infraestruturas contemporâneas. Nesse contexto, garantir *QoS* (Qualidade de Serviço) se torna crucial, pois é necessário assegurar que os serviços essenciais recebam a prioridade adequada no tráfego de rede. Para auxiliar nesse gerenciamento, ferramentas como de monitoramento são fundamentais, pois oferecem recursos avançados de monitoramento, permitindo a análise contínua do desempenho da rede.

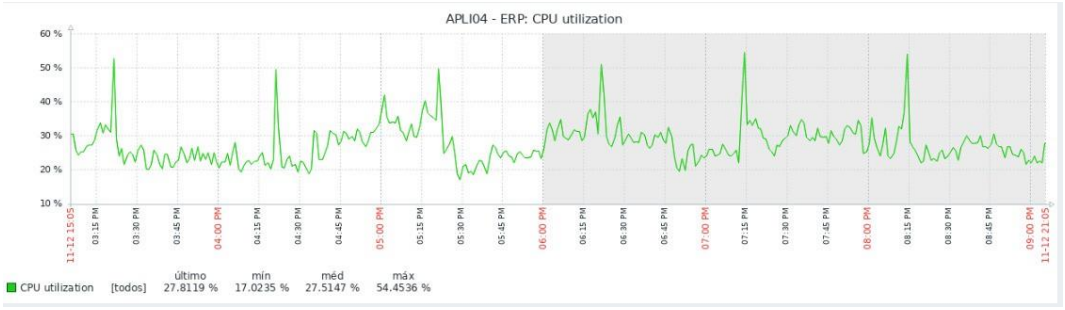
2.2 Monitoramento e observabilidade

O monitoramento é o processo contínuo de coleta, análise e acompanhamento de dados em tempo real, com o objetivo de verificar o funcionamento de sistemas, processos ou redes, visando garantir a sua operação adequada e identificar problemas ou falhas antes que se tornem críticos. Esse processo é essencial para a manutenção de um ambiente de TI eficiente, seguro e com alta disponibilidade (Comer, 2016).

Durante o monitoramento, são coletados diversos tipos de dados de um ativo. Esses dados são analisados para identificar padrões ou anomalias que possam indicar falhas iminente, sobrecarga de recursos ou até mesmo riscos de segurança. O monitoramento eficiente não só facilita a detecção de problemas, mas também proporciona uma visão detalhada sobre a saúde e o desempenho da infraestrutura de TI, o que possibilita uma resposta proativa e a otimização dos recursos.

Observabilidade no contexto de ativos de TI é a capacidade de entender o estado interno de um sistema complexo a partir de suas saídas. Em outras palavras, é a habilidade de olhar para dentro de um sistema e entender o porquê de ele estar se comportando de determinada forma (Majors; Fong-Jones; Miranda, 2022). A observabilidade transforma dados para que tenhamos uma visualização gráfica. *Dashboards* e gráficos personalizados permitem que os dados complexos sejam apresentados de forma clara e concisa, facilitando a identificação de padrões, tendências e anomalias. Métricas como uso de *CPU*, memória, latência de requisições e taxas de erro são representadas visualmente em gráficos de linha, histogramas, mapas de calor e outros tipos de visualizações.

Figura 2 – Monitoramento de utilização de *CPU*



Fonte: ZABBIX(2024).

Essa representação visual, conforme mostrada na Figura 2, permite que as equipes técnicas compreendam rapidamente o estado do sistema, detectem problemas em tempo real e tomem decisões mais assertivas para otimizar o desempenho e a disponibilidade dos serviços. A visualização gráfica é, portanto, uma ferramenta essencial para transformar dados brutos em conhecimento visualizável.

2.2.1 Ferramentas de monitoramento

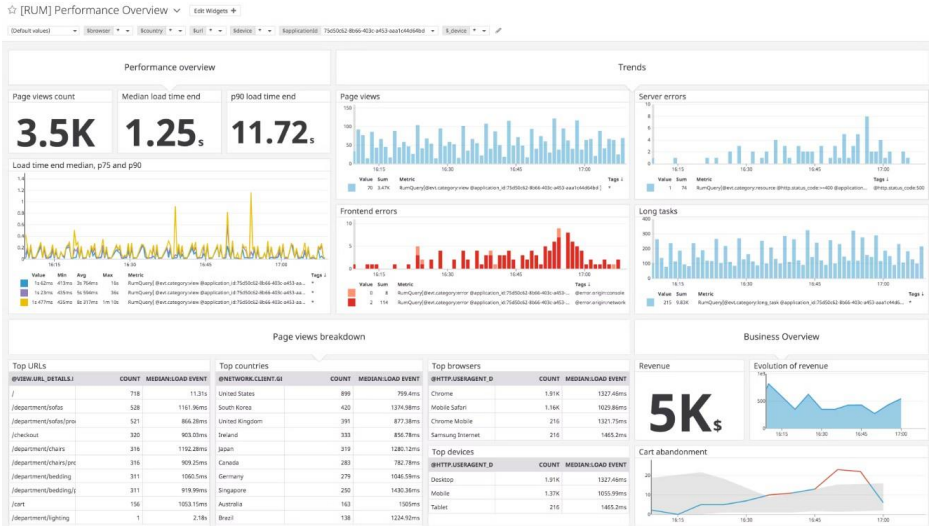
Existem diversas ferramentas de monitoramento no mercado, cada uma com suas especialidades. Essas soluções desempenham um papel fundamental na observabilidade e na gestão da infraestrutura de TI, ajudando a identificar problemas, otimizar desempenho e garantir a disponibilidade dos ativos de rede. A seguir, apresenta-se alguma destas ferramentas de monitoramento.

2.2.1.1 Datadog

O *Datadog* [1] conforme Figura 3, é uma ferramenta moderna de monitoramento em nuvem que integra métricas, logs e rastreamentos de diversas fontes. Oferece visualizações personalizáveis através de dashboards interativos e alertas em tempo real para incidentes críticos. Sua facilidade de uso e suporte a diversas plataformas fazem do *Datadog* uma escolha popular entre as equipes DevOps.

[1] Disponível em: https://www.datadoghq.com

Figura 3 – Dashboard *DataDog*



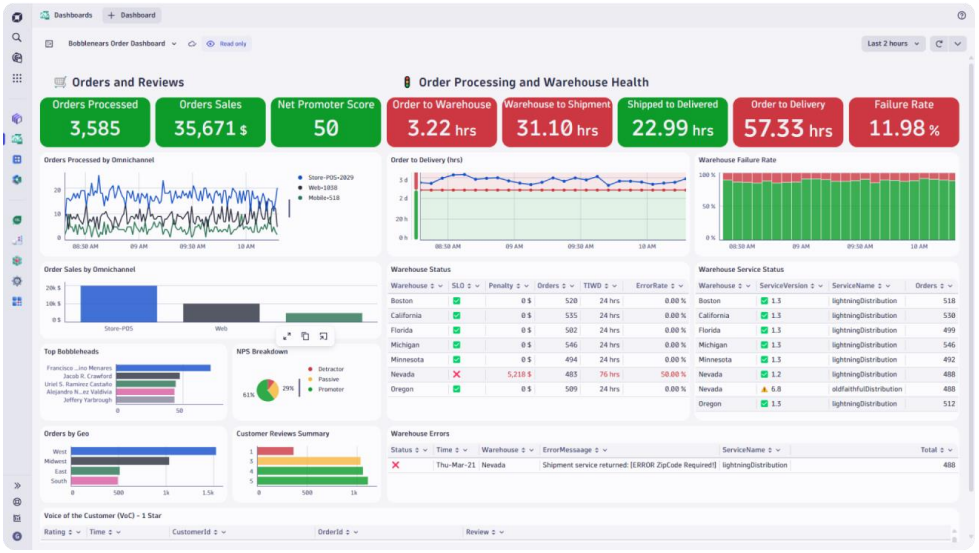
Fonte: *DATADOG*(2025).

2.2.1.2 Dynatrace

Dynatrace [2] é uma plataforma avançada de observabilidade que fornece monitoramento completo para ambientes nativos e híbridos da nuvem. Com recursos orientados por sua própria IA, a *Davis* é uma inteligência artificial hipermodal com capacidade de realizar análise preditiva, através da análise dos dados coletados, ela é capaz de analisar dados de observabilidade e agrupar as anomalias, destaca causas raízes e define prioridades com base no impacto comercial, tudo de forma automática. Possui também a *Davis Copilot*, que cria consultas, blocos de notas e painéis para simplificar a analise de dados. A *Dynatrace* oferece visibilidade completa sobre o desempenho digital. É ideal para organizações que buscam otimizar a experiência do usuário em aplicações complexas, exemplo na Figura 4.

[2] Disponível em: https://www.dynatrace.com

Figura 4 – Dashboard *DynaTrace*

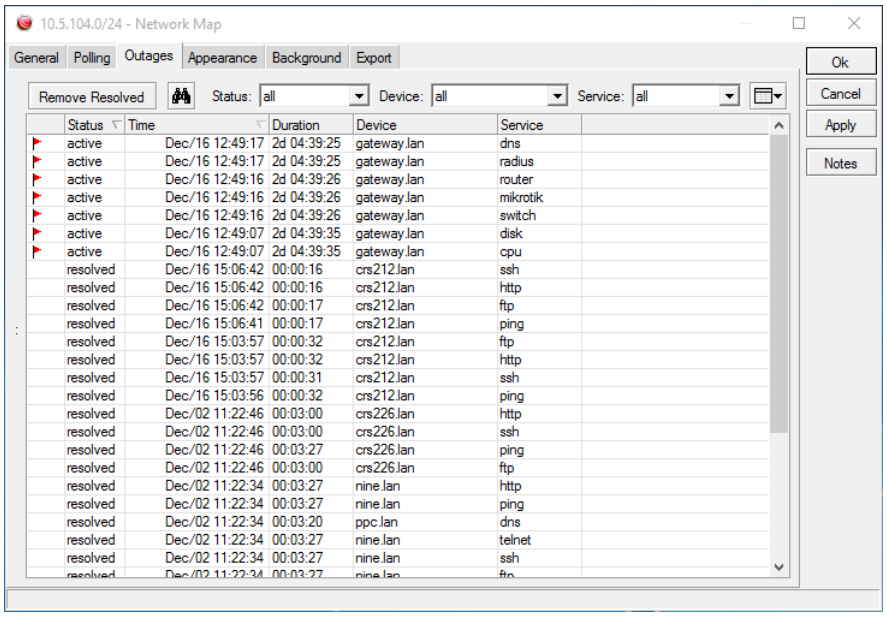


Fonte: *DYNATRACE*(2025).

2.2.1.3 The Dude

The Dude [3] conforme Figura 5, é um software de monitoramento de rede desenvolvido pela *MikroTik*, projetado para facilitar a gestão e supervisão de dispositivos e serviços em uma infraestrutura de TI. Este aplicativo permite que os usuários monitorem a disponibilidade, desempenho e estado dos equipamentos conectados à rede, oferecendo uma interface gráfica intuitiva que ajuda na visualização da topologia da rede.

Figura 5 – Mapa de Rede



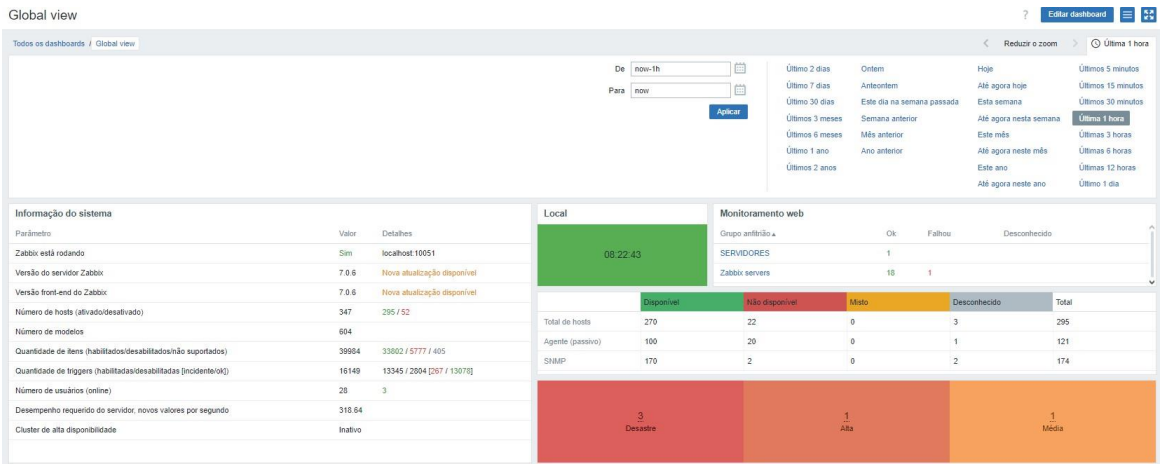
Fonte: *MIKROTIK*(2024).

[3] Disponível em: https://www.mikrotik.com/thedude

2.2.1.4 Zabbix

O *Zabbix* [4] oferece uma arquitetura robusta e modular que permite monitorar uma ampla gama de dispositivos e serviços. Sua estrutura centralizada, baseada em um servidor e agentes, garante a coleta eficiente de dados e a geração de alertas personalizados, exemplo conforme Figura 6.

Figura 6 – Dashboard Geral *Zabbix*



Fonte:ZABBIX (2024)

2.2.2 Por que escolher o Zabbix?

A escolha do *Zabbix* como ferramenta de monitoramento neste trabalho se fundamenta em diversos fatores que o tornam uma solução robusta, confiável e amplamente reconhecida no mercado. Diferente de outras opções como *Datadog*, *Dynatrace* e *The Dude*, o *Zabbix* oferece uma combinação única de características que atendem plenamente às necessidades de monitoramento de ambientes de TI modernos, especialmente para organizações que buscam soluções *open source* e altamente configuráveis. Entre os principais diferenciais do *Zabbix* destacam-se:

- Custo-benefício: O *Zabbix* é uma ferramenta gratuita e *open source*, eliminando custos com licenciamento, o que o torna acessível para empresas de todos os tamanhos.
- Flexibilidade: Possui suporte a diversos métodos de coleta de dados, como *SNMP*, *ICMP*, *IPMI*, *JMX* e agentes próprios, permitindo monitorar uma ampla variedade de dispositivos e serviços.
- Escalabilidade: O *Zabbix* é capaz de monitorar desde pequenas infraestruturas até ambientes corporativos distribuídos de grande escala, com suporte a arquiteturas distribuídas utilizando *proxies*.

[4] Disponível em: https://www.zabbix.com

- Personalização: Permite a criação de *triggers*, *templates*, *dashboards* e relatórios customizados, proporcionando um monitoramento adaptado às necessidades específicas de cada ambiente.
- Comunidade ativa e documentação ampla: O *Zabbix* conta com uma extensa comunidade de usuários e uma documentação oficial abrangente, o que facilita a resolução de dúvidas e a implementação de boas práticas.

Considerando esses fatores, o Zabbix foi escolhido como a solução principal para o monitoramento no escopo deste trabalho, por apresentar o melhor equilíbrio entre funcionalidades avançadas, flexibilidade de uso e ausência de custos de licença. Na próxima seção, será introduzido o *Zabbix* ao trabalho.

2.3 Zabbix: ferramenta de monitoramento

O *Zabbix* [5] é uma ferramenta *open source* de monitoramento. É um *software* que monitora diversos parâmetros de diversos ativos de rede diferentes. De acordo com (Lima, 2014):

> Zabbix possui a capacidade de monitorar milhares de itens em apenas um servidor, além de ser possível ter um monitoramento distribuído. Dessa forma, podemos ter um servidor central de monitoramento e vários outros servidores subordinados a ele enviando as métricas para o servidor central ou apenas replicar as informações. Também é possível separar os servidores web, servidor de banco de dados e servidor de monitora- mento para aumentar a flexibilidade e ganhar em desempenho.

Segundo (Lima, 2014) o *Zabbix* teve sua primeira versão licenciada lançada em 2001, na forma da versão 0.1 *alpha*. Em 2004, foi apresentada a versão estável 1.0. No ano seguinte, 2005, a empresa *Zabbix SIA* foi fundada com o objetivo de profissionalizar a gestão e o desenvolvimento da ferramenta. Desde 2006, o *Zabbix* passou por uma série de evoluções que o transformaram no sistema robusto que conhecemos atualmente.

2.3.1 Características do Zabbix

O *Zabbix* centraliza a coleta e o processamento de dados provenientes de agentes instalados em dispositivos monitorados. Esses agentes são leves, eficientes e garantem a coleta de métricas como *CPU*, memória, tráfego de rede, entre outros parâmetros. Além disso, o *Zabbix* também suporta monitoramento sem agentes, utilizando protocolos como *SNMP*, *ICMP*, *IPMI*, *JMX*, além do *Zabbix Agent*. Cada um desses protocolos tem uma aplicação específica para diferentes tipos de dispositivos e métricas, ampliando a abrangência da ferramenta, conforme descrito a seguir:

[5] Disponível em: https://www.zabbix.com

- *SNMP* (*Simple Network Management Protocol*): Ideal para monitorar dispositivos de rede, como switches, roteadores e outros equipamentos de infraestrutura.
- *ICMP* (*ping*): Utilizado para verificar a conectividade e latência de *hosts*, permitindo identificar problemas de rede.
- *IPMI* (*Intelligent Platform Management Interface*): Focado no monitoramento de hardware, especialmente em servidores, proporcionando dados como temperatura, voltagem e *status* de componentes críticos.
- *JMX* (*Java Management Extensions*): Utilizado para monitorar aplicativos baseados em *Java*, oferecendo visibilidade em tempo real sobre o desempenho desses aplicativos.
- *Zabbix Agent* : Coleta dados diretamente de servidores ou estações de trabalho, garantindo informações detalhadas e em tempo real sobre o desempenho do sistema.

  Esses protocolos e agentes permitem que o *Zabbix* seja flexível e abrangente, monitorando não apenas servidores, mas também uma variedade de dispositivos de rede e sistemas diversos, otimizando a gestão e a análise de infraestrutura de TI.

2.3.1.1 Escalabilidade e Flexibilidade

O *Zabbix* é altamente escalável, permitindo monitorar desde pequenos ambientes de TI até grandes infraestruturas corporativas distribuídas. Ele suporta a implementação de uma arquitetura distribuída, onde vários servidores podem ser utilizados para coletar dados de diferentes locais e enviá-los para um servidor central. Essa abordagem aumenta a flexibilidade e melhora o desempenho, permitindo o monitoramento de grandes volumes de dados de forma eficiente.

2.3.1.2 Monitoramento de Performance e Alertas

Além da coleta de dados, o *Zabbix* oferece funcionalidades de análise de desempenho e geração de alertas. Ele permite configurar *triggers* (gatilhos) que disparam alertas em tempo real quando determinadas condições são atendidas, como, por exemplo, quando o tráfego de rede ultrapassa um limite preestabelecido. Esses alertas podem ser enviados por e-mail, SMS, Telegram, Whatsapp ou outras formas de notificação, permitindo que os administradores de rede tomem medidas corretivas rapidamente.

2.3.1.3 Armazenamento e Visualização de Dados

O *Zabbix* armazena os dados coletados em banco de dados e oferece ferramentas para análise e visualização, como gráficos e *dashboards* customizáveis. Esses gráficos permitem que os administradores visualizem a performance dos sistemas ao longo do tempo, facilitando a detecção de tendências e problemas recorrentes.

2.3.1.4 Segurança e Controle de Acesso

O *Zabbix* também se destaca por suas funcionalidades de segurança. Ele permite a configuração de diferentes níveis de acesso para os usuários, garantindo que apenas administradores ou operadores específicos possam modificar configurações sensíveis ou visualizar certos tipos de dados. Além disso, o *Zabbix* suporta criptografia de comunicação entre agentes e servidores, assegurando a proteção dos dados transmitidos na rede.

2.3.2 Arquitetura do *Zabbix*

O *Zabbix* possui uma arquitetura centralizada, onde servidor central, ou os *proxies* coletam, processam e armazenam os dados. Essa arquitetura é modular, o que significa que os componentes do *Zabbix* podem ser distribuídos e dimensionados conforme a necessidade do ambiente monitorado. A principal estrutura do *Zabbix* é composta por:

- Servidor *Zabbix* : Este é o componente principal da arquitetura. O servidor *Zabbix* é responsável por coletar dados dos dispositivos monitorados, processá-los e gerar alertas quando as condições configuradas são atendidas. Ele também armazena todas as informações sobre os dispositivos monitorados e eventos gerados.
- O servidor *Zabbix* armazena as informações coletadas, como histórico de métricas e eventos, em um banco de dados relacional, como MySQL, MariaDB ou outros. O banco de dados é crucial para garantir a persistência das informações, permitindo consultas e geração de relatórios históricos.
- O banco de dados do *Zabbix* atua como o núcleo do sistema, armazenando todas as informações coletadas, como métricas de monitoramento, configurações, histórico de eventos e registros de usuários. O MariaDB é uma das opções mais populares para bancos de dados relacionais no Zabbix, devido ao seu desempenho, escalabilidade e compatibilidade com MySQL. O servidor Zabbix recebe os dados dos protocolos coletados e o inserem no banco de dados em tempo real, conforme configurado previamente no *Zabbix*. As informações são gravadas nas tabelas que posteriormente são processadas e analisadas. O

Zabbix mantém os dados retidos por um período configurável. Dados antigos são removidos para evitar sobrecarga no banco.

- *Front-end Web*: O *Zabbix* fornece uma interface *web* que permite aos administradores e operadores visualizar dados em tempo real, gerar gráficos, configurar alertas e gerenciar a infraestrutura monitorada. O *front-end* se comunica com o servidor *Zabbix*, permitindo a gestão remota e centralizada dos ativos de TI.
- *Proxy Zabbix* : O *Zabbix* também pode usar *proxies*, que são servidores intermediários responsáveis por coletar dados de dispositivos localizados em locais remotos ou distribuídos. Os *proxies* enviam esses dados para o servidor *Zabbix* central. Isso é útil quando há a necessidade de monitorar grandes ambientes ou locais geograficamente dispersos, melhorando a escalabilidade e a performance do monitoramento.

Assim, a arquitetura modular e escalável do *Zabbix* proporciona uma base sólida para a coleta, armazenamento e análise de métricas de diversos ativos de TI. No entanto, para aprimorar ainda mais a análise e a visualização desses dados, é fundamental integrar uma solução dedicada à criação de *dashboards* e *insights* visuais. Nesse contexto, o *Grafana* surge como uma poderosa ferramenta de observabi- lidade, complementando as capacidades do *Zabbix* ao transformar dados técnicos em representações gráficas interativas e intuitivas, conforme será apresentado a seguir.

2.4 Grafana: ferramenta de observabilidade

Grafana [6] é uma ferramenta *open source* para visualização e análise de dados. Ela possibilita consultar, visualizar, criar alertas e explorar métricas, *logs* e *traces*, independentemente do local de armazenamento. Com seus recursos avançados, *Grafana* transforma dados de bancos de séries temporais (TSDB) em gráficos e visualizações interativas, proporcionando *insights* claros e detalhados para análise e tomada de decisão.

2.4.1 Características do Grafana

O *Grafana*, conforme Figura 7, oferece a possibilidade de criar *dashboards* interativos e personalizados que permitem a visualização de métricas em tempo real. Esses *dashboards* são intuitivos e flexíveis, possibilitando a combinação de diferentes tipos de gráficos, tabelas e painéis para atender às necessidades específicas dos usuários.

[6] Disponível em: https://www.grafana.com

Figura 7 – Monitoramento de Celulares

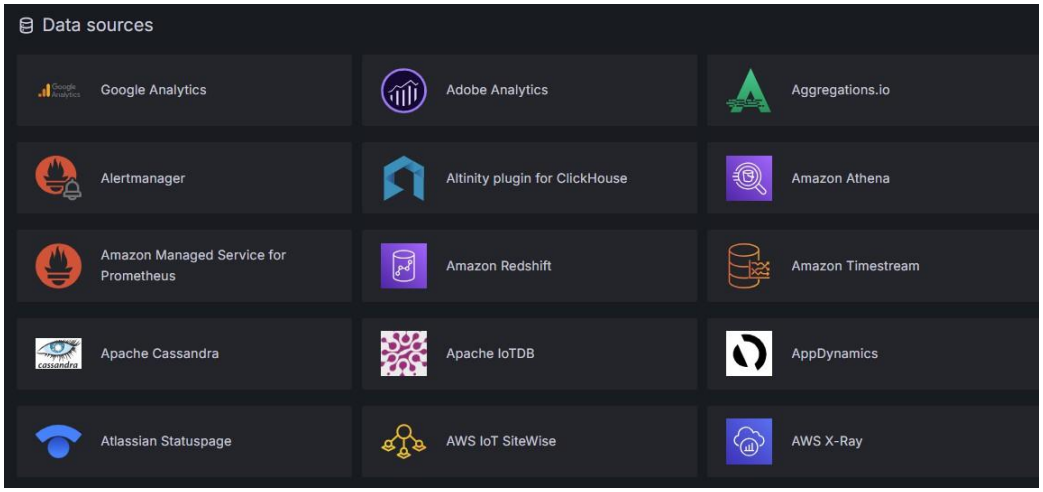
Fonte:GRAFANA (2024)

2.4.1.1 Suporte a Múltiplas Fontes de Dados

Uma das principais vantagens do *Grafana* é seu suporte a diversas fontes de dados, como bancos de dados e *plugins*, conforme podemos visualizar na Figura 8. Entre elas, as mais populares:

- *Prometheus;*
- *Elasticsearch;*
- *Zabbix;*
- *MySQL;*
- *PostgreSQL;*
- *InfluxDB.*

Figura 8 – Data Sources Grafana

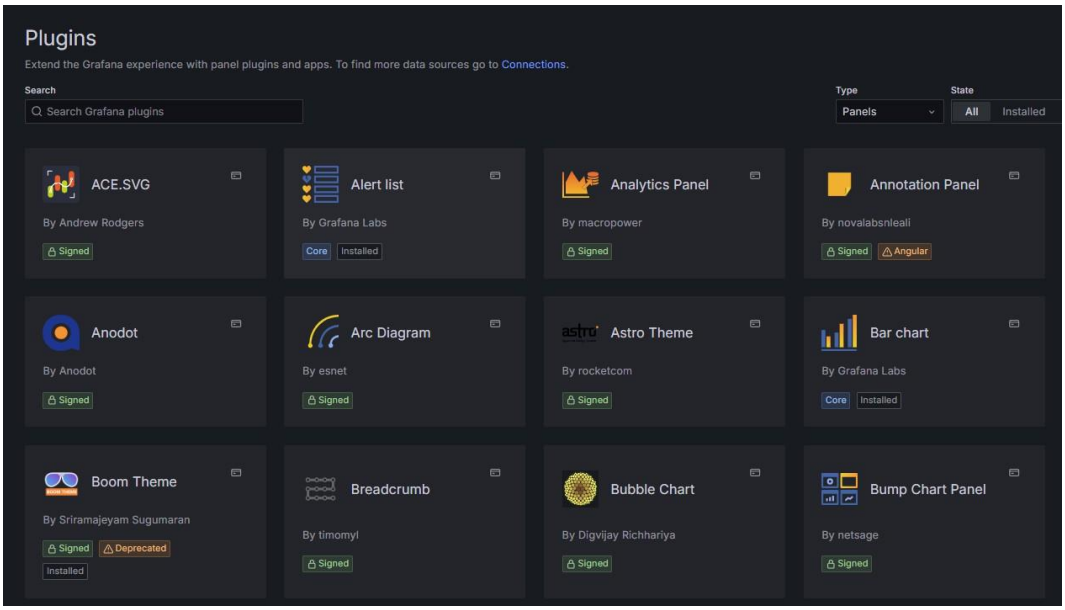


Fonte:GRAFANA (2024)

2.4.1.2 Extensibilidade por Meio de Plugins

A ferramenta possui uma grande variedade de *plugins* disponíveis, tanto oficiais quanto desenvolvidos pela comunidade. Esses *plugins* expandem as funcionalidades do *Grafana*, adicionando novos tipos de gráficos, conectores de fontes de dados e painéis prontos para uso, exemplo de *plugins* na Figura 9.

Figura 9 – *Plugins Grafana*



Fonte:GRAFANA (2024)

2.4.2 Por que usar o *Grafana*?

Uma das principais vantagens do *Grafana* é sua capacidade de consolidar as informações de monitoramento e observabilidade em um único ambiente visual. Essa centralização não se limita à visualização de métricas, *logs* e *traces*, mas também à unificação de fontes de dados heterogêneas, o que permite uma visão integrada e simplificada da infraestrutura monitorada.

O *Grafana* funciona como um ponto de convergência para os dados provenientes de diversas ferramentas e sistemas, eliminando a necessidade de alternar entre diferentes plataformas para coletar informações específicas. Essa abordagem integrada facilita a correlação entre métricas e eventos, contribuindo para a rápida identificação de padrões, anomalias e causas-raiz de problemas em ambientes complexos.

Dessa forma, o Grafana se consolida como uma ferramenta essencial para transformar as informações coletadas pelo Zabbix em dashboards dinâmicos e visualizações eficientes. Combinadas, essas duas soluções oferecem uma abordagem integrada que potencializa o monitoramento e a observabilidade dos ativos de TI. A seguir, será apresentada a proposta de implementação e integração prática entre o

Zabbix e o Grafana, evidenciando como essa união pode ser aplicada de maneira eficiente em ambientes de tecnologia.

2.5 *Zabbix* e *Grafana*: Implementação e Uso

Neste trabalho, será realizada a implementação prática das ferramentas *Zabbix* e *Grafana*, com o objetivo de demonstrar suas funcionalidades e os benefícios que podem oferecer em um ambiente de TI. Para isso, será utilizado uma maquina virtual para implementar os softwares, permitindo a configuração, integração e análise detalhada das ferramentas.

A implementação terá como foco principal a configuração do *Zabbix* como solução de monitoramento, coletando métricas de desempenho de servidores e dispositivos de rede. Em seguida, será realizada a integração com o *Grafana*, que será responsável por transformar os dados coletados em *dashboards* interativos e visuais. Essa abordagem possibilitará uma análise clara e objetiva, evidenciando o impacto positivo dessas ferramentas na gestão de ativos de TI.

Essa etapa prática complementará a fundamentação teórica apresentada até agora, oferecendo uma visão aplicada e consolidando a relevância das ferramentas *Zabbix* e *Grafana* no contexto do monitoramento e da observabilidade em ambientes de TI.

No próximo seção, apresenta-se as considerações sobre Inteligência Artificial.

2.6 Inteligencia Artificial

A inteligência artificial (IA) é um campo da ciência da computação dedicado à criação de sistemas capazes de simular comportamentos inteligentes, como aprender com dados, tomar decisões, reconhecer padrões e interagir por meio da linguagem natural. Segundo (Russell; Norvig, 2013), a IA envolve o desenvolvimento de agentes racionais que percebem seu ambiente e tomam decisões para maximizar suas chances de sucesso em determinada tarefa.

Entre os principais ramos da IA estão o *machine learning* (aprendizado de máquina) e o *deep learning* (aprendizado profundo). O primeiro consiste em algoritmos capazes de extrair conhecimento a partir de dados e aprimorar seu desempenho com o tempo, sem serem reprogramados explicitamente (Mitchell, 1997). Já o aprendizado profundo, uma subárea mais recente e poderosa, utiliza redes neurais com múltiplas camadas para realizar tarefas complexas como reconhecimento de voz e imagem, conforme explicam (Goodfellow; Bengio; Courville, 2016).

Outra área fundamental da IA é o *processamento de linguagem natural*

(NLP), que permite que máquinas compreendam e gerem textos em linguagem humana. Essa tecnologia é a base de assistentes virtuais, sistemas de busca e modelos generativos modernos, como o Gemini e o ChatGPT. Segundo (Domingos, 2017), o avanço desses modelos representa um passo importante para a criação de sistemas capazes de aprender qualquer tarefa intelectual realizada por humanos.

No contexto da tecnologia da informação, a IA tem sido amplamente utilizada para automatizar a análise de grandes volumes de dados, prever falhas, detectar anomalias e responder a incidentes com maior agilidade e precisão. A aplicação da IA em ferramentas de monitoramento permite transformar processos reativos em processos proativos e inteligentes.

Na próxima etapa deste trabalho, junto a implementação do *Zabbix* e *Grafana*, será realizada a integração do Zabbix com inteligência artificial, utilizando uma API baseada no modelo Gemini, desenvolvido pela Google (Google, 2025). Essa ferramenta de IA será responsável por analisar os incidentes detectados pelo Zabbix e fornecer respostas automatizadas e contextualizadas sobre como solucioná-los.

O Gemini atuará como um complemento ao Zabbix, processando os da- dos dos alertas gerados e oferecendo recomendações práticas para a resolução de problemas em tempo real. Em situações como sobrecarga de servidores ou falhas de conectividade, o modelo poderá sugerir ações corretivas, como reinicialização de serviços, ajustes de configuração ou verificações adicionais.

Essa integração visa automatizar parte do processo de diagnóstico e resposta a incidentes, reduzindo significativamente o tempo necessário para solução de problemas e aumentando a eficiência operacional. O objetivo é demonstrar como a aplicação de inteligência artificial, integrada a ferramentas como o Zabbix, pode transformar a gestão de ativos de TI, tornando-a mais ágil, proativa e eficiente.

3 PROCEDIMENTOS METODOLÓGICOS

O desenvolvimento deste trabalho envolveu uma revisão bibliográfica detalhada sobre os conceitos de monitoramento e observabilidade, bem como sobre as ferramentas *Zabbix* e *Grafana*. Foram consultadas fontes como livros especializados, artigos acadêmicos, guias técnicos, publicações científicas e a documentação oficial dessas ferramentas. O objetivo foi identificar os fundamentos teóricos, as melhores práticas e as aplicações mais relevantes para o tema abordado.

Além da revisão bibliográfica e do planejamento da implementação do Zabbix e do Grafana, o trabalho também contempla o estudo e a integração de inteligência artificial (IA) ao ambiente de monitoramento. Para isso, foi analisada a utilização do modelo Gemini, desenvolvido pelo Google DeepMind, como ferramenta de apoio na análise e resposta a incidentes detectados pelo Zabbix.

O procedimento metodológico para a integração da IA inclui:

- Estudo do funcionamento da *API* do *Gemini* e sua aplicação no contexto de monitoramento de TI;
- Desenvolvimento de um script capaz de receber eventos do Zabbix, enviar as informações para o Gemini e retornar recomendações automatizadas de solução para incidentes;
- Planejamento de testes simulados, com incidentes fictícios, para avaliar a precisão e a eficiência das respostas fornecidas pela IA;
- Avaliação qualitativa do impacto da IA no tempo de resposta e na assertividade da resolução de problemas em ambientes monitorados.

Essa abordagem busca inovar o processo tradicional de monitoramento, agregando inteligência analítica às ferramentas utilizadas e proporcionando um modelo mais proativo e eficiente no monitoramento de ativos de TI.

Nesta etapa, também foi realizado um planejamento detalhado para a aplicação do *Zabbix* e do *Grafana*. Esse planejamento incluiu:

- Definição de um ambiente virtual para implementação dos *softwares*;
- Estudo da ferramenta *Zabbix* para a criação de monitoramento de ativos de TI;
- Estudo da ferramenta *Grafana* para a criação de *dashboards* e telas de forma a complementar a coleta de dados realizados pelo *Zabbix*;
- Estudo de sistemas operacionais Linux para implementação das ferramentas;

- Análise de métricas e parâmetros que serão monitorados, como uso de *CPU*, memória, tráfego de rede e disponibilidade de serviços;
- Estudo preliminar sobre a integração das ferramentas com inteligência artifi- cial, utilizando o *Gemini* para fornecer respostas automatizadas aos incidentes identificados.

Para embasar as análises e propostas, foram estudados dados secundários a partir das fontes bibliográficas e documentações. Esses dados incluem:

- Funcionalidades específicas e características técnicas do *Zabbix* e do *Grafana*;
- Relatos de casos e exemplos de uso prático dessas ferramentas em empresas;
- Estudos sobre a aplicação de inteligência artificial no contexto de monitoramento e observabilidade

Após a coleta e organização das informações, os dados foram analisados qualitativamente, buscando identificar padrões, lacunas e oportunidades para aplicação prática. Essa análise teórica serviu de base para a elaborar de um plano detalhado, utilizado na implementação prática.

A escolha por uma abordagem exploratória e descritiva foi feita devido à necessidade de compreender a fundo os conceitos e ferramentas envolvidas, estabelecendo um embasamento sólido para o desenvolvimento do projeto. Esse método também permitiu identificar potenciais desafios e oportunidades, garantindo que a implementação seja bem estruturada e alinhada aos objetivos do trabalho.

O capítulo seguinte tem como objetivo detalhar a implementação prática do Zabbix e do Grafana.

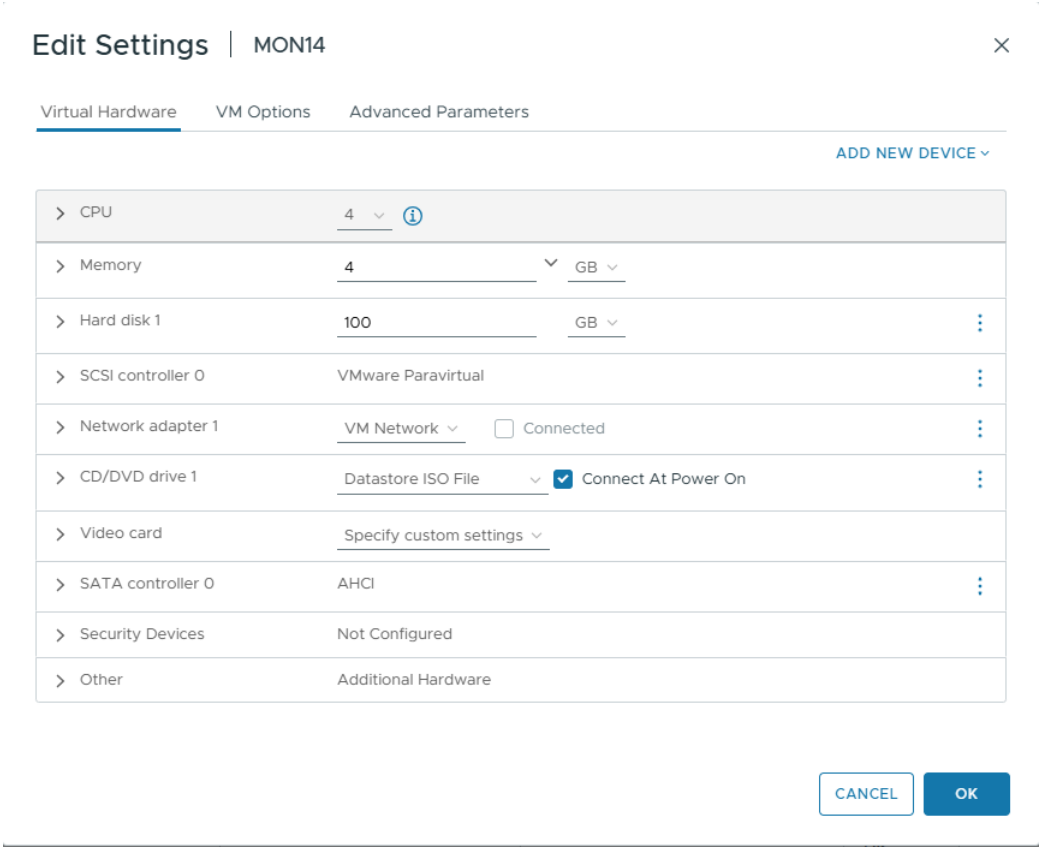
4 Implementação Prática Zabbix e Grafana

Após elaborar a fundamentação teórica e do planejamento metodológico, esta etapa do trabalho dedica-se à implementação prática das ferramentas Zabbix e Grafana, com o objetivo de validar os conceitos estudados e demonstrar, de forma aplicada, como essas soluções podem ser utilizadas no monitoramento e na observabilidade de ativos de TI. A seguir, são apresentados os detalhes técnicos da infraestrutura utilizada e cada uma das etapas do processo de implementação, bem como os principais resultados observados.

4.1 Estrutura do Ambiente de Implementação

A implantação foi realizada em uma máquina virtual, executada em um servidor gerenciado por meio do *hypervisor* da *VMware*, conforme visualizamos na Figura 10. Essa máquina virtual foi configurada com 4 *vCPUs*, 4 GB de memória *RAM* e 100 GB de armazenamento em disco. Para o sistema operacional, foi adotada a distribuição *Debian GNU/Linux*, na versão 12 *"Bookworm*", com *kernel* 6.1. A escolha desse sistema se deu por sua estabilidade, leveza e ampla compatibilidade com os softwares utilizados no projeto. Para facilitar o gerenciamento remoto da máquina virtual e a execução dos comandos de instalação e configuração, foi utilizado o software *PuTTY*, que permite o acesso via protocolo SSH ao servidor Debian. Com essa ferramenta, todas as etapas de implementação foram realizadas a partir de um terminal remoto, garantindo praticidade e controle total sobre o ambiente de testes.

Figura 10 – Tela de Configuração da Maquina virtual no VMWARE



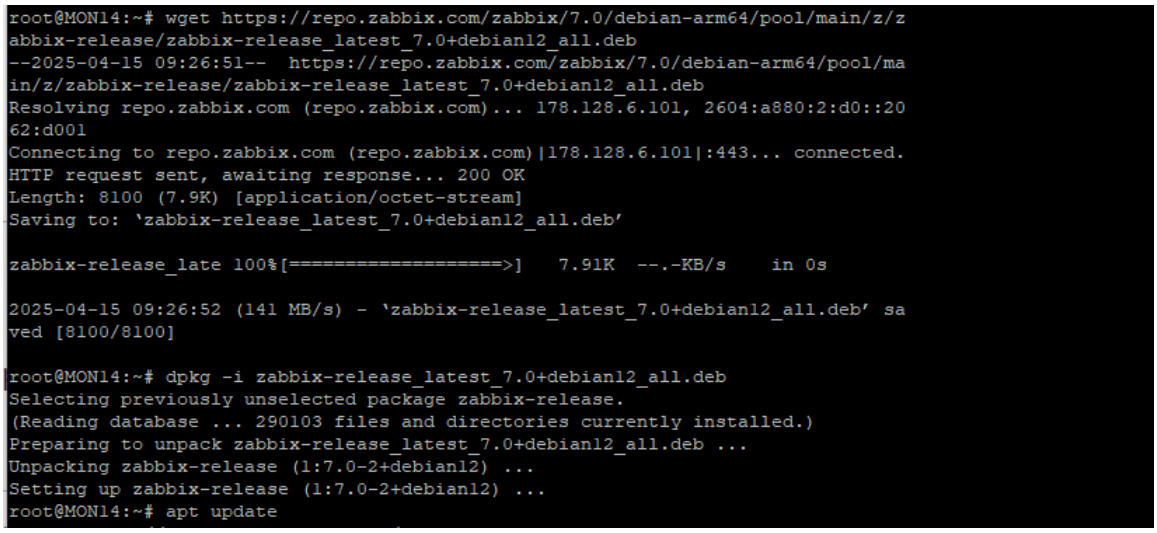
Fonte: *VMWARE* (2025)

A instalação do sistema operacional foi feita com base em uma imagem ISO oficial do Debian, que é disponibilizada no site oficial da distribuição em debian.org. A máquina virtual foi configurada para operar com IP fixo dentro da rede NAT gerenciada pela VMware, garantindo conectividade estável para o gerenciamento remoto e instalação de dependências via internet.

4.2 Instalação e Configuração do Zabbix

Com o servidor e o sistema operacional devidamente configurados, deu-se início à instalação do Zabbix. Primeiramente, foi adicionado o repositório oficial da versão 7.0 da ferramenta para Debian 12, conforme Figura 11. Os repositórios podem ser encontrados na documentação oficial do *Zabbix*.

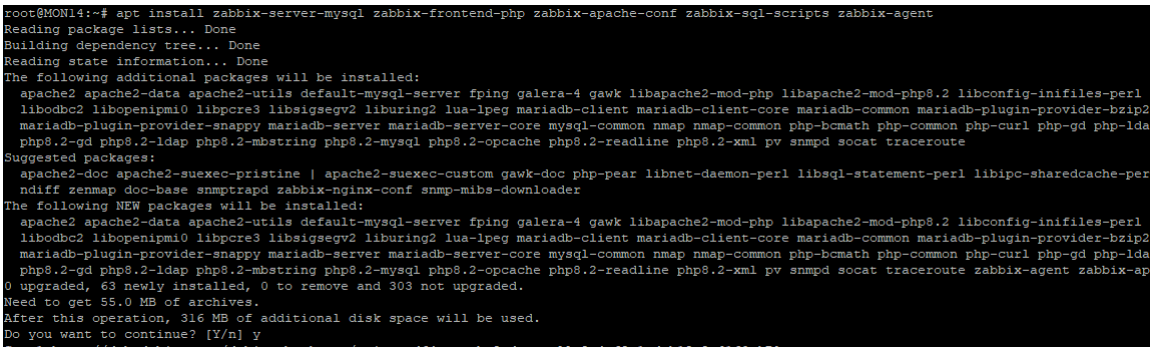
Figura 11 – Execução de comandos para instalação dos repositórios do Zabbix.



Fonte: AUTOR (2025)

O proximo passo é a instalação do *Zabbix Server*, *frontend* e *Agent*, conforme Figura 12.

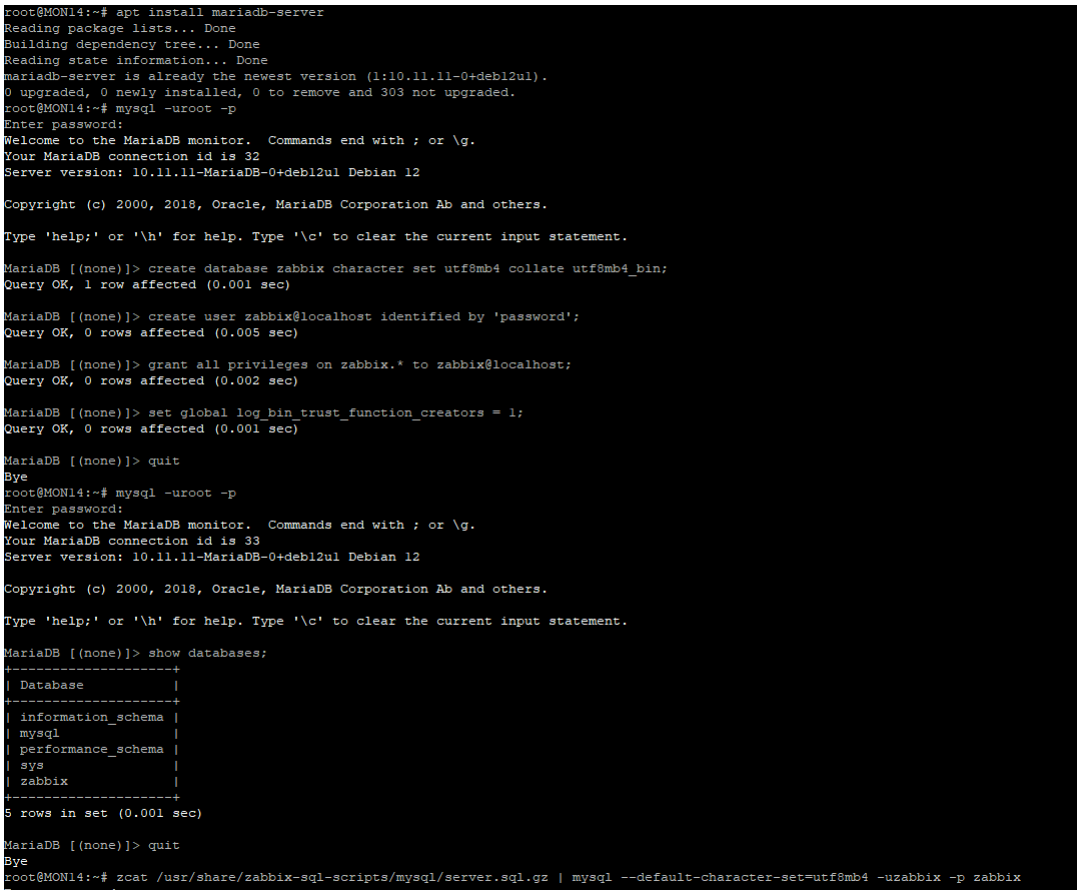
Figura 12 – Execução de comandos para instalação do Zabbix Server, Frontend, e Agent.



Fonte: AUTOR (2025)

Na Figura 13, procede-se à instalação do sistema gerenciador de banco de dados MariaDB, que será responsável por armazenar todas as informações operacionais do *Zabbix*. Após a instalação, são realizadas as configurações essenciais para garantir a integração entre o banco de dados e o servidor *Zabbix*, permitindo o correto funcionamento da ferramenta.

Figura 13 – Execução de comandos para criação do banco de dados e tabelas.

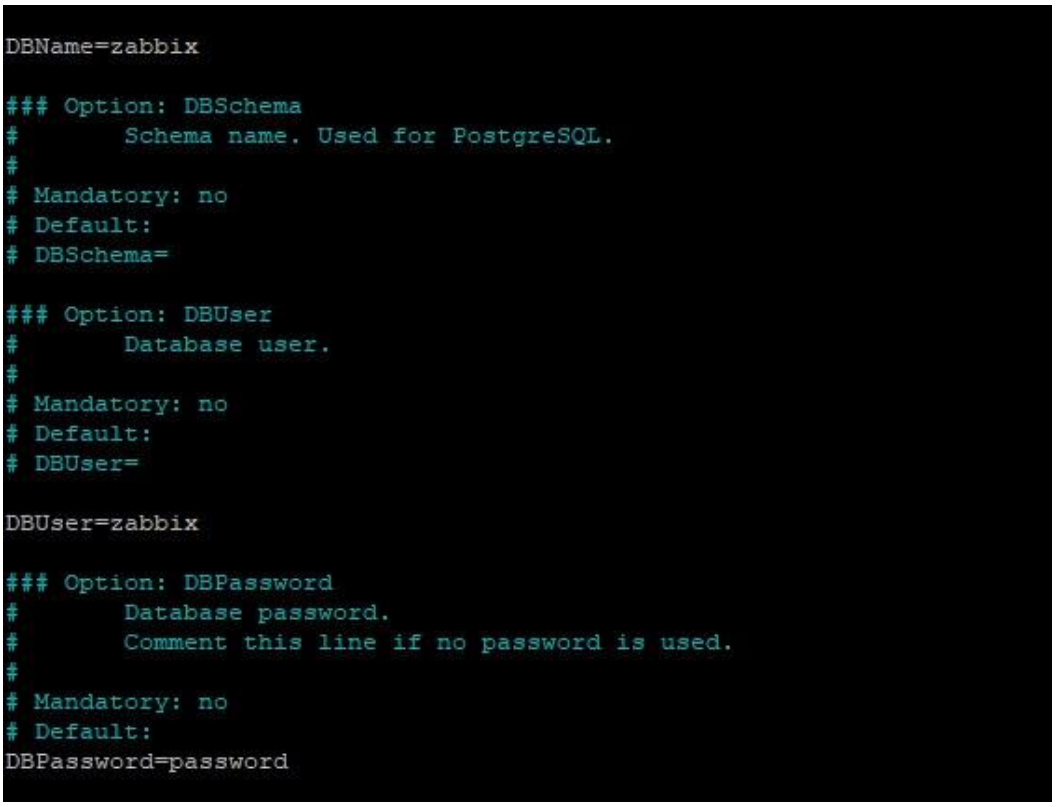


Fonte: AUTOR (2025)

Os comandos criam um usuário chamado *zabbix* com permissão total no banco e ajustam a configuração global do *MySQL* para garantir a compatibilidade com o *Zabbix*.

Para finalizar as configurações do banco, é necessário validar os campos DBName e DBPassword do arquivo de configuração do *Zabbix* conforme Figura 14.

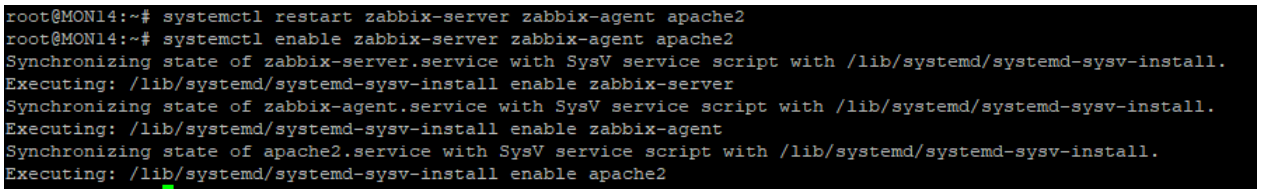
Figura 14 – Editando o arquivo de configuração do Zabbix Server.



Fonte: AUTOR (2025)

Na Figura 15, é executado os comandos para inicialização do servidor *web*.

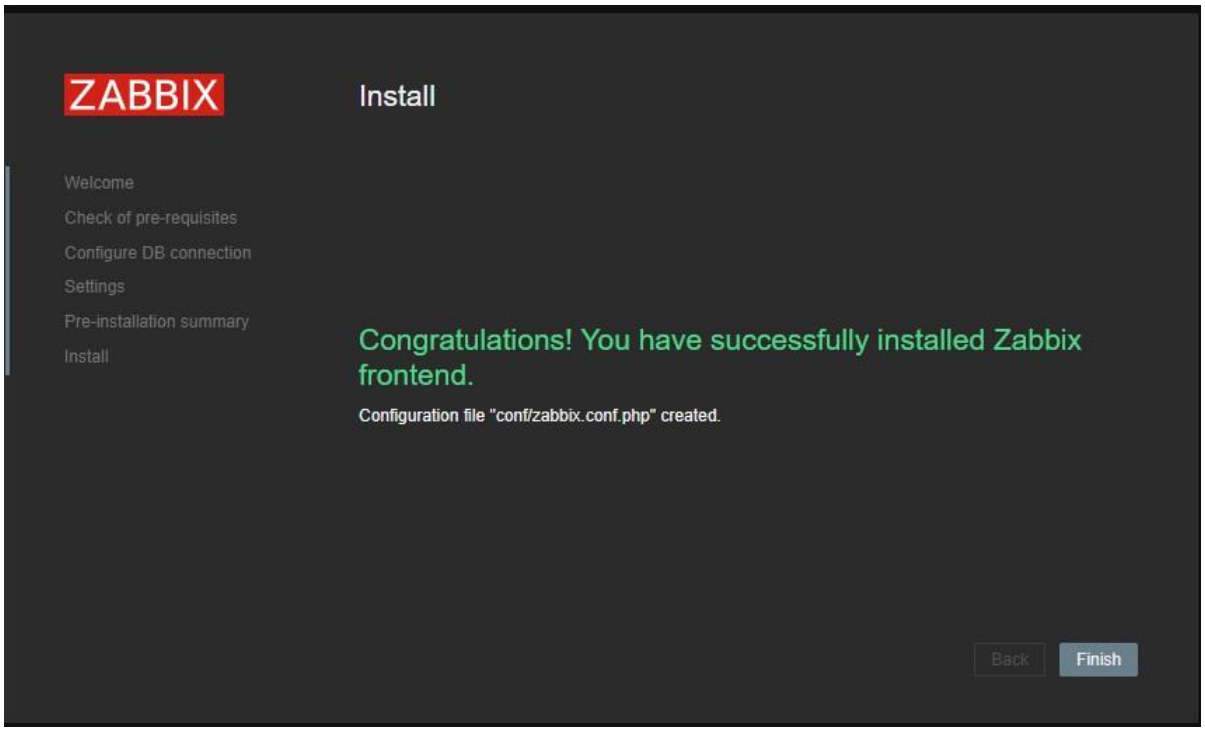
Figura 15 – Execução de comandos para inicialização do servidor web.



Fonte: AUTOR (2025)

A instalação é finalizada por meio da interface *web* do *Zabbix*, conforme conseguimos visualizar na Figura 16. A interface web é acessada através do *IP* do servidor e o sufixo /zabbix, ou seja, http://<IPDOSERVIDOR>/zabbix. Essa interface gráfica permite concluir a configuração inicial da ferramenta, definindo o idioma, o fuso horário, as credenciais administrativas e os parâmetros de conexão ao banco. Com o sistema operacional e o *Zabbix* plenamente operacionais, já é possível iniciar o monitoramento. Na próxima seção, será iniciado o monitoramento dos ativos de TI.

Figura 16 – Tela de finalização da configuração do Zabbix.



Fonte: ZABBIX (2025)

4.3 Instalação do Agente *Zabbix*

O agente *Zabbix* coleta dados cruciais, como uso de *CPU*, memória, disco e status de serviços, enviando essas informações ao servidor *Zabbix*. A seguir, detalha-se a instalação do agente em sistemas Linux e Windows.

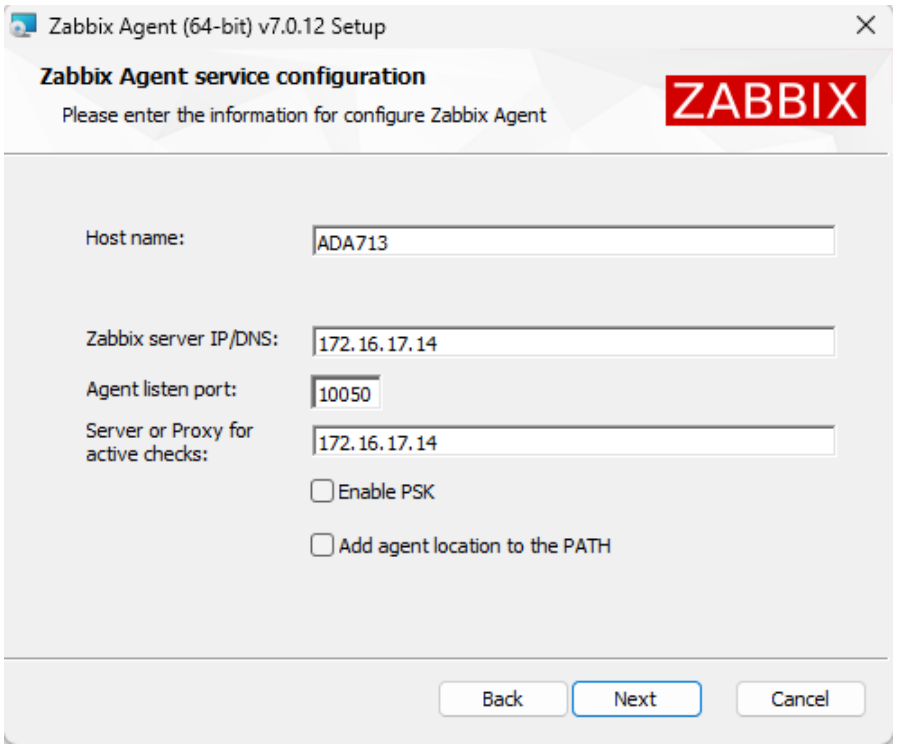
4.3.1 Instalação em ambiente Windows

No *Windows*, o agente pode ser baixado diretamente do site oficial do *Zabbix*. O processo consiste em:

- Baixar o instalador do agente compatível com a versão do Windows (32 ou 64 bits).
- Executar o instalador com privilégios de administrador.
- Durante a instalação, configurar os campos "endereço do servidor", e "nome do *host*".
- Após a instalação, o serviço *Zabbix Agent* é criado automaticamente.

Na figura 17, conseguimos visualizar o instalador do agente *Zabbix*.

Figura 17 – Instalação do agente Zabbix no Windows.



Fonte: ZABBIX (2025)

O próximo tópico abordará a instalação do agente em ambientes *Linux*.

4.3.2 Instalação em ambiente Linux

Em sistemas baseados em Debian, como utilizado no projeto, a instalação do agente pode ser realizada com os seguintes comandos demonstrados na Figura 18.

Figura 18 – Instalação do agente Zabbix no Debian.

```
sudo apt update
sudo apt install zabbix-agent
```

Fonte: AUTOR (2025)

Após a instalação, o arquivo de configuração do agente deve ser editado conforme figura 19.

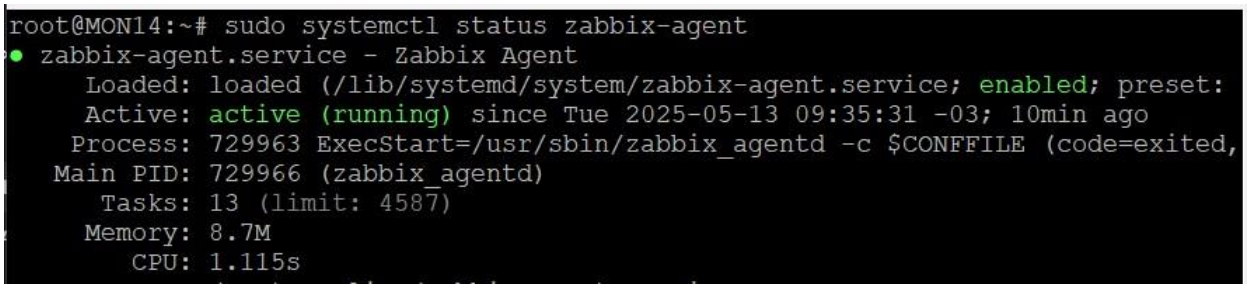
Figura 19 – Comando para editar configurações do agente.

```
sudo nano /etc/zabbix/zabbix_agentd.conf
```

Fonte: AUTOR (2025)

E depois é verificado se o agente está ativo com o seguinte comando demonstrado na figura 20.

Figura 20 – Comando para verificar o status do agente.



Fonte: AUTOR (2025)

Dessa forma, o agente foi instalado com sucesso e está pronto para se comunicar com o *Zabbix*. Na próxima seção, vamos abordar a inserção de *hosts* e configurações essenciais para o monitoramento no *Zabbix*.

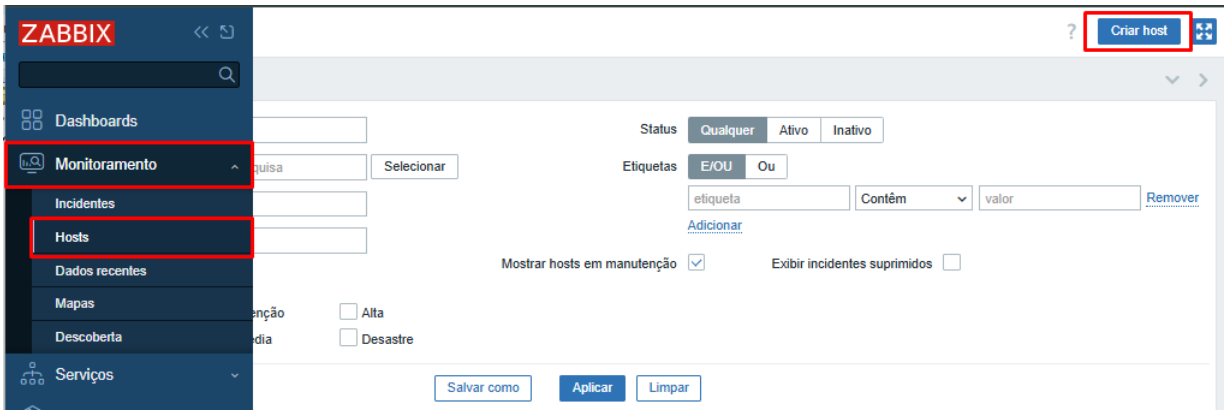
4.4 Inserção de *hosts* e configurações fundamentais para o monitoramento

Nesta seção será abordado o processo de inserção de *hosts* no *Zabbix* e a configuração dos parâmetros essenciais para o monitoramento dos ativos de TI. A etapa é fundamental para garantir que os dispositivos e serviços estejam devidamente cadastrados, com as métricas corretas sendo coletadas, os alertas devidamente configurados e as respostas a incidentes automatizadas conforme necessário.

4.4.1 Cadastros de *Hosts* no *Zabbix*

Após a instalação do *Zabbix*, o próximo passo é registrar os ativos que se deseja monitorar. Esse processo é realizado por meio do *frontend* web da ferramenta, acessível via navegador. Na interface, deve-se acessar a opção “Monitoramento > *Hosts*” e clicar em “Criar *Host*” conforme Figura 21.

Figura 21 – Criação de *host* no *Zabbix*.



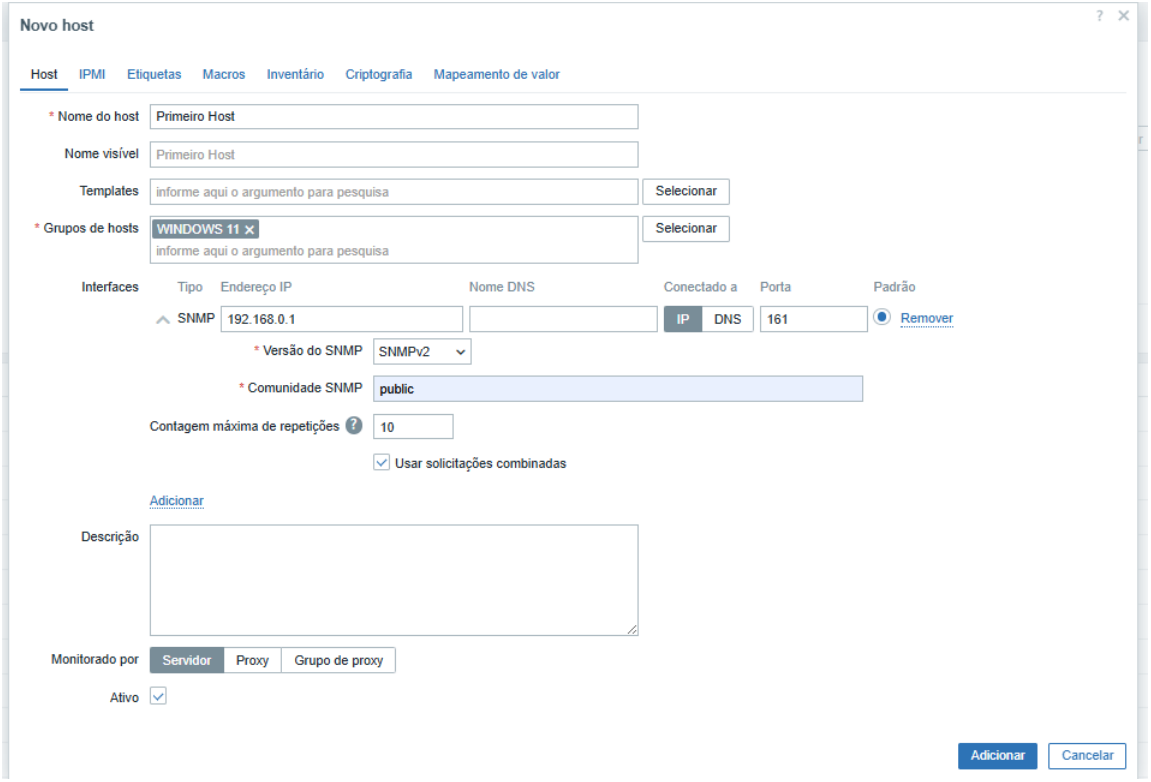
Fonte: ZABBIX (2025)

Ao criar um novo host, é necessário preencher algumas informações como as abaixo, demonstrado na Figura 22:

- Nome do Host: Nome de identificação, que aparecerá nos dashboards e alertas.
- Grupo: Categoria à qual o host pertence (ex.: servidores Linux, dispositivos de rede).
- Endereço IP ou DNS: Endereço da interface que será monitorada.
- Interface de monitoramento: Define o tipo de conexão (Zabbix agent, SNMP, etc.).

A organização adequada dos hosts em grupos facilita a gestão e visualização no ambiente.

Figura 22 – Novo host no Zabbix.



Fonte: ZABBIX (2025)

Dessa forma então, um *host* é criado, porém, ele ainda não possui nenhum item do monitoramento para coletar métricas. A melhor forma de adicionar esses itens será demonstrada na próxima seção.

4.4.2 Aplicação dos *Templates*

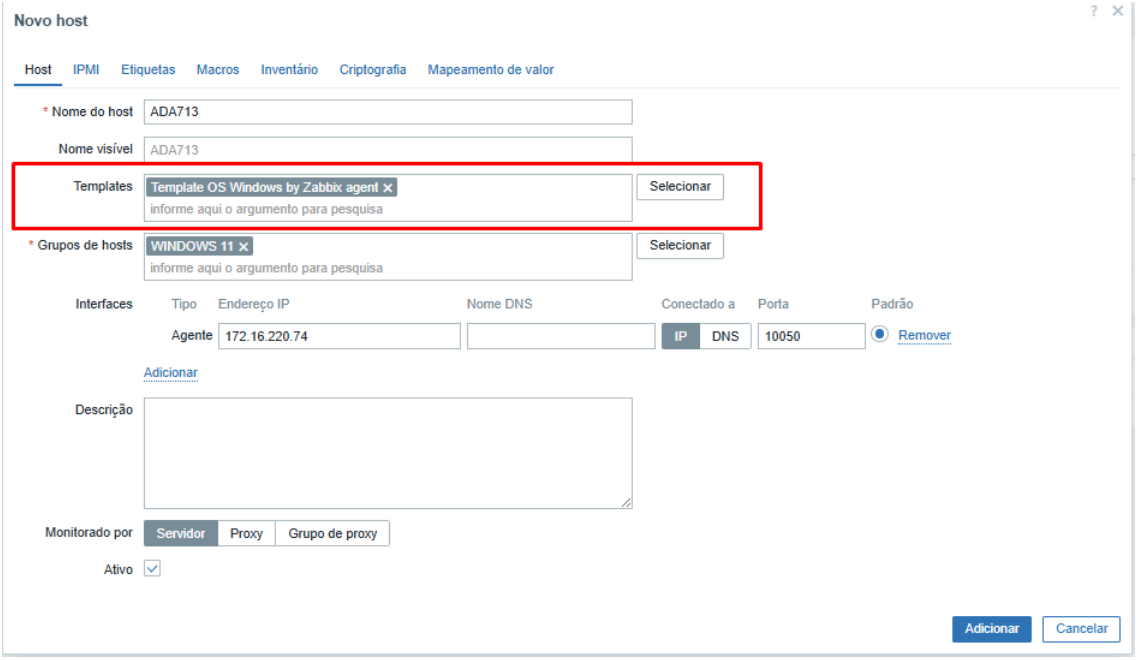
Os templates são modelos pré-configurados que agrupam itens de coleta, triggers, gráficos e regras de descoberta. A aplicação de templates reduz significativamente o tempo de configuração manual de cada host.

Por exemplo, ao adicionar um computador *Windows* ao monitoramento, pode-se utilizar o *template Template OS Windows by Zabbix agent*, que coleta os dados a partir do agente instalado na maquina. Esse *template* já contém os principais parâmetros de monitoramento, como uso de *CPU*, espaço em disco, utilização de memória e outros diversos itens de monitoramento.

O Zabbix disponibiliza diversos templates prontos em sua instalação padrão. Além disso, é possível importar templates adicionais desenvolvidos pela comunidade ou fornecidos pelos próprios fabricantes de dispositivos, o que amplia significativamente as possibilidades de monitoramento.

É possível adicionar um *template* tanto na criação quanto na edição de um host, através do campo *"Template"* conforme Figura a 23.

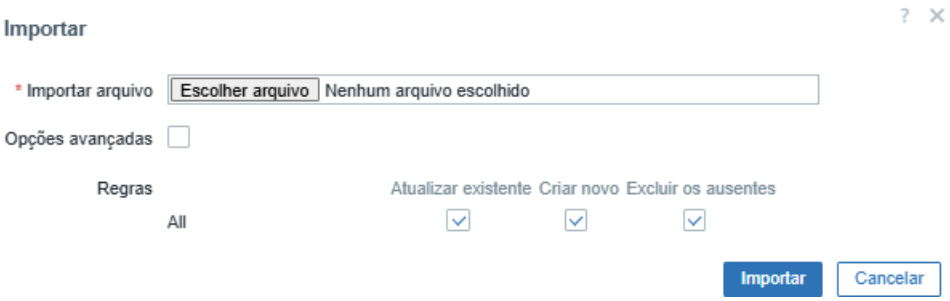
Figura 23 – Adicionando template em um host.



Fonte: ZABBIX (2025)

Para adicionar um *template* externo dentro do *Zabbix*, é acessado a aba "Dados Coletados > Templates", e então "Importar", isso nos dará a possibilidade de importar um *template* para dentro do *Zabbix* conforme Figura 24.

Figura 24 – Tela de importação de template.



Fonte: ZABBIX (2025)

Os *templates* representam uma abordagem eficiente e padronizada para configurar o monitoramento de diferentes tipos de ativos, reduzindo a complexidade e o tempo de implantação. Com a variedade de modelos nativos e a possibilidade de importar *templates* personalizados da comunidade ou de fabricantes, o *Zabbix* oferece grande flexibilidade para atender diferentes cenários. Na proxima seção, será demonstrado o resultado da aplicação do *template* em um *host*.

4.4.3 Itens de Monitoramento e Triggers

Os itens de monitoramento representam as métricas coletadas do *host*. Cada *template* contém dezenas de itens prontos, mas é possível criar itens personalizados conforme a necessidade.

Segue na Figura 25, um exemplo dos itens de monitoramento, criados a partir do *template Template OS Windows by Zabbix agent* em uma maquina Windows.

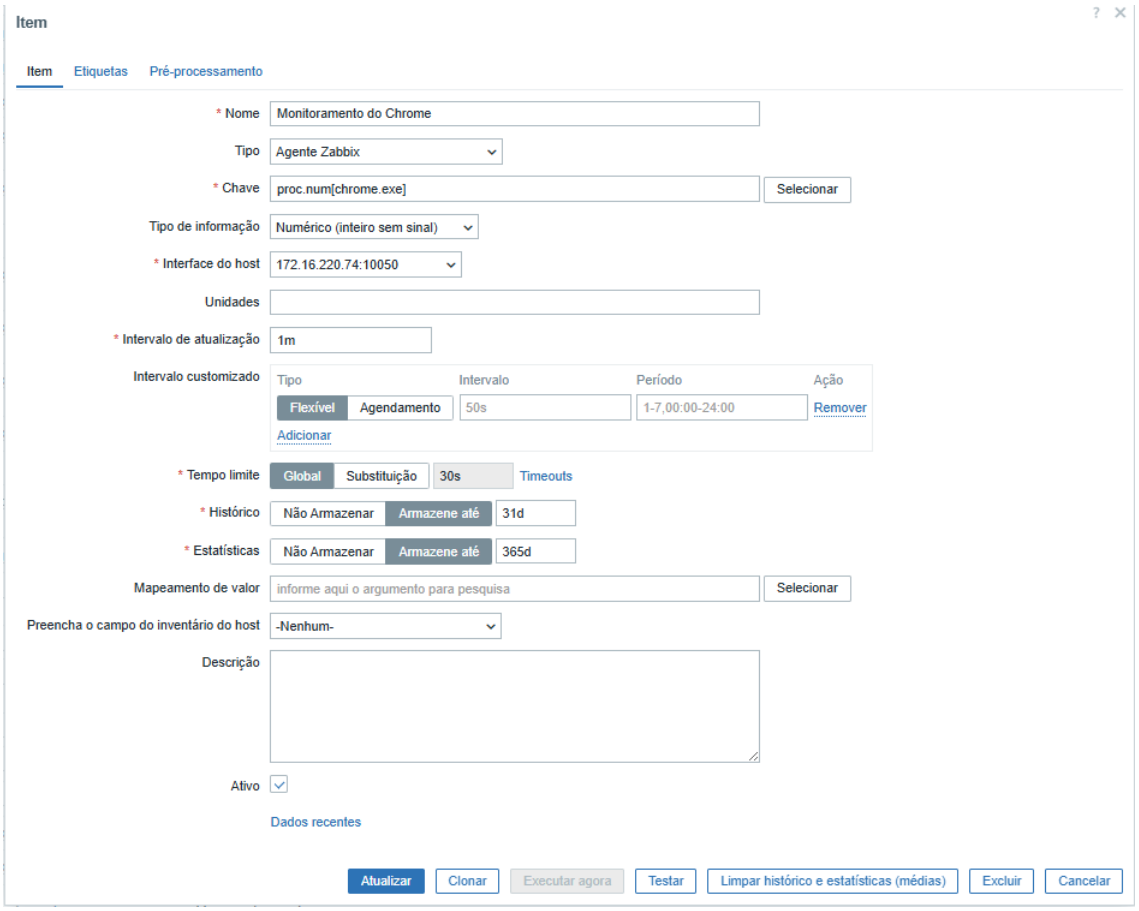
Figura 25 – Itens de monitoramento.

C:: Free space	24s	143.89 GB	+15.57 MB
C:: Space utilization	23s	35.4578 %	-0.006818 %
C:: Total space	22s	222.94 GB	
C:: Used space	21s	79.05 GB	-16.09 MB
Cache bytes	45s	235.88 MB	+292 KB
Context switches per second	55s	1718.7373	-4818.3109
CPU DPC time	59s	0 %	
CPU interrupt time	58s	0 %	
CPU privileged time	57s	0.7777 %	+0.5209 %
CPU queue length	54s	0	
CPU user time	56s	0 %	
CPU utilization	53s	2.1284 %	+1.5528 %
Free memory em %	46s	74.6086 %	-0.8998 %
Free swap space	39s	2.38 GB	
Free swap space in %	40s	100 %	
Free system page table entries	44s	16714584	+29

Fonte: ZABBIX (2025)

Para criamos um item manualmente, precisamos acessar a aba "Dados Coletados > Hosts", selecionar o *host* que você deseja adicionar um item, e utilizar a opção "Criar Item". Na Figura 26, é criou-se um item para monitorar o processo chrome.exe de um computador *Windows*, que verifica se o navegador *Google Chrome* está sendo executado ou não.

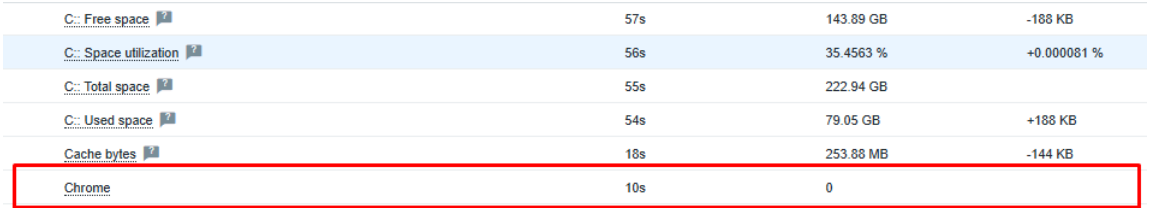
Figura 26 – Criação de item de monitoramento.



Fonte: ZABBIX (2025)

Na Figura 27, é visualizado o item chamado Chrome, monitorando o processo chrome.exe e nos retornado o valor 0, que significa que o processo chrome.exe não está sendo executado.

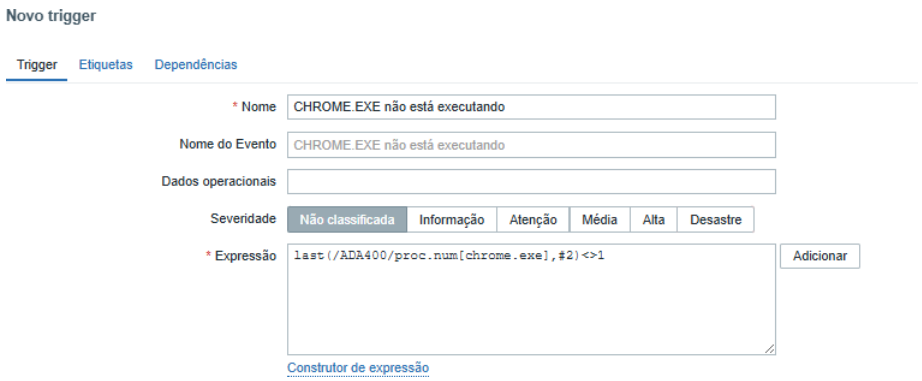
Figura 27 – Monitoramento do processo chrome.exe.



Fonte: ZABBIX (2025)

A partir de um item de monitoramento, é possivel criar *triggers* que são condições que, quando atendidas, disparam alertas. Neste caso, é possivel criar uma *trigger* que dispare um alarme toda vez que ela identifique que o processo chrome.exe não está executando, conforme Figura 28.

Figura 28 – Criação de trigger.



Fonte: ZABBIX (2025)

O alarme aparecerá no *dashboard* do *Zabbix* conforme Figura 29.

Figura 29 – Alarme Zabbix.

Fonte: ZABBIX (2025)

A flexibilidade dos itens e *triggers* permite ao Zabbix realizar desde monitoramentos básicos até verificações específicas com alta precisão, garantindo respostas rápidas e alertas relevantes para cada situação do ambiente.

Com a instalação do Zabbix concluída, os agentes configurados em diferentes sistemas operacionais e a criação de *hosts*, *templates*, itens e *triggers* devidamente realizada, é possível estabelecer um ambiente de monitoramento funcional e adaptável a diversos tipos de infraestrutura. As etapas apresentadas evidenciaram como o Zabbix permite não apenas coletar métricas básicas, mas também realizar verificações específicas e configurar alertas personalizados, proporcionando controle detalhado e reativo sobre os ativos de TI.

Com a infraestrutura básica de monitoramento devidamente configurada no Zabbix, torna-se possível explorar recursos mais avançados que ampliam sua capacidade analítica e operacional. Uma dessas possibilidades é a integração com sistemas de inteligência artificial, capazes de fornecer suporte automatizado à resolução de incidentes. Nesse contexto, a próxima etapa deste trabalho consiste na integração do Zabbix com a *API* Gemini, ferramenta de IA desenvolvida pela Google. Essa integração tem como objetivo fornecer sugestões automáticas de solução com base nas falhas detectadas, agregando inteligência ao processo de monitoramento e contribuindo para a redução do tempo de resposta a incidentes. A seguir, serão apresentados os requisitos, o funcionamento e a implementação dessa integração no ambiente monitorado.

4.5 Integração do Zabbix com Inteligência Artificial

Com o objetivo de ampliar a capacidade de resposta do Zabbix frente a incidentes e tornar o monitoramento mais inteligente, foi desenvolvida uma integração entre o Zabbix e o Gemini, a IA da Google. Essa integração permite que, ao ser gerado um evento por uma *trigger* dentro do Zabbix, uma sugestão automatizada de solução seja obtida por meio da IA Gemini e adicionada automaticamente ao evento como um reconhecimento no próprio Zabbix.

A solução foi implementada utilizando Python 3 no servidor do Zabbix, com chamadas à *API* tanto do Zabbix quanto da Gemini. O processo consiste em capturar a descrição da trigger e o ID do evento, enviando essas informações para a API da Gemini. A resposta retornada pela IA, contendo uma sugestão de solução para o problema descrito, é então enviada de volta ao Zabbix, sendo registrada diretamente no evento correspondente.

Para viabilizar a integração, foram necessárias as seguintes condições:

- Acesso ao servidor Zabbix com permissões de administrador;
- Python 3 instalado;
- API do Zabbix habilitada e *token* gerado;
- Chave de API da Google para acesso ao Gemini;
- Conectividade com a internet para consumo da API externa.

A seguir, será detalhado o funcionamento do *script* Python desenvolvido para realizar essa integração, destacando os principais componentes e seu papel no processo.

4.5.1 Script para integração com o Gemini.

O script inicia importando as bibliotecas essenciais e configurando as chaves de acesso às *APIs* do Gemini e do Zabbix, além do *endpoint* da *API* do Zabbix, conforme Código 4.1.

```
1  import os
2  import sys
3  import requests
4  import json
5  from datetime import datetime
6  import google.generativeai as genai
7
8  GEMINI_API_KEY = 'api'
9  ZABBIX_API_URL = 'http://ip_do_servidor/zabbix/api_jsonrpc.php'
10 ZABBIX_API_KEY = 'api'
```

```
11
12 genai.configure(api_key=GEMINI_API_KEY)
13 model = genai.GenerativeModel('gemini-1.5-flash')
14
```

Código 4.1 – Descrição das APIs

Ao receber a descrição do problema proveniente da *trigger* do Zabbix, o script executa a função `get_gemini_suggestion()`, que prepara um *prompt* detalhado, projetado para extrair uma resposta objetiva e útil da inteligência artificial. O *prompt* segue uma estrutura fixa, contendo a descrição do gatilho e uma orientação contextual para que o modelo atue como um especialista de TI em um ambiente corporativo. Essa abordagem visa gerar respostas curtas, práticas e adaptadas ao cenário de incidentes reais.

> O seguinte gatilho de falha foi ativado: '{*problem_description*}'. Você, como um Especialista de TI dentro de um ambiente corporativo, por favor, forneça uma solução concisa e resumida para este problema.

Em seguida, é realizada a chamada à API do Gemini utilizando o método `generate_content()`, com uma configuração personalizada definida por meio do objeto `generation_config`. Os principais parâmetros utilizados são:

- **candidate_count = 1**: solicita apenas uma sugestão de resposta, evitando múltiplas variações e agilizando o processamento.
- **temperature = 0.2**: define um nível baixo de aleatoriedade na resposta, favorecendo consistência e objetividade.
- **top_p = 0.9**: ativa o uso de *nucleus sampling*, restringindo a seleção de palavras às mais prováveis, o que contribui para respostas mais relevantes.
- **top_k = 40**: limita o número de tokens considerados a cada passo de geração, aumentando a precisão.
- **max_output_tokens = 400**: define o limite máximo de tokens na resposta, garantindo que as sugestões sejam diretas e não excessivamente longas.

Caso a descrição da *trigger* esteja ausente ou incompleta, o script interrompe a execução e retorna uma mensagem padrão de alerta, evitando chamadas desnecessárias à API. Todas as etapas são registradas em log, incluindo tanto a descrição recebida quanto a sugestão final gerada pela IA. O retorno da função é a resposta textual do Gemini, que é inserida automaticamente como reconhecimento do evento correspondente no Zabbix.

```
15 def get_gemini_suggestion(problem_description):
16     if not problem_description or len(problem_description.strip()) == 0:
17         log_message("Descrição do gatilho está vazia ou incompleta.")
18         return "A descrição do gatilho de falha está vazia ou incompleta."
19 
20     log_message(f"Descrição recebida: {problem_description}")
21 
22     detailed_prompt = (
23         f"O seguinte gatilho de falha foi ativado: '{problem_description}'. "
24         "Você, como um Especialista de TI dentro de um ambiente corporativo, Por
       favor, forneça uma solução concisa e resumida para este problema."
25     )
26     try:
27         response = model.generate_content(
28             detailed_prompt,
29             generation_config=genai.types.GenerationConfig(
30                 candidate_count=1,
31                 temperature=0.2,
32                 top_p=0.9,
33                 max_output_tokens=400,
34                 top_k=40
35             )
36         )
37         suggestion = response.text
38         log_message(f"Sugestão do Gemini: {suggestion}")
39 
40         return suggestion
41 
42     except Exception as e:
43         log_message(f"Erro na API do Gemini: {e}")
44         return f"Erro na API do Gemini: {e}"
45 
```

Com a sugestão gerada, o *script* utiliza a *API* do Zabbix para adicionar um reconhecimento ao evento original, registrando a recomendação da IA diretamente no incidente, facilitando a análise e resolução pelos operadores.

```
46 def acknowledge_event(event_id, message):
47     headers = {
48         'Content-Type': 'application/json-rpc',
49     }
50     payload = {
51         "jsonrpc": "2.0",
52         "method": "event.acknowledge",
53         "params": {
54             "eventids": [event_id],
55             "message": message,
56             "action": 6
57         },
58         "auth": ZABBIX_API_KEY,
59         "id": 1
60     }
61     try:
62         response = requests.post(ZABBIX_API_URL, json=payload, headers=headers)
63         response_json = response.json()
64 
65         if 'error' in response_json:
66             log_message(f"Erro na API do Zabbix: {response_json['error']['data']}")
67             return f"Erro na API do Zabbix: {response_json['error']['data']}"
68         else:
69             log_message("Reconhecimento adicionado com sucesso.")
70             return "Reconhecimento adicionado com sucesso."
71 
72     except Exception as e:
73         log_message(f"Erro ao comunicar-se com a API do Zabbix: {e}")
74         return f"Erro ao comunicar-se com a API do Zabbix: {e}"
75 
```

Para acompanhamento e auditoria, o *script* possui uma função que grava *logs* com data e hora, registrando eventos, mensagens de erro e resultados das chamadas às *APIs*.

```
76 def log_message(message):
77     with open("/tmp/zabbix_gemini.log", "a") as log_file:
78         log_file.write(f"{datetime.now()}: {message}\n")
79 
```

Por fim, o *script* é acionado via linha de comando, recebendo como parâmetros a descrição da *trigger* e o ID do evento. O fluxo consiste em gerar a sugestão do Gemini e registrar o reconhecimento no Zabbix, com todos os passos sendo registrados em *log* para garantir a rastreabilidade.

```
80 if len(sys.argv) > 2:
81     trigger_description = sys.argv[1]
82     event_id = sys.argv[2]
83 
84     log_message(f"Executando script com descrição: {trigger_description} e ID do
       evento: {event_id}")
85 
86     suggestion = get_gemini_suggestion(trigger_description)
87     result = acknowledge_event(event_id, suggestion)
88 
89     log_message(f"Resultado final: {result}")
90     print(result)
91 else:
92     log_message("Erro: Descrição da trigger e ID do evento não fornecidos.")
93     print("Erro: Descrição da trigger e ID do evento não fornecidos.")
94 
```

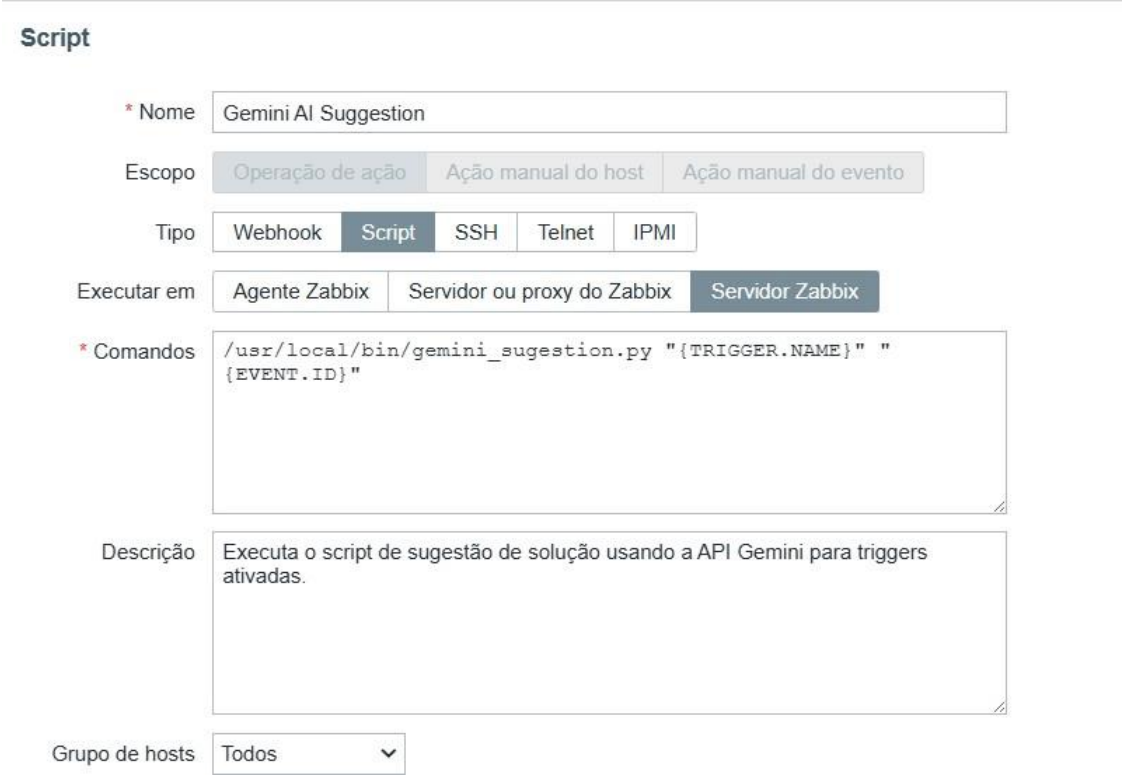
Agora o próximo passo é realizar as configurações dentro do Zabbix para a execução automática do *script* nos incidentes.

4.5.2 Configuração do Script no Zabbix.

Com o *script* Python implementado, a próxima etapa consiste em configurar o Zabbix para executar automaticamente esse script nos eventos de falha. Esse processo é realizado através do módulo de automações do Zabbix, utilizando as funcionalidades de *Scripts* e Ações.

Inicialmente, é necessário registrar o *script* na interface web do Zabbix. Para isso, acessa-se o menu Alertas > *Scripts* e cria-se um novo *script* com um nome descritivo, criaremos um *script* chamado "*Gemini AI Suggestion*". O tipo de execução deve ser definido como *Script*, e o local de execução como Servidor Zabbix, pois o *script* será executado diretamente no servidor onde está instalado o agente Python. No campo de comando do *script*, deve-se inserir a seguinte instrução, conforme Figura 30.

Figura 30 – Configuração do *Script* no Zabbix.

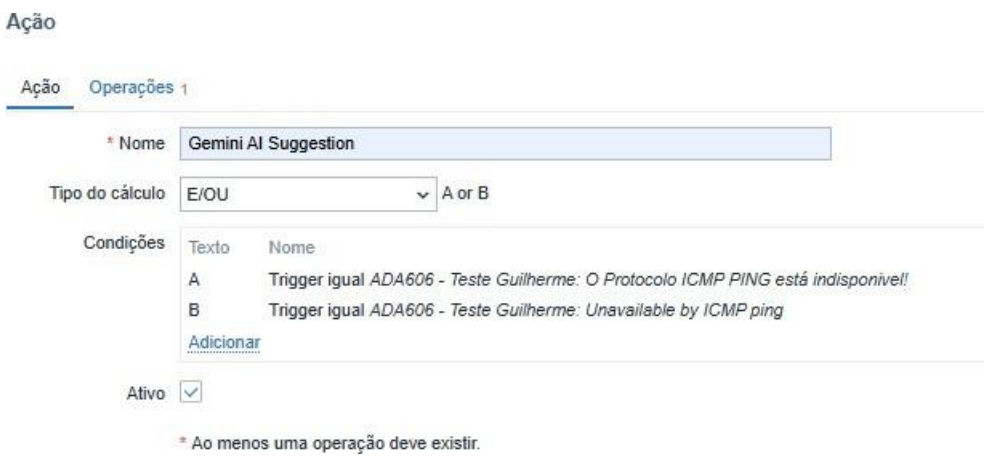


Fonte: ZABBIX (2025)

Essa chamada garante que, ao ser acionado, o *script* receba como argumentos a descrição do gatilho e o ID do evento que o originou. Essas informações são necessárias para que o *script* envie o contexto da falha para a *API* Gemini e registre a resposta no evento correspondente.

Após cadastrar o *script*, é necessário configurar uma ação que o execute automaticamente diante de eventos específicos. Essa configuração é feita em Alertas > Ações > Ações de *trigger*. Nessa seção, cria-se uma nova ação, definindo um nome e as condições de disparo. Para fins de teste e validação, foi utilizada uma trigger de indisponibilidade de rede (ICMP PING), mas a lógica pode ser aplicada a qualquer tipo de falha monitorada pelo Zabbix, a Figura 31 demonstra a ação configurada.

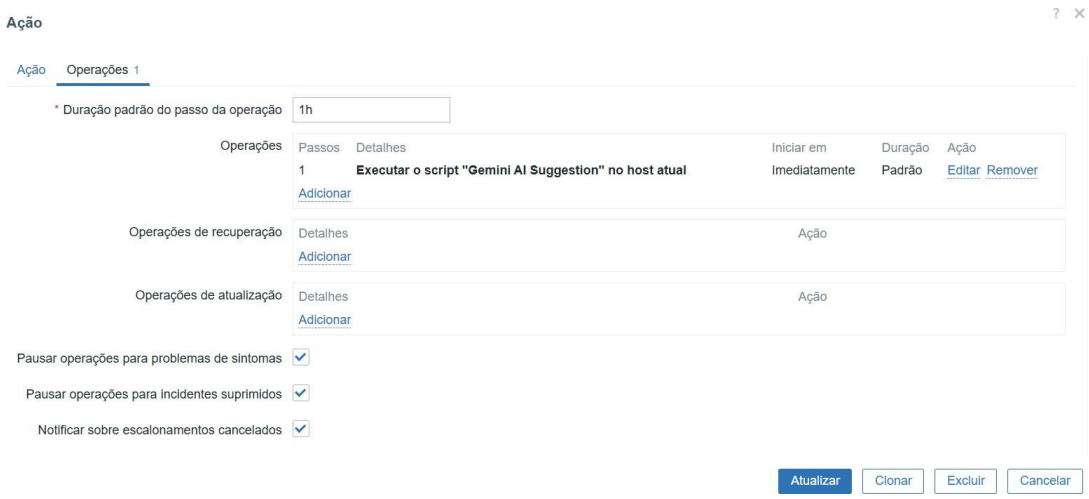
Figura 31 – Ação de Trigger.



Fonte: ZABBIX (2025)

Na aba operações, conforme Figura 32, adiciona-se a operação desejada e, em “tipo de operação”, seleciona-se "executar *script* personalizado". Em “*script* ”, deve-se escolher o *script* previamente cadastrado. É importante definir também que o destino da execução será o *host* atual, garantindo que o contexto da falha seja corretamente atribuído ao ativo afetado.

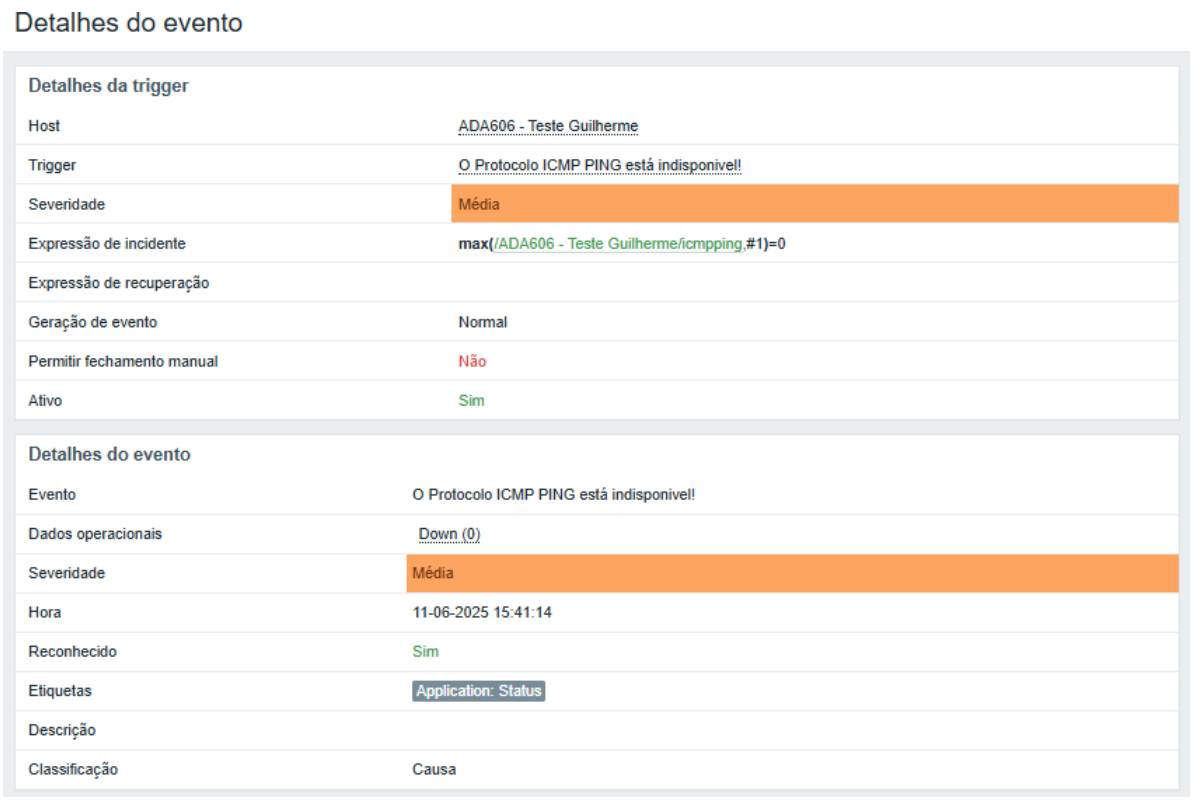
Figura 32 – Operação da Ação.



Fonte: ZABBIX (2025)

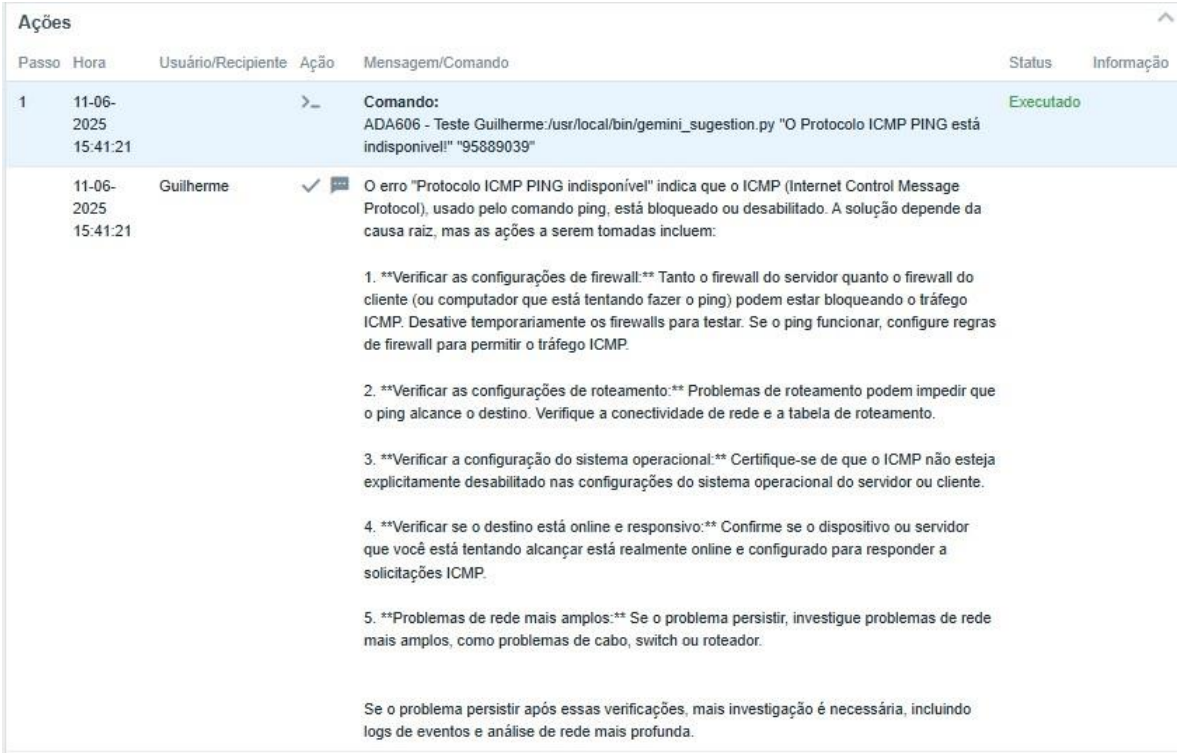
Com essa configuração, sempre que a *trigger* definida for ativada, o Zabbix executará o *script* automaticamente, enviará a descrição da falha à IA do Gemini e registrará a sugestão recebida como um reconhecimento no próprio evento, o resultado de todo esse processo se demonstra na Figura 33 e na Figura 34.

Figura 33 – Incidente de falha.



Fonte: ZABBIX (2025)

Figura 34 – Ação do Gemini nos Incidentes.



Fonte: ZABBIX (2025)

4.5.3 Considerações Finais sobre a Integração com Inteligência Artificial

A integração entre o Zabbix e a inteligência artificial representa um avanço significativo na forma como ambientes de TI são monitorados. Ao automatizar o processo de sugestão de soluções para eventos de falha, a IA atua como um agente de apoio à tomada de decisões, proporcionando respostas mais rápidas, precisas e contextualizadas. Essa abordagem amplia a capacidade analítica da ferramenta de monitoramento, transformando dados técnicos em recomendações práticas que contribuem diretamente para a agilidade e a eficiência na resolução de incidentes.

Além disso, a automatização do reconhecimento de eventos com sugestões inteligentes reduz a sobrecarga da equipe de TI, permitindo que os profissionais concentrem seus esforços em ações estratégicas. A integração com o Gemini demonstra como a inteligência artificial pode ser aplicada de forma eficaz no contexto corporativo, agregando valor ao monitoramento tradicional e elevando o nível de maturidade operacional da infraestrutura de tecnologia.

Com essa etapa concluída, evidencia-se a importância de explorar ferramentas que complementem o processo de monitoramento com recursos visuais avançados. Nesse sentido, a próxima seção deste trabalho será dedicada à instalação e configuração do Grafana, ferramenta de observabilidade que, ao ser integrada ao Zabbix, permitirá transformar os dados coletados em dashboards interativos, promovendo uma análise visual mais intuitiva e eficiente do ambiente monitorado.

4.6 Instalação e Configuração do *Grafana*

Após a instalação do Zabbix e a definição dos *hosts*, itens e *triggers*, a próxima etapa consiste na implantação do Grafana, ferramenta responsável por transformar os dados coletados em dashboards visuais e interativos. Esta seção abordará todo o processo de instalação do Grafana no sistema Debian 12, bem como a configuração da fonte de dados do Zabbix e a criação de um painel de monitoramento básico.

4.6.1 Instalação do *Grafana* em Ambiente *Debian*

Para realizar a instalação do *Grafana*, seguimos as instruções da documentação oficial.

O primeiro passo, é instalar os pacotes de pré requisitos conforme Figura 35.

Figura 35 – Pacotes de pré requisitos Grafana.

```
root@MON14:~# sudo apt-get install -y apt-transport-ht
tps software-properties-common wget
```

Fonte: AUTOR (2025)

Em seguida é importado a chave GPG conforme Figura 36, a chave GPG permite que o sistema verifique a autenticidade dos pacotes baixados no repositório do *Grafana*, garantindo que tudo é seguro.

Figura 36 – Importação da chave GPG.

```
root@MON14:~# sudo mkdir -p /etc/apt/keyrings/
wget -q -O - https://apt.grafana.com/gpg.key | gpg
 --dearmor | sudo tee /etc/apt/keyrings/grafana.gp
g > /dev/null
```

Fonte: AUTOR (2025)

Agora, é adicionado o repositório das versões estáveis, conforme Figura 37.

Figura 37 – Repositório das versões estáveis.

```
root@MON14:~# echo "deb [signed-by=/etc/apt/keyrin
gs/grafana.gpg] https://apt.grafana.com stable mai
n" | sudo tee -a /etc/apt/sources.list.d/grafana.l
ist
```

Fonte: AUTOR (2025)

Para finalizar, é realizado a instalação da aplicação *Grafana*, conforme Figura 38.

Figura 38 – Comando de instalação do *Grafana*.

```
root@MON14:~# sudo apt-get install grafana
```

Fonte: AUTOR (2025)

Agora é possível acessar a interface web do *Grafana* por meio do endereço IP do servidor, utilizando a porta padrão 3000. O acesso deve ser feito pelo navegador, no formato: http://<IPDOSERVIDOR>:3000. Na Figura 39, é apresentada a tela de login do *Grafana*. As credenciais padrão para o primeiro acesso são: usuário admin e senha admin.

Figura 39 – Tela de login do *Grafana*

Fonte: GRAFANA (2025)

Na próxima seção, será mostrado como conectar o *Grafana* com o *Zabbix*.

4.6.2 Integração do *Grafana* com *Zabbix*

Para possibilitar a visualização dos dados do *Zabbix* no *Grafana*, é necessário integrar ambas as ferramentas através de um *plugin*. Esse *plugin* permite ao *Grafana* acessar os dados coletados pelo Zabbix,

A instalação do plugin é feita utilizando a ferramenta de linha de comando grafana-cli, dentro do servidor. Execute o seguinte comando demonstrado na Figura 40.

Figura 40 – Instalando o *plugin Zabbix*.

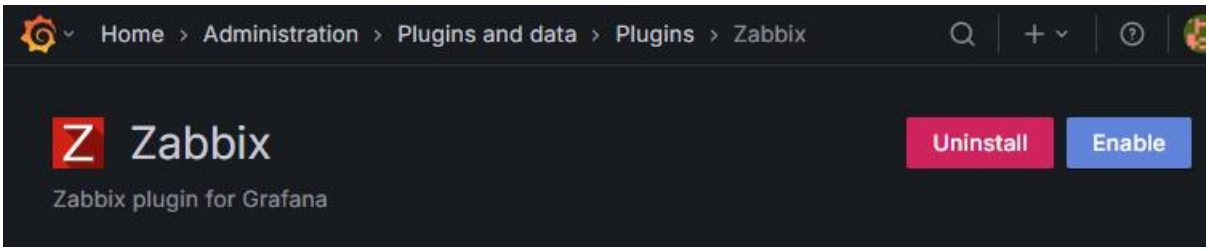
```
root@MON14:~# sudo grafana-cli plugins install alexanderzobnin
-zabbix-app
```

Fonte: AUTOR (2025)

Para o funcionamento do *plugin*, é necessário reiniciar o serviço do *Grafana* no servidor.

O próximo passo é habilitar o *plugin* dentro das configurações do *Grafana*, conforme Figura 41.

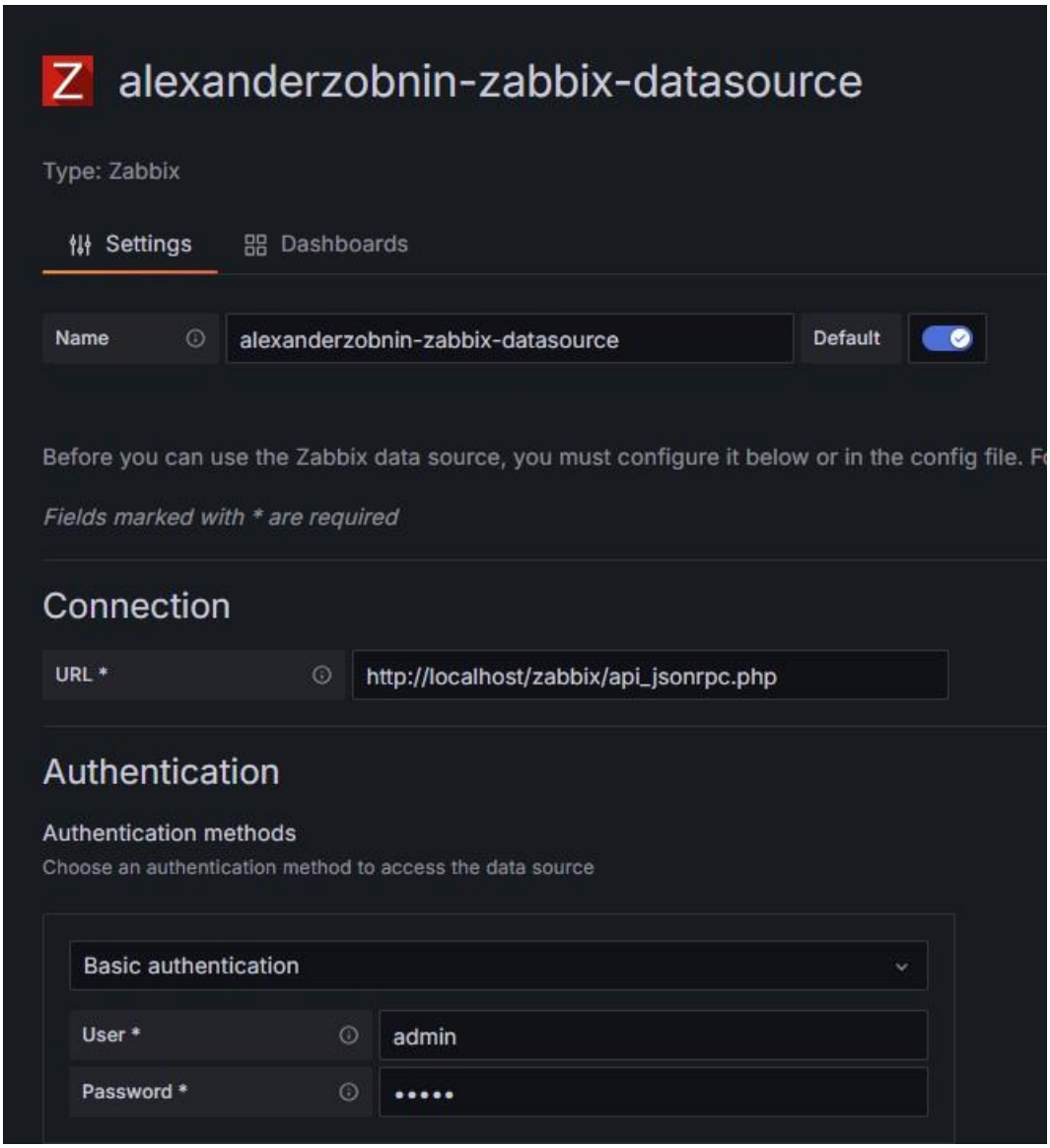
Figura 41 – Habilitando o plugin do *Zabbix*.



Fonte: GRAFANA (2025)

No seguinte passo, vamos adicionar a fonte de dados do *Zabbix* no *Grafana*. É necessário preencher o campo de URL indicando o caminho *Zabbix* e os campos de credenciais para conexão, as credencias devem possuir os privilégios de leitura no *Zabbix*, exemplo na Figura 42.

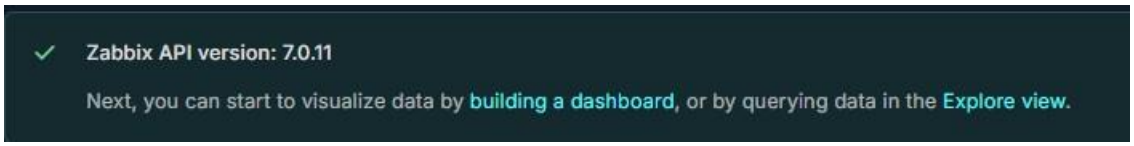
Figura 42 – Adicionando a fonte de dados do Zabbix.



Fonte: GRAFANA (2025)

Se a configuração estiver correta, uma mensagem de sucesso será exibida, conforme Figura 43, indicando que o *Grafana* está apto a consultar os dados do *Zabbix*.

Figura 43 – Sucesso na adição da fonte de dados.



Fonte: GRAFANA (2025)

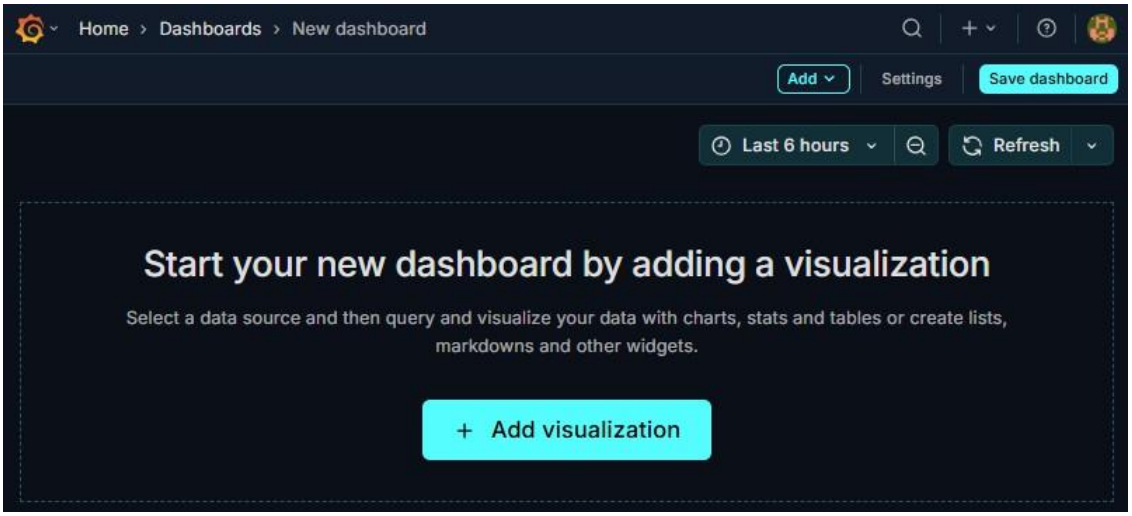
Com a configuração completa, a próxima etapa é a criação dos *dashboards* dentro do *Grafana*.

4.6.3 Criação de *Dashboard* no *Grafana*

Com o *Grafana* devidamente instalado e integrado ao *Zabbix*, é possível criar *dashboards* personalizados que permitem uma visualização clara e objetiva dos ativos monitorados. Nesta seção, será exemplificada a criação de um painel simples e funcional voltado para o acompanhamento de um ponto de acesso *Wi-Fi*, destacando duas métricas principais: o tempo ligado e o *ping*.

Para a construção desse painel, é necessário acessar o ambiente web do *Grafana* e criou-se um novo *dashboard* por meio da opção “*New Dashboard”*, conforme Figura 44.

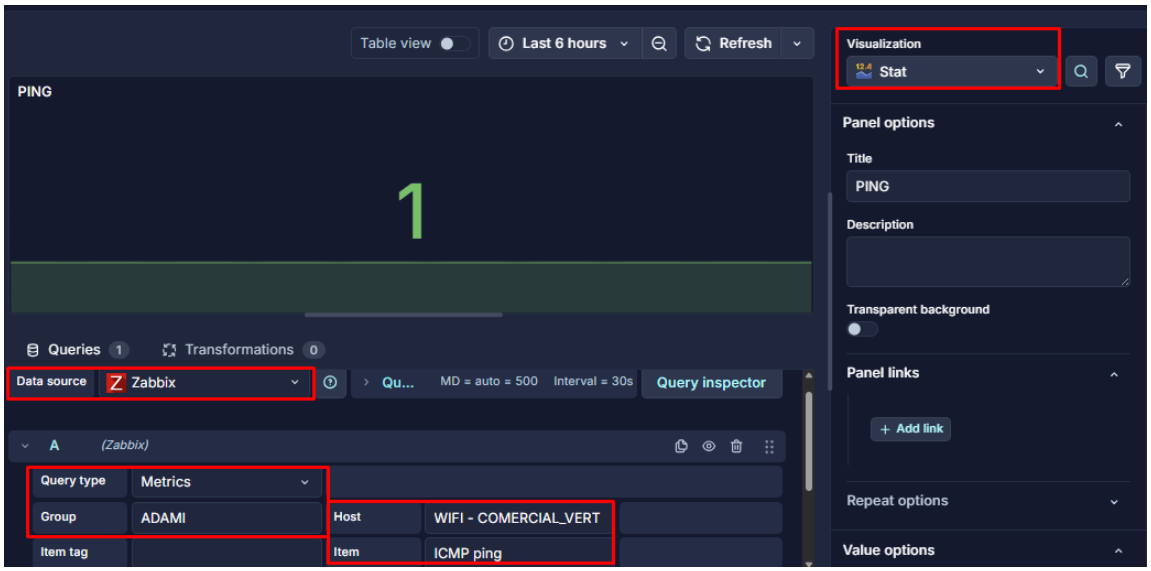
Figura 44 – Criação de *Dashboard*.



Fonte: GRAFANA (2025)

Dentro do *dashboard*, foi é adicionado um novo painel do tipo "*stat* ", configurando como fonte de dados o Zabbix previamente integrado. Na aba de consultas (*Query*), é selecionado o *host* correspondente ao ponto de acesso, e, em seguida, adicionadas as métricas desejadas. A primeira métrica utilizada foi a de *ping*, representando a disponibilidade do dispositivo por meio da checagem *ICMP*, conforme Figura 45.

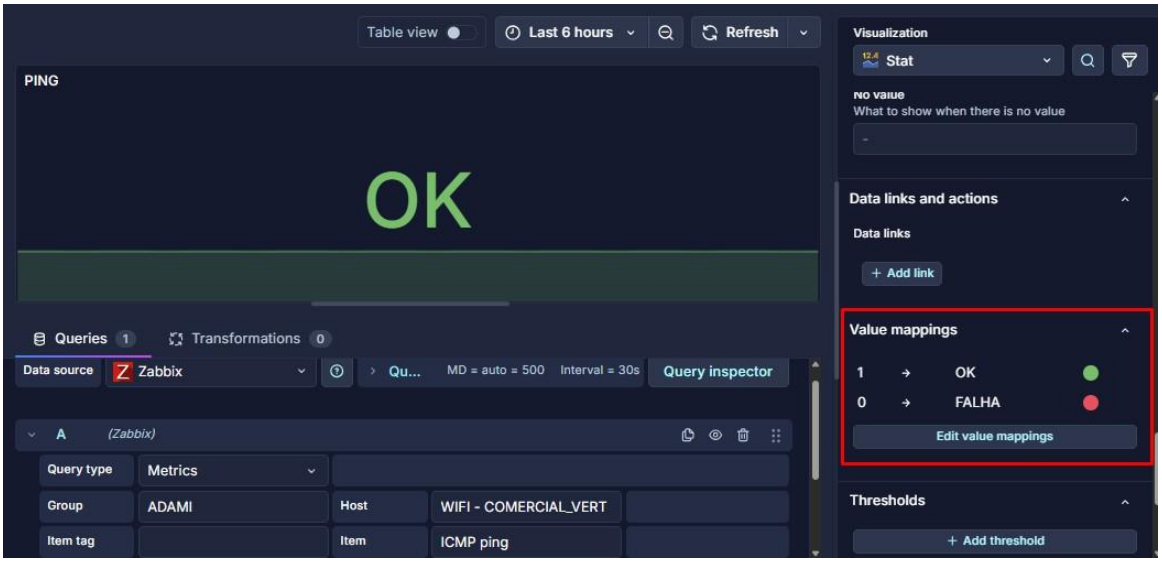
Figura 45 – Criação do painel de ping.



Fonte: GRAFANA (2025)

Como a métrica de ping retorna valores binários, sendo 0 para falha e 1 para funcionamento normal, é necessário realizar um ajuste na configuração do painel para garantir uma representação adequada. Essa tratativa é feita na opção *"value mappings"*, disponível nas configurações do painel do tipo "*stat* ". Nessa seção, define-se que o valor 0 seja exibido como "FALHA", e o valor 1 como "*OK* ", permitindo que os dados sejam interpretados de forma clara e intuitiva, tanto visual quanto funcionalmente, conforme Figura 46

Figura 46 – Mapeamento de valor.

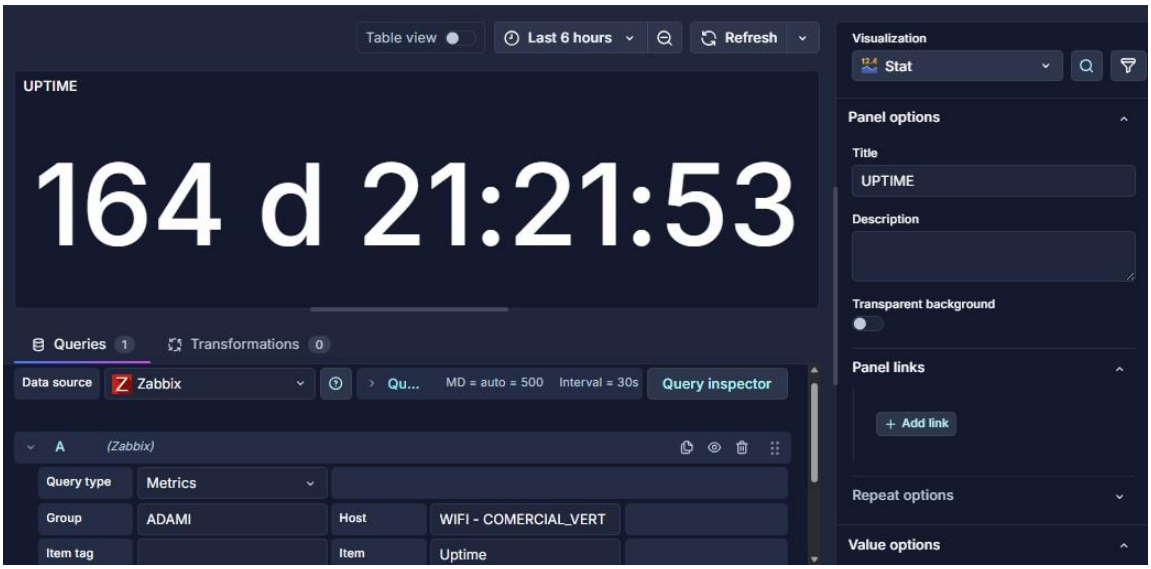


Fonte: GRAFANA (2025)

A segunda métrica, conforme Figura 47, foi o *uptime*, que indica o tempo contínuo de atividade do ponto de acesso, útil para verificar se houve reinicializações

ou quedas no serviço, ele foi configurado seguindo os mesmos princípios do painel de *ping*.

Figura 47 – Painel de uptime.



Fonte: GRAFANA (2025)

Também foi criado um painel com o propósito de ilustrar visualmente o objeto monitorado, neste caso, um ponto de acesso Wi-Fi. Para isso, utilizou-se um painel do tipo "text", no qual foi inserido um código em *HTML* contendo a referência a uma imagem representativa, acompanhada do nome do ponto de acesso.

A Figura 48 ilustra o resultado final do painel, com três seções principais: o nome do setor monitorado, o tempo de atividade do ponto de acesso, e o status da conectividade via ping. Esse tipo de *dashboard* é extremamente útil em ambientes operacionais, pois permite o acompanhamento em tempo real do funcionamento de dispositivos críticos, contribuindo para a identificação rápida de falhas e para a tomada de decisões assertivas pela equipe técnica.

Figura 48 – Monitoramento de ponto de acesso.

Fonte: GRAFANA (2025)

A criação deste *dashboard* demonstra a flexibilidade e a capacidade do Grafana em representar graficamente os dados coletados pelo Zabbix de forma clara e acessível. Mesmo com um painel simples, é possível obter informações relevantes e em tempo real sobre o estado de dispositivos críticos na infraestrutura de rede. Ao combinar elementos visuais, como ícones, indicadores de status e métricas numéricas, o painel fornece uma visão consolidada que facilita a tomada de decisões rápidas por parte da equipe de TI.

Além disso, a utilização de painéis visuais melhora significativamente a experiência do usuário e a eficiência operacional, especialmente em ambientes que exigem monitoramento contínuo. A construção desse tipo de *dashboard* evidencia como o Grafana, quando integrado ao Zabbix, não apenas complementa, mas potencializa as capacidades de observabilidade dos ativos de TI.

Com a implementação do Grafana finalizada e a criação de um *dashboard* funcional estabelecido, conclui-se a etapa prática de visualização. A próxima fase do trabalho será dedicada à análise dos resultados esperados, destacando os benefí- cios operacionais e estratégicos alcançados com a integração entre as ferramentas utilizadas.

5 RESULTADOS

A implementação prática apresentada ao longo deste trabalho demonstrou o potencial significativo das ferramentas Zabbix, Grafana e da inteligência artificial integrada para o monitoramento e a observabilidade de ativos de TI. A escolha de ferramentas *open source* e amplamente consolidadas no mercado permitiu montar uma arquitetura de baixo custo, porém altamente funcional, escalável e flexível para diferentes tipos de infraestrutura.

Antes da adoção dessa solução integrada, o ambiente de monitoramento da organização era composto por ferramentas isoladas e com funcionalidades limitadas, como o *The Dude*, o *Unifi Controller*, além do acesso manual e local a dispositivos de rede, como *switches* e *access points*. Essa abordagem descentralizada exigia que os profissionais de TI alternassem constantemente entre diferentes sistemas e interfaces, o que resultava em perda de tempo, aumento da complexidade operacional e ausência de uma visão unificada da infraestrutura. A falta de padronização também dificultava a correlação entre eventos, a análise de tendências e a resposta eficiente a falhas, comprometendo a agilidade na tomada de decisões e a continuidade dos serviços.

A partir da configuração do Zabbix, foi possível realizar o monitoramento detalhado de ativos físicos e virtuais, tanto em sistemas Windows quanto Linux, com coleta de métricas em tempo real de diversas métricas. A utilização de *templates*, itens personalizados e *triggers* possibilitou uma cobertura abrangente e ajustável às necessidades do ambiente.

Com a integração ao Grafana, esses dados foram transformados em *dash-boards* interativos, capazes de transmitir uma visão clara e acessível do estado da infraestrutura monitorada. A visualização das informações em tempo real contribuiu para facilitar a análise de tendências, identificação de gargalos e acompanhamento da disponibilidade dos ativos. Essa interface visual demonstrou ser uma aliada importante para equipes técnicas e tomadores de decisão, promovendo agilidade na resposta a incidentes.

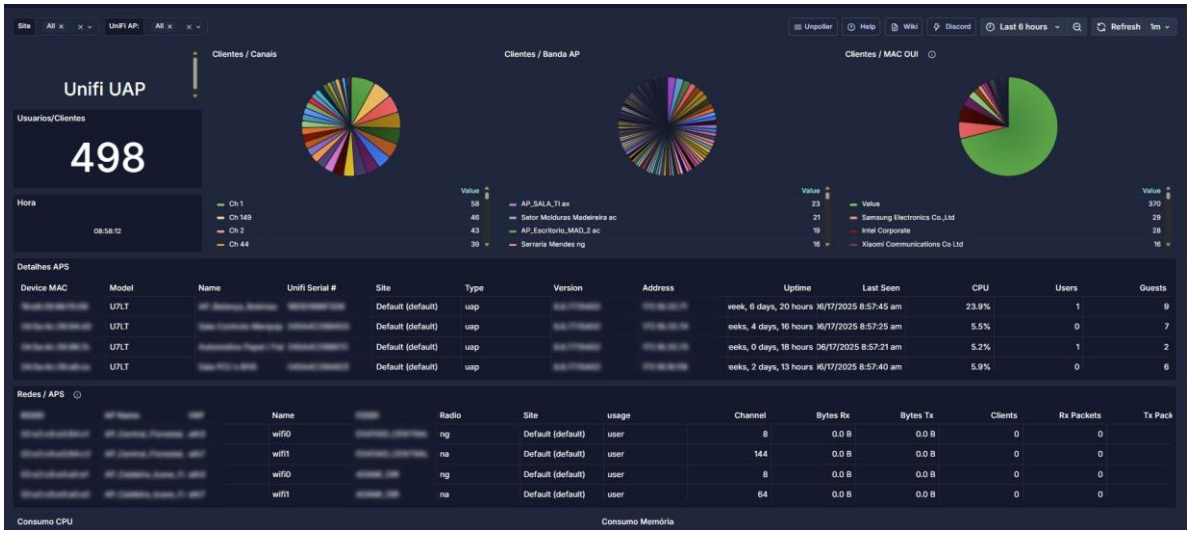
A Figura 49 exemplifica um dos *dashboards* construídos para monitoramento da rede Wi-Fi, apresentando indicadores como a disponibilidade, tempo ligado e a identificação do dispositivo.

Figura 49 – Dashboard Unifi Simplificado.

Fonte: GRAFANA (2025)

A Figura 50 também exemplifica um dos *dashboards* construídos para monitoramento da rede Wi-Fi, porém mais complexo, apresentando diversas métricas como tipo do dispositivo, versão, MAC, nome, número de série, endereço IP, uso de CPU, quantidade de usuários entre diversos outras métricas.

Figura 50 – Dashboard Unifi.



Fonte: GRAFANA (2025)

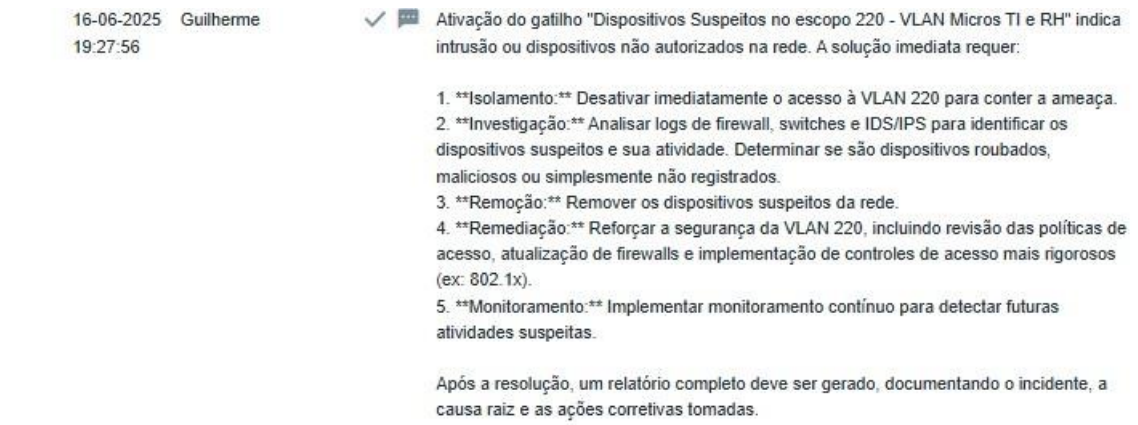
Esses *dashboards* também foram incluídos em TVs de monitoramento dentro do departamento de TI. A Figura 51 ilustra duas telas onde os *dashboards* mostram em tempo real o status de conectividade, disponibilidade de equipamentos e falhas em aberto.

Figura 51 – TV com monitoramento ativo.

Fonte: AUTOR (2025)

Por fim, o diferencial mais expressivo foi a aplicação da inteligência artificial, por meio da integração com o Gemini. A execução automática de *scripts*, que geram sugestões de solução com base na descrição das falhas detectadas, agrega uma camada de inteligência proativa ao processo de monitoramento. A Figura 52 apresenta um exemplo de evento registrado no Zabbix, com uma sugestão automatizada adicionada como reconhecimento.

Figura 52 – Ação da inteligência artificial em um incidente de falha dentro do Zabbix.



Fonte: AUTOR (2025)

A arquitetura proposta não apenas se mostrou viável em ambiente de testes, como também foi aplicada com sucesso em um ambiente de produção real. Mesmo sem uma análise quantitativa formal, foi possível identificar melhorias substanciais na rotina da equipe técnica. Antes da implementação, a detecção de incidentes dependia, em muitos casos, da abertura de chamados por parte dos usuários, o que resultava em atrasos consideráveis na resposta. Com a nova estrutura baseada em Zabbix, Grafana e inteligência artificial, diversos tipos de falhas, como quedas de conectividade,

sobrecarga de CPU ou indisponibilidade de serviços, passaram a ser identificadas automaticamente, muitas vezes antes mesmo de qualquer percepção por parte do usuário final. A equipe passou a atuar com base em alertas e dashboards em tempo real, o que facilitou o diagnóstico, reduziu o tempo médio de resposta e aumentou a eficácia das ações corretivas. A automação de sugestões por meio da IA também contribuiu para padronizar procedimentos, diminuir a carga cognitiva dos analistas e promover um ambiente mais confiável, estável e proativo.

Em conjunto, esses elementos demonstram que o monitoramento, quando bem estruturado e integrado a recursos de visualização e inteligência, é um pilar essencial para garantir disponibilidade, continuidade e desempenho em ambientes corporativos modernos.

6 Considerações Finais

Este trabalho teve como objetivo principal demonstrar como o monitoramento e a observabilidade de ativos de TI podem ser significativamente potencializados através da combinação de ferramentas *open source* e inteligência artificial. A partir da aplicação prática das soluções Zabbix, Grafana e Gemini, foi possível construir um ambiente robusto, escalável e inteligente, capaz de coletar dados em tempo real, exibi-los de maneira visual, acessível e ainda sugerir respostas automatizadas para incidentes.

A escolha do Zabbix como plataforma principal de monitoramento se mostrou acertada, tendo em vista sua flexibilidade, vasto suporte a diversos protocolos e interfaces, além da possibilidade de personalização com *templates*, itens e *triggers*. Por sua vez, o Grafana complementou a solução ao oferecer visualizações ricas, *dashboards* dinâmicos e uma interface intuitiva que facilitou a análise de métricas por parte das equipes técnicas. A integração entre ambas as ferramentas resultou em uma plataforma completa de observabilidade, capaz de fornecer uma visão ampla e profunda sobre o ambiente de TI monitorado.

A inovação mais relevante deste trabalho foi a implementação da inteligência artificial, utilizando o Gemini, da Google. Essa integração permitiu automatizar a análise de falhas, oferecendo recomendações contextualizadas e relevantes diretamente no painel do Zabbix. Essa funcionalidade demonstrou não apenas ganhos em agilidade na resposta a incidentes, mas também na padronização e qualificação do atendimento técnico, agregando um novo patamar de inteligência ao processo de gestão de infraestrutura.

Importante destacar que o ambiente desenvolvido e descrito neste trabalho foi efetivamente aplicado em um cenário de produção real, o que conferiu ao projeto um grau elevado de aplicabilidade e relevância prática. A adoção do sistema possibilitou uma redução significativa no tempo médio de resposta a incidentes, bem como um maior controle e visibilidade sobre os ativos de TI. Além disso, os painéis de visualização exibidos em monitores e TVs de operação contribuíram para uma atuação mais proativa das equipes, com alertas mais rápidos e decisões mais assertivas.

Conclui-se, portanto, que a proposta apresentada é viável, eficaz e recomendável para empresas que desejam adotar uma estratégia moderna de gestão de infraestrutura. A combinação entre monitoramento, visualização e inteligência artificial não apenas amplia a capacidade técnica das ferramentas envolvidas, mas transforma o processo de monitoramento em uma ação estratégica para a continuidade e a eficiência dos negócios.

Dessa forma, conclui-se que a integração entre o Zabbix, o Grafana e a inteligência artificial representa uma solução eficaz, moderna e aplicável ao contexto real

das organizações. A arquitetura implementada demonstrou capacidade de promover maior visibilidade operacional, agilidade na resposta a incidentes e apoio técnico inteligente por meio da automação. Os resultados obtidos com a implantação do sistema em ambiente de produção validam a importância de unir ferramentas consolidadas de monitoramento com recursos de visualização avançada e inteligência computacional. O trabalho aqui desenvolvido evidencia que, com o uso adequado da tecnologia, é possível alcançar um alto nível de maturidade no gerenciamento de infraestrutura de TI, elevando a eficiência, a resiliência e a confiabilidade dos ambientes corporativos.

REFERÊNCIAS

COMER, D. E. *Redes de Computadores e Internet - 6.ed*. [S.l.]: Bookman Editora, 2016. 11, 16

DOMINGOS, P. *O Algoritmo Mestre: Como a busca por um algoritmo de aprendizado de máquina vai mudar o mundo*. Rio de Janeiro: Zahar, 2017. 28

FOROUZAN, B. A.; MOSHARRAF, F. *Redes de Computadores: Uma Abordagem Top-Down*. [S.l.]: AMGH Editora, 2013. 11, 14, 15

GARTNER. Gartner. *https://www.gartner.com/en/newsroom/press-releases/2024-07-16-gartner-forecasts-worldwide-it-spending-to-grow-7-point-5-percent-in-2024. Acesso em: 17 de Março de 2025*, 2024. 15

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge, MA: MIT Press, 2016. 27

GOOGLE. Google. *https://blog.google/intl/pt-br/produtos/explore-e-encontre-respostas/apresentamos-o-gemini-20-nosso-novo-modelo-para-a-era-dos-agentes-de-ia/*, 2025. 28

LIMA, J. dos R. *Monitoramento de Redes com Zabbix: Monitore a saúde dos servidores e equipamentos de redes*. [S.l.]: Brasport, 2014. 21

MAJORS, C.; FONG-JONES, L.; MIRANDA, G. *Observability Engineering: Achieving Production Excellence*. [S.l.]: O'Reilly Media, 2022. 16

MITCHELL, T. M. *Machine Learning*. New York: McGraw-Hill, 1997. 27

RUSSELL, S.; NORVIG, P. *Inteligência Artificial*. 3. ed. Rio de Janeiro: Elsevier, 2013. 27

SOUZA, D. C. de *et al. Gerenciamento de Redes de Computadores*. [S.l.]: Sagah, 2021. 15

TANENBAUM, A.; FEAMSTER, N.; WETHERALL, D. *Redes de Computadores (coedição Bookman e Pearson)*. [S.l.]: Bookman Editora, 2021. 15