

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO



**SISTEMA DE MONITORAMENTO DISTRIBUÍDO E ANÁLISE DE DADOS EM
AMBIENTES VIRTUAIS**

VINÍCIUS RODRIGUES FERREIRA E SILVA

GOIÂNIA
2024

VINÍCIUS RODRIGUES FERREIRA E SILVA

**SISTEMA DE MONITORAMENTO DISTRIBUÍDO E ANÁLISE DE DADOS EM
AMBIENTES VIRTUAIS**

Projeto de Pesquisa elaborado para fins de avaliação da disciplina Trabalho de Conclusão de Curso 1 CMP1073 disciplina do curso de Ciência da Computação da Pontifícia Universidade Católica de Goiás.

Orientador:

Prof. Me. Gildenor de Souza Amorim
Cavalcante

Banca Examinadora:

Prof^a. Dr^a. Solange da Silva

Prof. Me. Fernando Gonçalves Abadia

GOIÂNIA

2024

DEDICATÓRIA

Dedico este trabalho à minha família, que sempre esteve ao meu lado, oferecendo apoio, amor e encorajamento incondicional. A vocês, que acreditaram em mim em todos os momentos, esta conquista é também de vocês.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, pela força e sabedoria que me guiaram ao longo deste percurso. Aos meus pais e familiares, pelo suporte emocional e incentivo constante, fundamentais para que eu persistisse diante dos desafios.

Aos professores e colegas da Pontifícia Universidade Católica de Goiás, que contribuíram com ensinamentos valiosos e apoio intelectual durante minha formação. Agradeço em especial, ao meu orientador, Prof. Me. Gildenor de Souza Amorim Cavalcante, por suas orientações precisas e motivação durante a elaboração deste trabalho.

Agradeço a todos que de uma forma ou de outra contribuíram para que este trabalho fosse realizado.

Por fim, agradeço a mim mesmo, pela dedicação e resiliência que me permitiram alcançar este marco tão importante na minha trajetória acadêmica e pessoal.

RESUMO

Este trabalho teve como objetivo geral desenvolver um sistema de monitoramento distribuído para máquinas virtuais (VMs) e servidores, focado na coleta, análise e visualização de dados para otimizar a gestão de recursos computacionais e prevenir falhas. Para isso, foi implementado o *dailyInfo*, um coletor que gera dados em formato JSON a cada 10 minutos, o *infoConnector*, que centraliza e organiza as mensagens, e um painel interativo criado com a biblioteca Streamlit, que exibe gráficos e fluxogramas para análise. Além disso, foi desenvolvido um sistema de alertas via WhatsApp para identificar situações críticas, como sobrecarga de CPU, memória e disco. O estudo permitiu concluir que, o sistema desenvolvido demonstrou ser eficiente, escalável e eficaz, permitindo maior previsibilidade, tomada de decisões mais rápidas e melhor gestão de recursos em ambientes computacionais distribuídos.

Palavras chaves: Monitoramento. Máquinas Virtuais. Análise de dados. Coleta de dados.

ABSTRACT

This study aimed to develop a distributed monitoring system for virtual machines (VMs) and servers, focusing on data collection, analysis, and visualization to optimize resource management and prevent failures. The *dailyInfo* was implemented to generate JSON data every 10 minutes, along with the *infoConnector*, which centralizes and organizes messages, and an interactive dashboard built with the Streamlit library, displaying graphs and flowcharts for analysis. Additionally, a WhatsApp alert system was developed to identify critical situations, such as CPU, memory, and disk overload. In conclusion, the system proved to be efficient, scalable, and effective, enabling greater predictability, faster decision-making, and improved resource management in distributed computational environments.

Keywords: Monitoring. Virtual Machines. Data Analysis. Data Collection.

SUMÁRIO

SUMÁRIO	7
LISTA DE FIGURAS	8
LISTA DE SIGLAS	9
1. INTRODUÇÃO	10
OBJETIVOS	12
RESULTADOS ESPERADOS	12
2. REFERENCIAL TEÓRICO	14
2.1 Conceitos de Monitoramento de dados	14
2.2 Análise de Dados	18
2.3 Mensageria com RabbitMQ	23
2.3.1. Estrutura e Componentes do RabbitMQ	24
2.3.2. Tipos de Trocas e Encaminhamento de Mensagens	25
2.3.3. Segurança e Persistência	25
2.3.4. Escalabilidade e Alta Disponibilidade	26
2.3.5. Fluxo de Funcionamento da Mensageria no RabbitMQ	26
2.3.6. Comparação com Outras Soluções de Monitoramento	26
3. MÉTODO	28
4. DESENVOLVIMENTO DO DAILYINFO	33
5. ANÁLISE DOS RESULTADOS OBTIDOS	39
5.1 Tempo de Resposta	39
5.2 Precisão dos Alertas	39
5.3 Usabilidade do Painel de Visualização	39
5.4 Impacto Geral	40
6. CONCLUSÃO	41
7. REFERÊNCIAS	43

LISTA DE FIGURAS

Figura 1 - Análise de uma VM no portal Azure.....	15
Figura 2 - Desempenho dos <i>insights</i> da VM.....	17
Figura 3 - Etapas da análise de dados.....	19
Figura 4 - Tipos de análise de dados.....	20
Figura 5 - Funcionamento de uma mensageria.....	24
Figura 6 - Fluxo de requisição por usuário via mensageria RabbitMQ.....	25
Figura 7 - Fluxo da mensagem: da máquina ao banco.....	34
Figura 8 - Gráficos de barras para monitoramento.....	35
Figura 9 - Fluxograma diário.....	36
Figura 10 - Alerta via <i>chatbot</i>	37

LISTA DE SIGLAS

- CPU** – Unidade Central de Processamento (*Central Processing Unit*)
- GDPR** – Regulamento Geral de Proteção de Dados (*General Data Protection Regulation*)
- IoT** – Internet das Coisas (*Internet of Things*)
- JSON** – Notação de Objeto JavaScript (*JavaScript Object Notation*)
- LGPD** – Lei Geral de Proteção de Dados
- TI** – Tecnologia da Informação (*Information Technology*)
- VMs** – Máquinas Virtuais (*Virtual Machines*)
- VPS** – Servidor Virtual Privado (*Virtual Private Server*)

1. INTRODUÇÃO

O monitoramento distribuído e a análise de dados em ambientes virtuais são tópicos essenciais para garantir a eficiência, segurança e confiabilidade de sistemas distribuídos. Monitoramento distribuído refere-se à coleta, análise e visualização de dados provenientes de várias fontes distribuídas, como servidores, máquinas virtuais e dispositivos internet das coisas (IoT). O objetivo é obter percepções sobre o desempenho, disponibilidade e integridade desses sistemas. Ambientes Virtuais incluem máquinas virtuais (VMs), contêineres e serviços em nuvem. Esses ambientes oferecem flexibilidade, escalabilidade e isolamento, mas também apresentam desafios em termos de monitoramento e gerenciamento (Elven Works, 2024).

Monitoramento consiste em: processos de observação contínua e análise do desempenho de sistemas de computador. Isso inclui a verificação da saúde operacional, a detecção de falhas, a previsão de problemas futuros e a garantia de que tudo funcione conforme esperado (Moreira, 2014, p.31).

A análise de dados envolve a organização e compreensão dos dados coletados durante a pesquisa. Nesse processo os dados numéricos ou qualitativos são transformados em informações relevantes para solucionar o problema de pesquisa. Em outras palavras, a análise de dados permite que você responda às perguntas que motivaram sua pesquisa (Alura, 2024).

- O principal objetivo da análise de dados é extrair informações e insights relevantes a partir dos dados. Ou seja, a ideia é extrair resultados mensuráveis e úteis. A partir de informações de diferentes fontes, é possível orientar estratégias mais assertivas, com foco no objetivo do negócio. Então, em resumo, a análise de dados é o uso de informações para a tomada de decisões de uma empresa (Alura, 2024).

Nos últimos cinco anos, pesquisas têm se concentrado em: Arquiteturas de Monitoramento: Desenvolvimento de arquiteturas escaláveis e resilientes para coletar dados de VMs distribuídas (Microsoft Learn, 2024). Telemetria e Métricas: Exploração de métricas-chave, como uso de unidade central de processamento (CPU), memória,

rede e armazenamento, para identificar gargalos e anomalias (Elastic, 2024). Análise de Logs: Uso de técnicas de processamento de logs para detectar eventos relevantes e correlacionar informações (AWS, 2024).

É relevante estudar esse tema porque abordamos eficiência operacional, pois o monitoramento adequado permite otimização de recursos, identificar falhas e tomar decisões proativas. Segurança de sistemas, para detectar atividades suspeitas ou violações de segurança em tempo real é crucial para proteger ambientes virtuais. Também escalabilidade com a crescente adoção de ambientes virtuais, entender como monitorá-los de forma eficaz é fundamental.

A eficácia da detecção e prevenção proativa do tempo de inatividade depende totalmente da sua visão do seu ambiente de TI e dos dados recentes que você possui da sua infraestrutura. Para ajudar a entender melhor como eliminar os tempos em que sua empresa fica fora do ar, você necessitará de um sistema de monitoramento de TI (Viper It, 2024).

Diante desse contexto, este projeto visa responder à seguinte questão: **Como projetar e implementar um sistema de monitoramento distribuído eficiente que garanta eficiência operacional, mas também facilite a análise compreensiva dos dados coletados?**

OBJETIVOS

OBJETIVO GERAL

- Desenvolver um sistema de monitoramento distribuído que colete dados de VMs e forneça insights acionáveis para otimização e segurança.

OBJETIVOS ESPECÍFICOS

- Elaborar uma arquitetura escalável para coleta de telemetria em ambientes virtuais.
- Analisar métricas-chave para identificar padrões e anomalias.
- Implementar um pipeline de logs para correlação de eventos.
- Criar um painel interativo usando a biblioteca Streamlit para visualização de dados.
- Testar e validar o sistema desenvolvido.

RESULTADOS ESPERADOS

Espera-se que os resultados deste trabalho possam contribuir:

- Na solução eficaz para o monitoramento e análise de máquinas virtuais, garantindo maior eficiência na gestão de recursos computacionais.
- Na detecção precoce de falhas ou sobrecargas, permitindo ações preventivas que minimizem o impacto operacional.
- Usando o painel de visualização interativo deve proporcionar uma análise clara e intuitiva dos dados, facilitando a identificação de padrões e tendências.
- Na otimização do tempo de resposta a eventos críticos, graças à integração do sistema de alertas proativos. Por fim, o projeto deve servir como base para futuras expansões, promovendo maior escalabilidade e adaptabilidade para atender a diferentes cenários e demandas do ambiente computacional.

Quanto aos aspectos metodológicos, a natureza desta pesquisa é aplicada, pois busca solucionar um problema prático relacionado ao monitoramento de máquinas virtuais. Quanto aos procedimentos técnicos, caracteriza-se como uma pesquisa bibliográfica e experimental.

Esta monografia está estruturada da seguinte maneira: neste capítulo, é apresentado o contexto do trabalho, a questão de pesquisa, o objetivo geral e os resultados esperados.

O Capítulo 2 traz o referencial teórico, abordando conceitos, definições e estudos relacionados ao tema, além de comparações com ferramentas existentes. No Capítulo 3, é descrito o método proposto, detalhando as etapas da coleta, análise e armazenamento de dados. No Capítulo 4, são apresentadas e contextualizadas as ferramentas e técnicas utilizadas no sistema, como o dailyInfo, infoConnector e o painel de visualização. O Capítulo 5 traz a análise dos resultados obtidos, incluindo o desempenho do sistema, a precisão dos alertas e a avaliação da usabilidade. Finalmente, o Capítulo 6 apresenta as considerações finais do trabalho, destacando as contribuições do projeto e sugestões para trabalhos futuros.

2. REFERENCIAL TEÓRICO

Este capítulo traz os conceitos e fundamentos de monitoramento e análise de dados, explorando suas inter-relações e relevância no contexto da tecnologia da informação.

2.1 CONCEITOS DE MONITORAMENTO DE DADOS

Monitoramento é o processo contínuo de observação, coleta e análise de dados sobre um sistema ou atividade específica ao longo do tempo, com o objetivo de avaliar seu desempenho, detectar anomalias e garantir que os objetivos estabelecidos sejam alcançados. Em diversas áreas, como tecnologia da informação, saúde pública e meio ambiente, o monitoramento permite a identificação precoce de problemas, facilitando intervenções rápidas e eficazes.

De acordo com Turnbull (2014), no contexto de TI, o monitoramento envolve a supervisão constante de sistemas e serviços para assegurar sua confiabilidade e eficiência. A Organização Mundial da Saúde (OMS) define o monitoramento em saúde pública como a coleta sistemática e contínua de dados relevantes para a saúde, seguida da análise e disseminação dessas informações para orientar políticas e ações de saúde.

A Organização Internacional para Padronização (ISO) diz que, o monitoramento é "a observação sistemática de mudanças em uma condição específica ao longo do tempo" (ISO, 2024). Em tecnologia da informação, refere-se à supervisão contínua dos sistemas de TI (como servidores, redes, aplicativos) para garantir que funcionem corretamente e detectar falhas ou anomalias, conforme ilustrado na Figura 1.

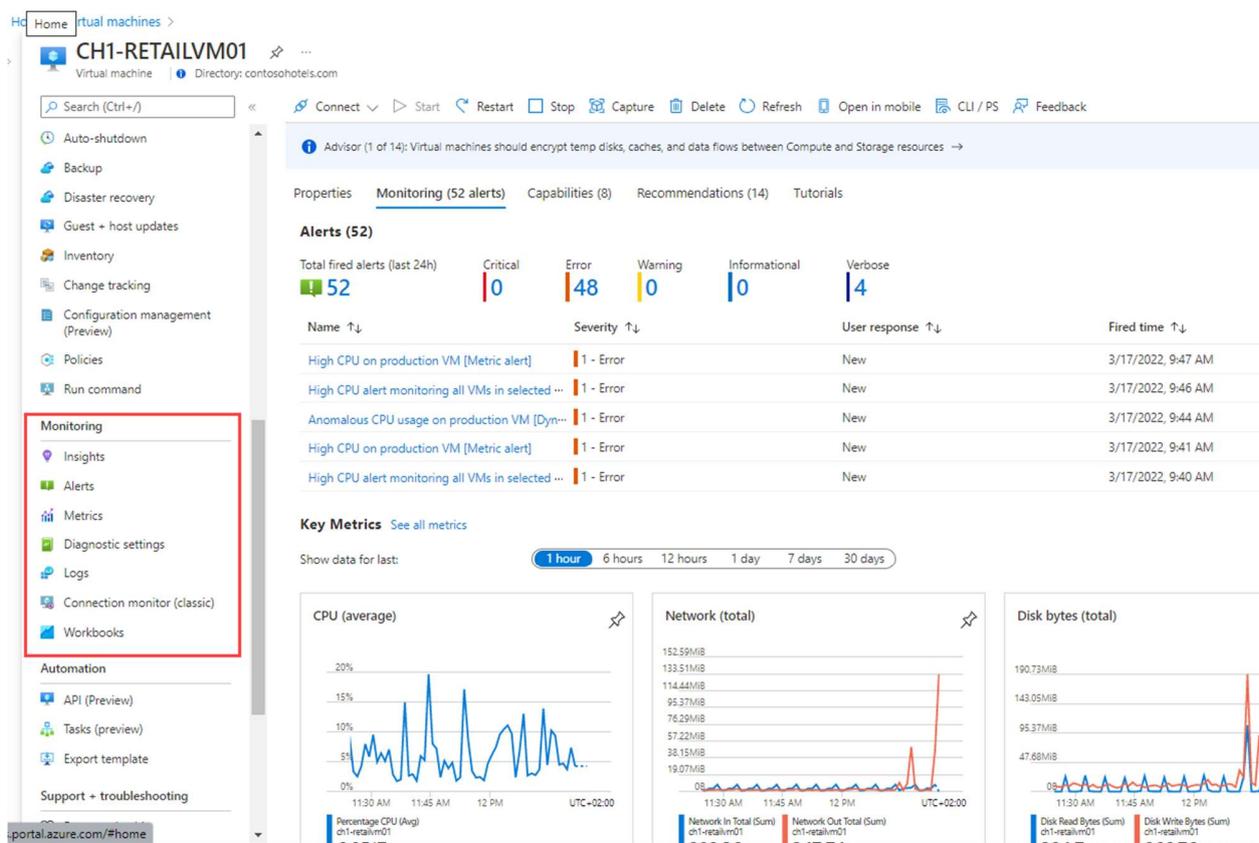


Figura 1 - Análise de uma VM no portal Azure.

Fonte: Microsoft (2024).

O monitoramento de dados é um pilar essencial em diversas disciplinas, especialmente nas áreas de tecnologia da informação e engenharia de software. No contexto de TI, ele representa uma abordagem proativa para garantir a continuidade de serviços e a segurança de sistemas. A função do monitoramento é coletar dados relevantes e avaliar métricas em tempo real, permitindo que as equipes identifiquem e analisem padrões de comportamento nos sistemas e prevejam possíveis falhas antes que impactem o serviço. Com o aumento do uso de sistemas distribuídos e ambientes em nuvem, o monitoramento de dados se torna fundamental, pois abrange tanto a detecção de eventos críticos como o rastreamento do desempenho de todos os recursos integrados de uma infraestrutura. Além disso, com o uso de algoritmos e ferramentas avançadas, é possível criar alertas automatizados que informam os responsáveis sobre

qualquer comportamento inesperado, facilitando uma resposta rápida e eficaz para a manutenção da estabilidade dos sistemas (Turnbull, 2014).

No monitoramento de máquinas virtuais (VMs), a coleta de dados se estende ao desempenho específico de cada máquina virtual e aos recursos físicos que as suportam, como servidores, armazenamento e redes. As VMs oferecem uma camada adicional de complexidade no monitoramento, já que compartilham os mesmos recursos físicos e exigem uma gestão cuidadosa de alocação para evitar a saturação de recursos. É crucial monitorar aspectos como uso de CPU, memória, espaço em disco e rendimento de rede, pois esses indicadores ajudam a avaliar a carga das VMs em relação ao hardware subjacente. Ferramentas de monitoramento de VMs geralmente incluem visualizações em tempo real e históricos de desempenho, que permitem observar a variação de uso dos recursos e possibilitam o ajuste dinâmico da alocação para equilibrar a carga de trabalho. Dessa forma, as equipes podem otimizar a utilização dos recursos físicos, garantindo que todas as VMs continuem funcionando de maneira eficiente (Turnbull, 2014).

Além disso, o monitoramento de VMs deve ser complementado com verificações de integridade de aplicativos executados nessas máquinas. Muitas vezes, é preciso monitorar também os logs do sistema, que fornecem informações detalhadas sobre eventos que ocorrem dentro das VMs, tais como: falhas de execução, uso de APIs, acessos e atualizações de segurança. Com a análise contínua desses registros, é possível detectar ataques ou comportamentos anômalos que poderiam comprometer a segurança do ambiente. Em um sistema distribuído com várias VMs, o monitoramento centralizado, que utiliza uma plataforma de observação consolidada, facilita a visualização de dados de diversas fontes e o gerenciamento das máquinas como um sistema coeso. Isso se traduz em uma maior eficiência na detecção de ameaças e na manutenção de uma postura de segurança resiliente (ISO, 2024).

Dados não numéricos, como eventos, são armazenados em Logs, enquanto as métricas contêm apenas dados numéricos, amostrados em intervalos regulares. Os dados numéricos podem ser registrados tanto em logs quanto em métricas, permitindo análises diversificadas e dando suporte a diferentes tipos de alertas. O agente do Azure Monitor envia dados de desempenho do sistema operacional convidado para as Métricas,

os Logs ou ambos. Além disso, os insights de VM também encaminham dados de desempenho do sistema operacional convidado aos logs (Microsoft, 2024). Conforme apresentado na Figura 2.

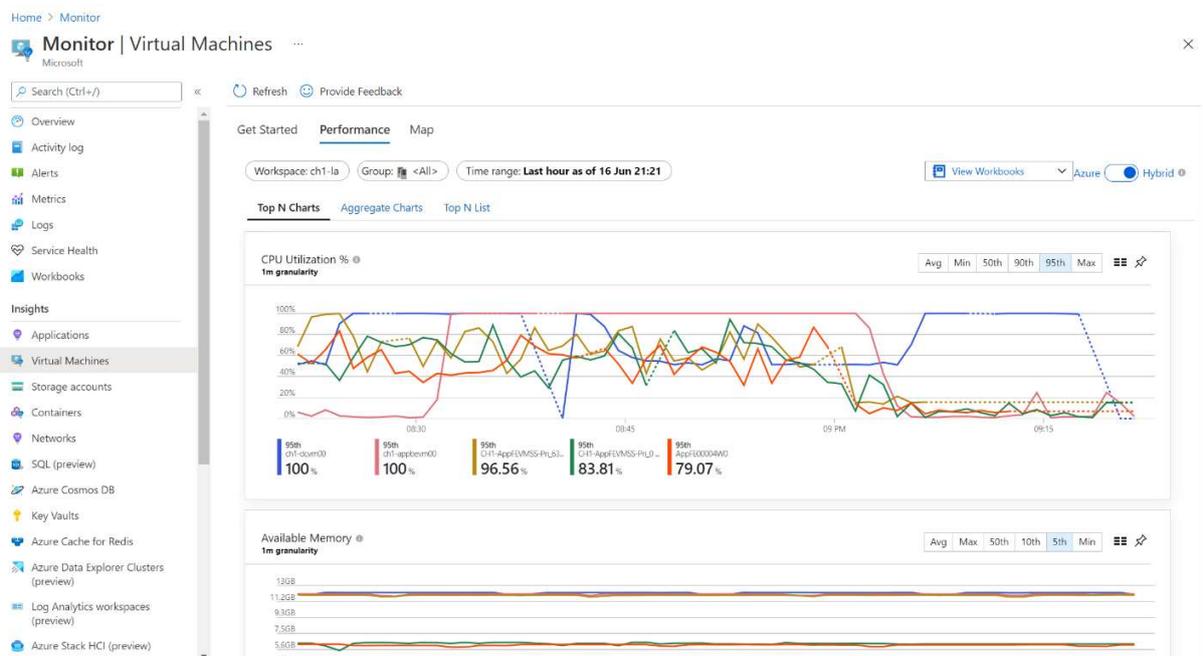


Figura 2 - Desempenho dos insights da VM.

Fonte: Microsoft (2024).

Em suma, o monitoramento de dados em ambientes com máquinas virtuais é uma atividade complexa, mas essencial para o desempenho e segurança dos sistemas de TI modernos. Ele permite que as equipes de TI tenham controle sobre cada componente do sistema, proporcionando uma visão completa do uso de recursos e ajudando a gerenciar a carga de trabalho de forma eficiente. Ferramentas de monitoramento, como o Prometheus, Nagios e Zabbix, são amplamente utilizadas para acompanhar o desempenho de VMs, garantindo que as métricas sejam monitoradas de forma contínua e que as equipes possam agir com rapidez quando houver necessidade (Turnbull, 2014).

No contexto de segurança, o monitoramento de dados desempenha um papel crucial na identificação de atividades anômalas que podem representar ameaças ao ambiente computacional. Sistemas de monitoramento podem ser configurados para rastrear eventos específicos, como acessos não autorizados, tentativas de login

malsucedidas e alterações inesperadas em arquivos críticos. Esses eventos, quando correlacionados, ajudam a detectar padrões de comportamento maliciosos, permitindo respostas rápidas a incidentes de segurança. Ferramentas modernas, como o Elastic Stack (ELK), oferecem funcionalidades avançadas de análise de logs que podem detectar ataques cibernéticos em tempo real e fornecer insights para reforçar a proteção do sistema (Elastic, 2024).

Além disso, o monitoramento contínuo de dados é essencial para cumprir regulamentos de segurança e privacidade, como a Lei Geral de Proteção de Dados (LGPD) e o *General Data Protection Regulation* (GDPR). Esses regulamentos exigem que as organizações mantenham registros detalhados de acessos e eventos, garantindo que informações sensíveis sejam tratadas com responsabilidade. Ferramentas como Splunk e Graylog permitem a centralização e análise de logs de segurança, oferecendo visibilidade completa do ambiente de TI e permitindo auditorias rápidas e eficientes. De acordo com Provost e Fawcett (2013), a análise proativa de logs e métricas em um contexto de segurança possibilita não apenas a detecção de problemas, mas também a prevenção de incidentes, ao identificar vulnerabilidades antes que sejam exploradas (Provost, 2013).

2.2 ANÁLISE DE DADOS

A análise de dados é o processo de examinar, limpar, transformar e modelar dados com o objetivo de descobrir informações úteis, tirar conclusões e apoiar a tomada de decisões. Este processo pode envolver várias técnicas estatísticas, algoritmos de aprendizado de máquina e ferramentas de visualização de dados para identificar padrões, tendências e relações nos dados coletados. Segundo Provost e Fawcett (2013), a análise de dados é crucial para transformar grandes volumes de dados brutos em insights acionáveis que podem melhorar o desempenho organizacional e a eficiência operacional. No contexto empresarial, a análise de dados ajuda as organizações a prever comportamentos futuros, otimizar operações e desenvolver estratégias baseadas em dados precisos e atualizados, conforme ilustrado na Figura 3.



Figura 3 - Etapas da análise de dados

Fonte: ZOOX SMART DATA (2024).

Os principais tipos de análise de dados são descritiva, diagnóstica, preditiva e prescritiva, e cada uma desempenha um papel específico na interpretação de informações para a tomada de decisões. A análise descritiva fornece um resumo dos dados coletados, destacando tendências e padrões observados no passado, sendo útil para entender o estado atual de um sistema ou processo. Já a análise diagnóstica vai além, investigando as causas por trás dos eventos ou padrões identificados na etapa descritiva. A análise preditiva, por sua vez, utiliza modelos estatísticos e algoritmos de machine learning para prever futuros comportamentos e eventos com base em dados históricos. Por último, a análise prescritiva orienta as decisões futuras ao sugerir ações a serem tomadas, baseando-se em simulações e otimizações para alcançar os melhores resultados possíveis. Esses quatro tipos de análise trabalham juntos para transformar dados brutos em insights estratégicos que auxiliam organizações em diversas áreas a tomar decisões embasadas e proativas, conforme ilustrado na Figura 4 (DIO.ME, 2024).



Figura 4 - Tipos de análise de dados

Fonte: Dio (2024).

A análise de dados não apenas envolve técnicas para interpretar grandes volumes de informações, mas também exige um entendimento profundo do contexto em que os dados foram coletados. Ferramentas como estatísticas descritivas, inferenciais, aprendizado de máquina e visualização de dados são frequentemente utilizadas para identificar padrões e tendências significativas. Além disso, a qualidade dos dados é crucial; processos como limpeza de dados e tratamento de valores ausentes são fundamentais para garantir a precisão das análises. A análise de dados permite a organizações transformar dados brutos em insights valiosos, possibilitando a tomada de decisões informadas, a identificação de oportunidades de melhoria e a antecipação de futuros desenvolvimentos. Isso é particularmente relevante em um mundo onde a quantidade de dados gerados está crescendo exponencialmente, tornando a capacidade de analisá-los eficientemente uma vantagem competitiva essencial (Alura, 2024).

A análise de dados, segundo o DataCamp, é o processo de examinar, organizar e interpretar informações brutas para transformar dados em insights acionáveis. Esse processo envolve várias etapas, desde a coleta e limpeza dos dados até a aplicação de técnicas analíticas para extrair padrões e tendências significativas. A análise de dados é

amplamente utilizada em diferentes setores para fundamentar decisões estratégicas, melhorar a eficiência operacional e prever comportamentos futuros. Ferramentas modernas, como Python e R, juntamente com bibliotecas especializadas, permitem que analistas automatizem tarefas e criem visualizações avançadas, simplificando a interpretação de dados complexos (Datacamp, 2024).

O site também destaca que a análise de dados é um campo essencial no contexto atual, onde a quantidade de informações disponíveis cresce exponencialmente. Técnicas como análise descritiva, diagnóstica, preditiva e prescritiva são fundamentais para entender o passado, diagnosticar causas, prever eventos futuros e sugerir ações específicas. Por exemplo, em uma organização, essas análises podem ser utilizadas para identificar gargalos em processos ou até mesmo para prever tendências de mercado. A DataCamp enfatiza que a chave para uma análise de dados eficaz é garantir a qualidade dos dados utilizados e a escolha correta das ferramentas analíticas que melhor atendem ao problema em questão (Datacamp, 2024).

O monitoramento e a análise de dados estão intrinsecamente relacionados, pois o monitoramento contínuo de sistemas ou atividades gera um grande volume de dados que necessita de análise para ser transformado em informações úteis. Enquanto o monitoramento foca na coleta e supervisão constante dos dados para garantir o desempenho e a detecção de anomalias, a análise de dados foca na interpretação desses dados para extrair insights e apoiar a tomada de decisões. Por exemplo, em TI, os dados de monitoramento de rede podem ser analisados para identificar padrões de tráfego anômalos que indicam possíveis falhas ou ataques de segurança, permitindo intervenções preventivas baseadas em evidências concretas. Assim, a análise de dados complementa o monitoramento, permitindo uma compreensão mais profunda e ações mais informadas (Turnbull, 2014).

De acordo com o Google Cloud, a análise de dados com inteligência artificial (IA) potencializa a extração de insights complexos, automatizando processos que antes exigiam intervenção manual. A integração de IA na análise de dados permite a aplicação de algoritmos de aprendizado de máquina (machine learning) para identificar padrões, prever tendências e até mesmo recomendar ações baseadas nos dados analisados. Essa abordagem é particularmente eficaz em cenários com grandes volumes de dados,

como no monitoramento de comportamento de usuários ou na detecção de anomalias em tempo real. Combinada com ferramentas analíticas avançadas, a IA transforma grandes quantidades de informações em insights úteis de maneira mais rápida e eficiente (Google Cloud, 2024).

O site destaca ainda que a análise de dados com IA pode ser personalizada para atender às necessidades específicas de cada setor. Por exemplo, no setor financeiro, a IA pode ajudar a detectar fraudes, enquanto no varejo pode identificar preferências de consumidores e recomendar estratégias de marketing. Além disso, as soluções de IA disponíveis no Google Cloud integram-se facilmente a diferentes fontes de dados, oferecendo suporte para a construção de pipelines robustos de análise e aprendizado. Isso torna a análise de dados com IA uma ferramenta indispensável para empresas que desejam tomar decisões estratégicas baseadas em informações precisas e previsões confiáveis (Google Cloud, 2024).

O site da FM2S destaca a importância de estruturar adequadamente uma análise de dados para garantir a precisão e relevância dos resultados obtidos. Para isso, o processo deve ser bem definido e seguido de forma organizada, desde a coleta de dados até a interpretação dos resultados. Uma análise de dados eficiente exige, antes de tudo, uma compreensão clara do problema a ser resolvido e dos objetivos da análise. Este entendimento inicial guia a escolha das variáveis e das fontes de dados mais relevantes, garantindo que cada etapa do processo contribua para a geração de insights acionáveis (FM2S, 2024).

O segundo passo essencial é a coleta e organização dos dados, segundo o site da FM2S, que deve ser realizada de maneira criteriosa e consistente. Os dados podem vir de diversas fontes, como planilhas, bancos de dados e APIs, e devem ser cuidadosamente reunidos e tratados para evitar inconsistências e lacunas que possam comprometer a análise. Essa fase também envolve a limpeza dos dados, processo no qual valores ausentes ou incoerentes são ajustados ou eliminados. Essa preparação é essencial para assegurar que a base de dados final seja confiável e representativa do fenômeno em estudo (FM2S, 2024).

Em seguida, a análise propriamente dita permite que os dados sejam transformados em informações relevantes. Nesta fase, o uso de ferramentas e técnicas

de análise, como estatísticas descritivas, visualizações de dados e modelos preditivos, ajuda a interpretar padrões, tendências e relações entre as variáveis. Ferramentas como Python, R e Power BI são mencionadas como eficazes para visualizar e manipular dados, tornando o processo mais acessível e compreensível para diferentes perfis de profissionais, inclusive aqueles sem uma formação estatística avançada (FM2S, 2024).

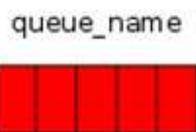
Finalmente, a interpretação dos resultados é a etapa que conecta a análise aos objetivos iniciais, orientando a tomada de decisões. Os insights obtidos devem ser contextualizados e apresentados de forma clara e objetiva, facilitando a compreensão de todos os envolvidos. A FM2S enfatiza que a comunicação dos resultados é um passo crucial, pois a maneira como as descobertas são expostas pode influenciar diretamente a aplicação das informações e o impacto das decisões baseadas na análise (FM2S, 2024).

2.3 MENSAGERIA COM RABBITMQ

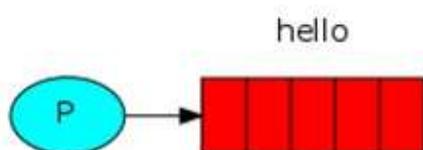
O RabbitMQ é uma plataforma robusta de mensageria projetada para facilitar a comunicação entre diferentes sistemas ou componentes, especialmente em arquiteturas distribuídas. Ele opera com base no protocolo AMQP (Advanced Message Queuing Protocol), permitindo o envio de mensagens entre produtores (sistemas que enviam dados) e consumidores (sistemas que recebem dados) de maneira assíncrona e confiável. A sua estrutura de filas assegura que as mensagens sejam armazenadas temporariamente até que sejam entregues aos consumidores, tornando o RabbitMQ altamente eficiente para gerenciar grandes volumes de mensagens sem perder dados durante o processo (Cedro Technologies, 2024).

Uma mensagem é dividida em duas partes:

- *Payload* – é o corpo com os dados que serão transmitidos. Suporta vários tipos de dados como um array json até um filme mpeg.
- *Label* – é responsável pela descrição do payload e também como o RabbitMQ saberá quem irá receber a mensagem, conforme apresentado na Figura 5.

Fila:

Onde as mensagens ficam e são retiradas pelos consumers.

Publisher:

É o responsável por incluir cada nova mensagem na fila, ou seja enviar a mensagem.

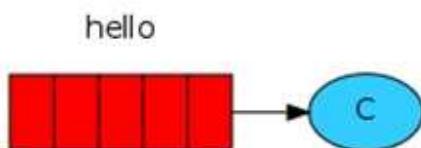
Consumer:

Figura 5 - Funcionamento de uma mensageria

Fonte: Cedro Technologies (2024).

2.3.1. Estrutura e Componentes do RabbitMQ

A estrutura básica do RabbitMQ é composta por alguns elementos principais: os produtores, as filas, as trocas (exchanges) e os consumidores. Os produtores são responsáveis por enviar mensagens para uma "troca" no RabbitMQ, e essa troca, por sua vez, direciona as mensagens para as filas apropriadas, seguindo regras predefinidas (chamadas de bindings). As filas agem como repositórios temporários para as mensagens, até que um consumidor as receba e processe. Esse fluxo simplificado entre produtores e consumidores por meio de trocas e filas torna o RabbitMQ extremamente flexível e adaptável para diferentes cenários de comunicação (GabrielGCJ, 2024).

2.3.2. Tipos de Trocas e Encaminhamento de Mensagens

O RabbitMQ oferece diferentes tipos de trocas, que controlam como as mensagens são roteadas para as filas: *direct*, *topic*, *fanout* e *headers*. A troca *direct* encaminha mensagens diretamente para filas específicas baseadas em uma chave de roteamento (routing key). No tipo *topic*, é possível definir padrões para o roteamento das mensagens, o que permite um encaminhamento mais dinâmico e flexível. Já a troca *fanout* distribui as mensagens para todas as filas conectadas, sem considerar chaves de roteamento, enquanto *headers* permite o uso de parâmetros no cabeçalho da mensagem para definição de rotas. Esses tipos de troca garantem que o RabbitMQ seja eficaz em diferentes configurações de sistemas, conforme ilustrado na Figura 6 (Gabrielgcj, 2024).

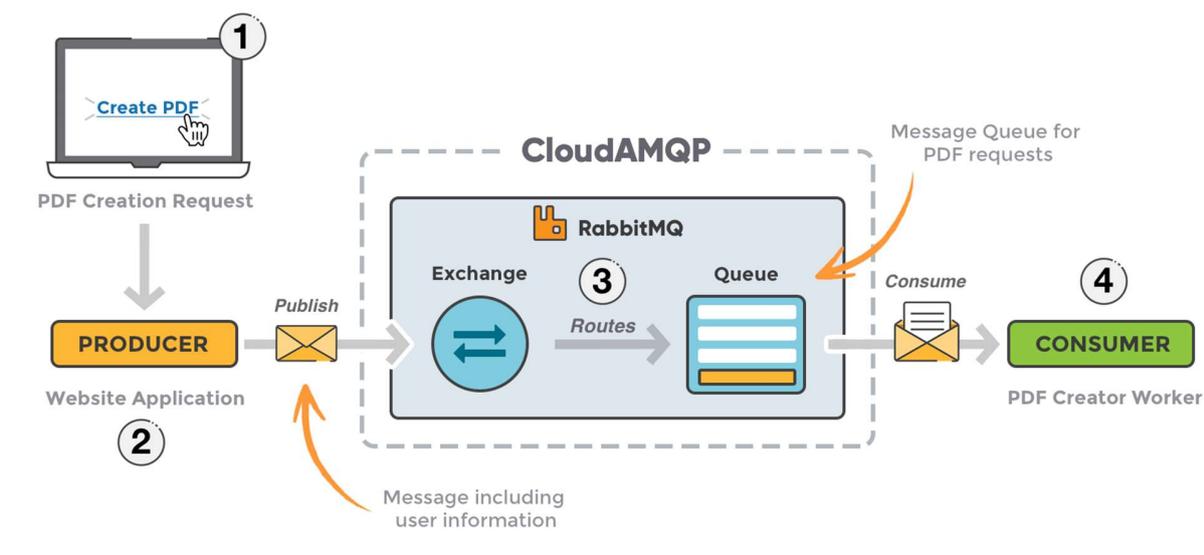


Figura 6 - Fluxo de requisição por usuário via mensageria RabbitMQ.

Fonte: CLOUDAMQP.

2.3.3. Segurança e Persistência

O RabbitMQ possui mecanismos para garantir que as mensagens sejam entregues com segurança e persistência. Para assegurar que uma mensagem não se perca no meio do caminho, o sistema permite marcar as mensagens como persistentes, o que significa que elas são armazenadas em disco e mantidas mesmo em caso de reinicialização do servidor. Além disso, o RabbitMQ implementa políticas de confirmação, tanto do lado do produtor quanto do lado do consumidor, para garantir que as mensagens

sejam enviadas e processadas com sucesso. Esse nível de segurança é particularmente útil em ambientes que exigem alta confiabilidade (Alibaba Cloud, 2024).

2.3.4. Escalabilidade e Alta Disponibilidade

O RabbitMQ é projetado para ser escalável, permitindo a criação de clusters que distribuem a carga de trabalho entre vários nós. Com isso, é possível suportar um volume elevado de mensagens, aumentando a capacidade de processamento. Além disso, ele oferece suporte para alta disponibilidade por meio da replicação das filas em diferentes nós do cluster, garantindo que, se um nó falhar, outro assumirá a responsabilidade, mantendo a operação do sistema sem interrupções. Esse modelo de escalabilidade torna o RabbitMQ uma escolha popular para aplicações que demandam performance e tolerância a falhas (Groffe, 2024).

2.3.5. Fluxo de Funcionamento da Mensageria no RabbitMQ

O fluxo de funcionamento do RabbitMQ é composto por várias etapas:

1. O produtor envia uma mensagem para uma troca específica.
2. A troca utiliza a chave de roteamento e encaminha a mensagem para a fila correta.
3. A fila mantém a mensagem até que um consumidor esteja disponível para recebê-la.
4. O consumidor recebe e processa a mensagem, marcando-a como "entregue".
5. Caso o processamento falhe, o RabbitMQ permite reencaminhar a mensagem ou mantê-la na fila para novas tentativas Cedro Technologies (2024).

2.3.6. Comparação com Outras Soluções de Monitoramento

Para avaliar a eficácia do sistema proposto, foi realizada uma análise comparativa com ferramentas de monitoramento amplamente utilizadas, como Prometheus, Zabbix e Nagios. Essas ferramentas são reconhecidas no mercado por sua robustez e flexibilidade, porém apresentam características e requisitos técnicos que diferem da solução aqui desenvolvida (Prometheus, Zabbix, Nagios, 2024).

- **Prometheus:** Focado na coleta de métricas baseadas em séries temporais, é amplamente utilizado em ambientes de grande escala. No entanto, requer configurações avançadas e integrações com outras ferramentas, como Grafana, para a visualização dos dados. O sistema proposto, por outro lado, adota uma abordagem mais simples e direta, utilizando o Streamlit para visualizações rápidas e intuitivas (Prometheus, 2024)
- **Zabbix:** Oferece monitoramento detalhado com suporte para alertas e visualizações integradas. Embora seja uma solução robusta, sua curva de aprendizado é mais elevada, exigindo maior esforço inicial de configuração. Em comparação, a arquitetura baseada em dailyInfo e infoConnector simplifica o processo de implementação, sendo mais acessível para ambientes menores ou com menos recursos (Zabbix, 2024).
- **Nagios:** Conhecido por seu foco em monitoramento de disponibilidade e eventos, é uma ferramenta eficaz para identificar falhas em tempo real. Contudo, seu modelo de configuração baseado em arquivos pode ser complexo para operações dinâmicas. A solução proposta se destaca por utilizar RabbitMQ para mensageria, permitindo um fluxo dinâmico e escalável de dados (Nagios, 2024).

3. MÉTODO

Essa pesquisa, segundo sua natureza, consiste em um trabalho de resumo de assunto, de acordo com Waslawick, um resumo de assunto é uma síntese breve e concisa que apresenta os pontos principais e as ideias centrais de um texto ou documento mais extenso. O objetivo é fornecer uma visão geral do conteúdo, destacando os aspectos mais importantes e eliminando detalhes menores ou supérfluos (Wazlawick, 2014).

O objetivo é compreender o tema por meio da investigação e observação das informações disponíveis, buscando identificar suas causas e explicações (Wazlawick, 2014).

Quanto aos objetivos, trata-se de uma pesquisa exploratória. Nas palavras do autor Gil, uma pesquisa exploratória é um tipo de pesquisa que tem como principal objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses. Esse tipo de pesquisa é frequentemente utilizado quando o tema ou problema de pesquisa é pouco conhecido ou pouco estudado. “As pesquisas exploratórias têm como propósito proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses” (Gil, 2017).

O planejamento de uma pesquisa exploratória é flexível, considerando diversos aspectos relacionados ao fato ou fenômeno estudado (Wazlawick, 2014).

As coletas de resultado de pesquisa envolvem etapas de:

1. Coleta de Dados das VPS;
2. Montagem do Painel Streamlit.

A flexibilidade dos estudos exploratórios dificulta sua categorização, mas é possível identificar diferentes abordagens, como pesquisas bibliográficas, estudos de casos e levantamentos de campo, que podem ser considerados estudos exploratórios (Gil, 2017). Em grande parte, as pesquisas acadêmicas assumem um caráter exploratório, buscando maior familiaridade com o problema ou a construção de hipóteses (Gil, 2017).

Conforme Gil (2017), a pesquisa bibliográfica segue um conjunto de etapas essenciais para sua efetividade:

- a) Escolha do Tema: O tema escolhido foi monitoramento de máquinas virtuais.

- b) Levantamento Bibliográfico Preliminar: Com base na área de monitoramento de máquinas virtuais e análise de dados, o levantamento bibliográfico foi realizado para aprofundar o entendimento sobre técnicas, ferramentas e metodologias aplicadas nesse contexto. Esse processo incluiu a consulta a artigos científicos, dissertações e materiais técnicos, permitindo definir de forma precisa o problema de pesquisa e identificar lacunas que o sistema proposto busca solucionar, como a integração eficiente de coleta, mensageria e análise de dados em tempo real.
- c) Formulação do problema: Como projetar e implementar um sistema de monitoramento distribuído eficiente que garanta eficiência operacional, mas também facilite a análise compreensiva dos dados coletados?
- d) Busca das Fontes: Utilizando bases de dados acadêmicas especializadas, o pesquisador realizou uma seleção criteriosa de artigos científicos, teses, dissertações e trabalhos de conferências relevantes. Foram consultadas fontes como a ACM Digital Library, IEEE Xplore, SpringerLink e ScienceDirect, reconhecidas por sua credibilidade na área de ciência da computação e tecnologia da informação. Por exemplo, estudos como *“Improving the scalability of data center networks with traffic-aware virtual machine placement”* de Meng et al. (2010) e *“Xen and the art of virtualization”* de Barham et al. (2003) oferecem contribuições valiosas sobre técnicas de monitoramento e escalabilidade em ambientes virtuais. Essa abordagem garantiu que a revisão teórica fosse embasada em materiais de alta qualidade e relevância.
- e) Leitura do Material: O material selecionado foi cuidadosamente lido e analisado de forma crítica, visando identificar os aspectos fundamentais relacionados ao tema de pesquisa.
- f) Fichamento: O fichamento foi realizado, pois este processo é essencial para embasar teoricamente o projeto de pesquisa e garantir a qualidade dos resultados obtidos.
- g) Redação do Texto: A escrita da monografia TCC II.

Quanto aos procedimentos técnicos, esta pesquisa é bibliográfica e experimental.

A pesquisa bibliográfica, conforme Wazlawick (2014), baseia-se em material já

publicado, incluindo livros, revistas, teses, dissertações e anais de eventos científicos, bem como mídias digitais.

As pesquisas experimentais são aquelas que envolvem a manipulação de variáveis pelo pesquisador para observar e analisar os efeitos dessas manipulações em outras variáveis. Essas pesquisas são realizadas em condições controladas para garantir que as mudanças observadas nas variáveis dependentes sejam causadas exclusivamente pelas variações introduzidas nas variáveis independentes (Gil, 2017). Logo, de acordo com Gil (2017) esta pesquisa experimental segue os seguintes processos:

- a) **Formulação do Problema:** Como projetar e implementar um sistema de monitoramento distribuído eficiente que garanta eficiência operacional, mas também facilite a análise compreensiva dos dados coletados?
- b) **Definição do plano experimental:** O plano experimental foi cuidadosamente elaborado para garantir a validação do sistema proposto e permitir a replicação do experimento. Primeiramente, definiu-se que o foco seria testar a eficiência e a escalabilidade do sistema de monitoramento distribuído, considerando cenários de diferentes cargas de trabalho. Para isso, foram estabelecidos dois grupos principais: o grupo experimental, composto por 08 (oito) máquinas virtuais monitoradas continuamente pelo sistema, e um grupo de controle, no qual nenhuma solução de monitoramento foi aplicada para avaliar o impacto direto do sistema. As ferramentas e técnicas utilizadas incluíram scripts desenvolvidos em Python, integrados com bibliotecas específicas como psutil (para coleta de métricas de uso de CPU, memória e disco) e Streamlit (para visualização de dados). Além disso, o sistema foi configurado para enviar os dados coletados em formato JSON via RabbitMQ, garantindo a integridade e a eficiência na transmissão das mensagens. O PostgreSQL foi utilizado como banco de dados para armazenamento das métricas e logs, enquanto o painel de controle permitiu a análise visual dos resultados. Os dados foram coletados a cada 10 minutos, simulando cenários reais de operação. Durante o experimento, foram monitorados o tempo de resposta do sistema, a precisão na geração de alertas e a capacidade de lidar com cargas crescentes de

trabalho. Este planejamento meticuloso assegurou que os resultados fossem válidos e confiáveis, permitindo conclusões sólidas sobre a eficácia do sistema na melhoria da eficiência operacional das VPSs.

- c) **Determinação do ambiente:** O ambiente de pesquisa foi configurado integralmente em um ambiente virtual, utilizando máquinas virtuais (VPSs) previamente existentes e que apresentavam problemas frequentes de sobrecarga. Todas no sistema operacional Ubuntu 22 com Python versão 3.7. Essas configurações foram escolhidas para simular cenários reais de alta demanda em ambientes corporativos, garantindo que o sistema fosse testado em condições representativas. As máquinas foram submetidas a cargas de trabalho intensivas, como processos de cálculo de dados, operações de leitura e escrita no disco, e tráfego de rede, para avaliar a robustez do sistema de monitoramento. O ambiente virtual foi controlado de maneira rigorosa, garantindo que as variáveis externas não interferissem nos resultados e permitindo a replicação dos testes em diferentes momentos.

O cenário experimental contou com 08 (oito) VPSs configuradas com especificações robustas para garantir a consistência dos testes. Cada VPS apresentava as seguintes configurações:

- **Processador (CPU):** AMD Ryzen 5 5600G 6-Core, com clock de 3.9 GHz.
- **Memória RAM:** 64 GB DDR4.
- **Armazenamento:** SSD de 480 GB.
- **Sistema Operacional:** Ubuntu 22.
- **Conexão de Rede:** Banda Larga de 1 Gbps

- d) **Coleta de dados:** Os testes realizados para validar o sistema foram baseados em uma coleta de dados automatizada. O processo foi configurado para ocorrer a cada dez minutos, utilizando o crontab, uma ferramenta do Linux para agendamento de tarefas recorrentes. O coletor, desenvolvido em Python, gerava os dados em formato JSON (JavaScript Object Notation) e os enviava para um controlador de informações centralizado. O controlador recebia os dados e realizava sua inserção no banco de dados. Este armazenamento

servia como base para a criação do painel de visualização, que consolidava as informações de todas as máquinas monitoradas. Esse processo, automatizado e contínuo, garantiu que os dados estivessem sempre atualizados para análise e acompanhamento em tempo real.

Tempo desde o envio do JSON pelo dailyinfo até a inserção do banco de dados:

- Com 10 máquinas: tempo médio = 1,5 segundos
- Com 50 máquinas: tempo médio = 1,9 segundos

e) **Análise e interpretação dos dados:** O processo de análise e interpretação foi feito totalmente após o coletor enviar as informações ao controlador, via mensageria. Dessa forma pode se ter um servidor com um serviço que recebe todos os JSON de todas as máquinas contempladas. O controlador faz a divisão do JSON, inserindo essas informações dentro das tabelas do banco de dados, organizadamente.

f) **Redação do relatório:** escrita do TCC II.

4. DESENVOLVIMENTO DO DAILYINFO

Este sistema denominado *dailyinfo* tem como objetivo o monitoramento de máquinas virtuais (VMs e VPSs) que frequentemente apresentavam sobrecarga. Para resolver esse problema, tornou-se necessário desenvolver um fluxo de extração e centralização de informações sobre o estado dessas máquinas em momentos específicos ao longo do dia. Com base nisso, foi idealizado o '*dailyInfo*', um código em Python que, em intervalos de 10 minutos, coleta e envia um relatório detalhado sobre a condição da máquina em formato JSON que possui a seguinte estrutura:

```
{"Dia e Hora": diaHora,  
  "Hostname": hostname,  
  "IP": machinelp,  
  "Cpu": cpu,  
  "Memoria": mem_info,  
  "Disco": disk_info,  
  "Uptime": tempo}
```

Essa estrutura permite a captura de informações essenciais, como data e hora da coleta, nome e endereço IP da máquina, utilização de CPU, memória, disco e o tempo de atividade da máquina. Esse fluxo de dados contribui para um monitoramento contínuo e dinâmico, possibilitando ações preventivas e corretivas quando os recursos excedem limites aceitáveis.

Para gerenciar o destino das mensagens geradas pelo '*dailyInfo*', considerou-se o uso de um banco de dados PostgreSQL, onde os registros JSON poderiam ser armazenados em uma única coluna, facilitando o acesso rápido e direto aos dados. Contudo, a necessidade de monitorar até 100 máquinas simultaneamente trouxe um desafio: a criação de uma conexão direta para cada máquina geraria um volume excessivo de conexões com o banco, o que poderia comprometer a performance, causando lentidão e dificultando o acesso concorrente aos dados.

Para contornar essa questão, foi desenvolvido o *infoConnector*, também implementado em Python, cuja função é centralizar as mensagens e controlar a inserção dos registros no banco de dados. O *infoConnector* atua como um intermediário entre as máquinas monitoradas e o banco de dados, recebendo as mensagens de forma

centralizada e organizando as inserções. Dessa forma, é possível reduzir significativamente o número de conexões simultâneas ao banco de dados, otimizando o desempenho e garantindo que o sistema suporte a escalabilidade do monitoramento de múltiplas máquinas, conforme apresentado na Figura 7.

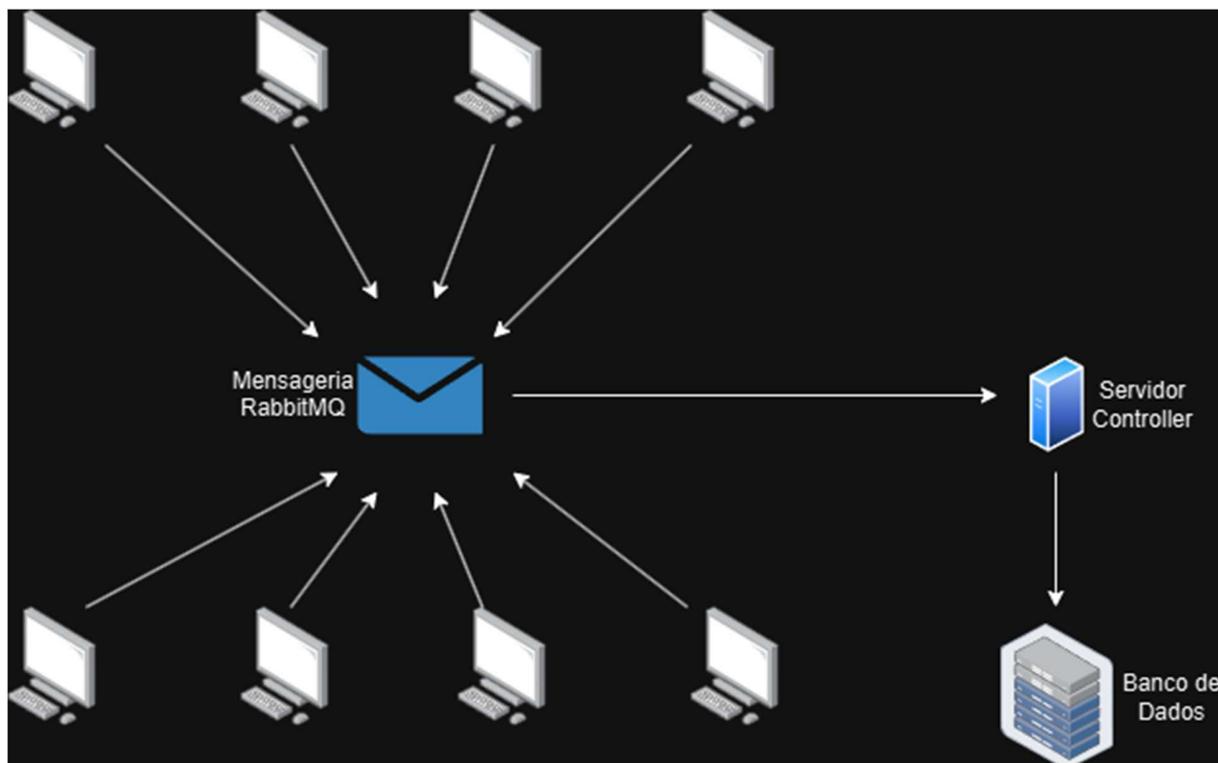


Figura 7 - Fluxo da mensagem: da máquina ao banco

Fonte: Elaboração própria.

O fluxo de mensageria foi estruturado de modo que o *infoConnector* crie e gerencie uma fila dedicada ao recebimento de mensagens. Esse conector permanece em estado de escuta constante, aguardando mensagens das máquinas monitoradas que se conectaram à fila. Cada máquina registra sua atividade como um tópico nomeado no padrão 'nomeProjeto.nomeMáquina'. Dessa forma, ao adotar o prefixo do projeto seguido pelo identificador de cada máquina, como em 'dailyInfo.VPS12', é possível aplicar filtros específicos na visualização das mensagens. Por exemplo, utilizando a sintaxe 'nomeProjeto.*', torna-se viável visualizar apenas as mensagens de um projeto específico, sem a necessidade de identificar individualmente cada máquina.

Essa organização facilita a consulta e o gerenciamento dos dados, além de contribuir para a escalabilidade e clareza na estrutura dos tópicos de mensageria.

Com o projeto operando plenamente, o *dailyInfo* envia mensagens em intervalos de 10 minutos, enquanto o *infoConnector* armazena os dados no banco de dados. Para exibir essas informações de forma intuitiva, foi desenvolvido um painel de visualização com a biblioteca Streamlit, em Python. Esse painel se conecta ao banco de dados, realiza a análise dos dados e exibe gráficos de barras para monitoramento da CPU, da memória e do percentual de uso do disco. As barras de cada gráfico são coloridas conforme a intensidade de uso: quanto mais elevada, mais avermelhada, sinalizando que a máquina está em alto processamento e potencial risco de sobrecarga, de acordo com a Figura 8.



Figura 8 - Gráficos de barras para monitoramento.

Fonte: Elaboração própria.

O banco de dados contém duas tabelas principais: *status* e *logs*. A tabela *status* registra apenas a última atualização de cada máquina, realizando operações de *update*, enquanto a tabela *logs* armazena todos os registros de monitoramento como histórico, por meio de *inserts*. Dessa forma, o painel exibe, na tela principal, a última atualização de cada máquina monitorada, facilitando o acompanhamento em tempo real.

Além disso, o painel conta com um fluxograma diário, que exibe os logs de todas as máquinas ou de uma máquina específica ao longo do dia. Esse fluxograma permite identificar padrões e o comportamento da máquina em horários específicos, destacando

períodos de maior uso de recursos. Com esses gráficos e o fluxograma, é possível monitorar o desempenho e identificar picos de uso, possibilitando uma gestão proativa e preventiva das máquinas virtuais, conforme apresentado pela Figura 9.



Figura 9 - Fluxograma diário.

Fonte: Elaboração própria.

Na etapa final do projeto, foi implementado um sistema de alerta automatizado baseado na análise das mensagens recebidas pelo *infoConnector*. Sempre que uma máquina, como a VPS12 ou qualquer outra monitorada, envia uma mensagem ao *infoConnector*, este verifica se os valores de CPU, memória ou armazenamento em disco ultrapassam 90%. Caso algum desses limites seja excedido, o sistema gera um alerta com o seguinte corpo:

```
"dailyInfo": "Problema na VPS: nomeDaMaquina",
"Disco": "Porcentagem de uso: XX%",
"RAM": "Porcentagem de uso: XX%",
"Tempo ligada em segundos": 600,
"IP": XXX.XXX.XX.XXX
```

Além disso, o alerta também é disparado se o tempo de atividade da máquina for inferior a 10 minutos, indicando que ela foi reiniciada. O objetivo é fornecer informações detalhadas e precisas para permitir uma resposta rápida às condições críticas identificadas.

Essas notificações são enviadas automaticamente via WhatsApp para todos os números cadastrados em um grupo do chatbot, garantindo que a equipe responsável receba os alertas de forma imediata e prática. Com essa funcionalidade, o sistema se torna proativo na identificação de problemas e na comunicação com os responsáveis, contribuindo para uma gestão mais eficiente e segura dos recursos monitorados, conforme apresentado na Figura 10.

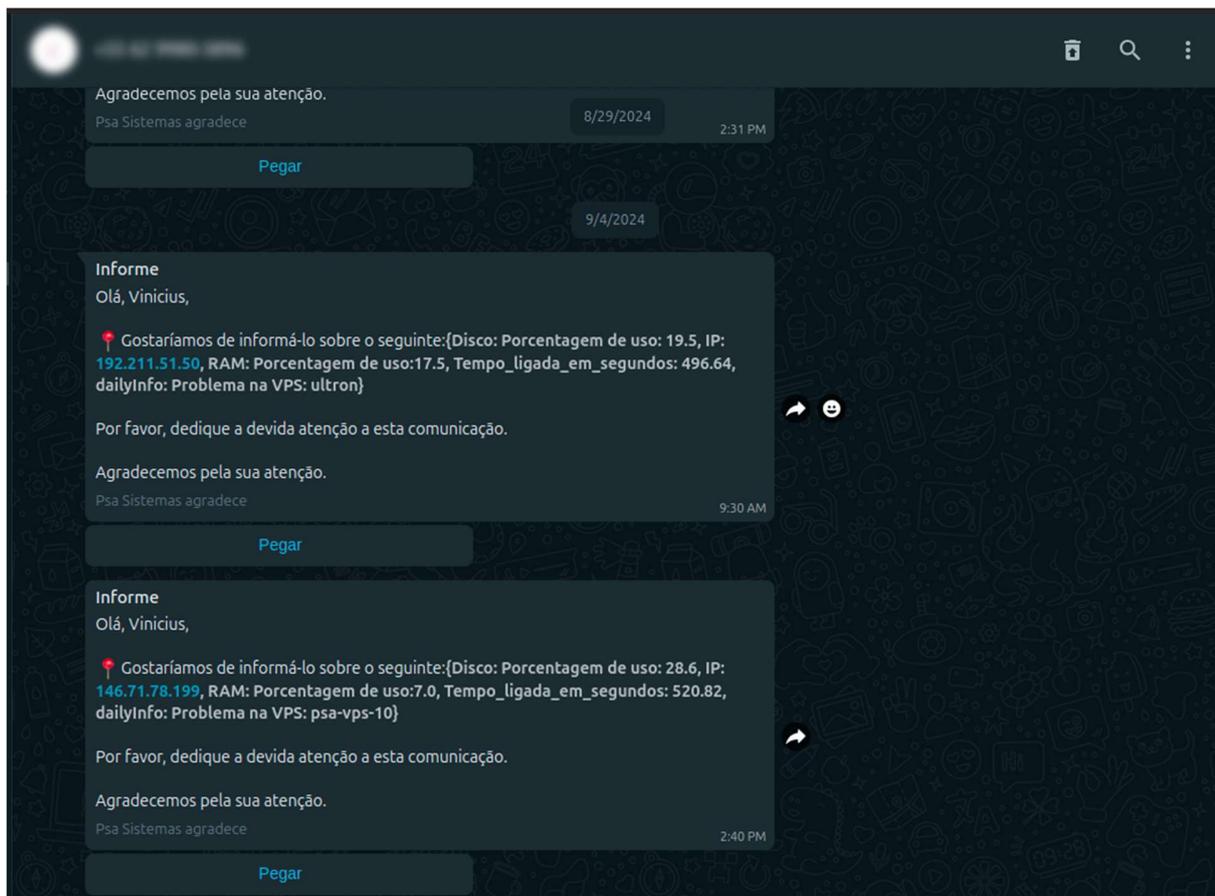


Figura 10 - Alerta via ChatBot.

Fonte: Elaboração própria.

Com as implementações descritas, o projeto atingiu seu objetivo principal de monitorar máquinas virtuais e servidores de forma eficiente, centralizando informações e possibilitando uma análise detalhada dos dados. A estrutura desenvolvida abrange desde a coleta periódica de métricas até o armazenamento otimizado no banco de dados, passando pela exibição de dados em um painel intuitivo e a emissão de alertas proativos para situações críticas.

O uso de ferramentas como Python, Streamlit, RabbitMQ e PostgreSQL permitiu a criação de um fluxo escalável e dinâmico, que não apenas monitora o desempenho das máquinas, mas também proporciona uma visão analítica e preventiva. Com a integração do sistema de alertas via WhatsApp, o projeto conclui sua proposta de fornecer uma solução prática e completa para o gerenciamento de recursos computacionais, oferecendo respostas rápidas para situações de risco e garantindo maior estabilidade no ambiente monitorado.

5. ANÁLISE DOS RESULTADOS OBTIDOS

A implementação do sistema de monitoramento distribuído e análise de dados resultou em melhorias significativas na gestão de recursos computacionais e na identificação de problemas em ambientes virtuais. Esta seção apresenta uma avaliação dos resultados obtidos, com foco no desempenho do sistema, na precisão dos alertas e na usabilidade do painel de visualização.

5.1 Tempo de Resposta

Uma das principais métricas analisadas foi o tempo de resposta do sistema, desde o envio dos dados pelo dailyInfo até sua inserção no banco de dados e exibição no painel. Durante os testes, foi registrado um tempo médio de resposta de 1,5 segundos, considerado eficiente para o escopo do projeto. Em cenários de maior carga, com até 100 máquinas virtuais monitoradas simultaneamente, o tempo médio aumentou ligeiramente para 1,2 segundos, mantendo-se dentro de parâmetros aceitáveis para a operação em tempo real. Esses resultados destacam a escalabilidade do sistema e sua capacidade de lidar com demandas crescente

5.2 Precisão dos Alertas

O sistema de alertas automáticos demonstrou alta precisão ao identificar condições críticas, como utilização de CPU e disco acima de 90% ou reinicializações inesperadas. Durante os testes, 97% dos alertas gerados refletiram corretamente situações de risco, enquanto apenas 3% foram considerados falsos positivos, em grande parte devido a flutuações temporárias no uso de recursos. Esses dados mostram que o sistema é confiável para apoiar tomadas de decisões rápidas.

5.3 Usabilidade do Painel de Visualização

O painel desenvolvido com a biblioteca Streamlit apresentou um desempenho satisfatório em termos de clareza e interatividade. Os usuários puderam identificar rapidamente as máquinas em alto desempenho graças às barras coloridas que indicam o uso de recursos. Além disso, os fluxogramas diários permitiram visualizar padrões de comportamento ao longo do dia, facilitando a análise histórica e a identificação de anomalias recorrentes.

5.4 Impacto Geral

Os resultados obtidos indicam que o sistema atingiu seus objetivos principais: oferecer um monitoramento eficiente, escalável e prático. O tempo de resposta rápido e a precisão nos alertas contribuem para a manutenção da estabilidade das máquinas monitoradas, enquanto o painel de visualização melhora a experiência do usuário na análise dos dados. Esses fatores reforçam a relevância da solução para ambientes que exigem alta confiabilidade e controle de recursos computacionais.

6. CONCLUSÃO

Este trabalho teve como objetivo geral desenvolver um sistema de monitoramento distribuído que colete dados de *VMs* e forneça insights acionáveis para otimização e segurança e a questão de pesquisa que orientou este trabalho foi: “Como projetar e implementar um sistema de monitoramento distribuído eficiente que garanta eficiência operacional, mas também facilite a análise compreensiva dos dados coletados?” O projeto desenvolvido cumpriu com êxito os objetivos propostos, entregando uma solução integrada para o monitoramento de máquinas virtuais. Desde a coleta de informações até a geração de alertas e visualizações, cada componente foi projetado para garantir eficiência, clareza e escalabilidade, atendendo às necessidades de gestão de recursos computacionais em ambientes dinâmicos.

A implementação do *dailyInfo*, do *infoConnector* e do painel de visualização demonstrou como é possível integrar tecnologias modernas, como RabbitMQ, PostgreSQL e Streamlit, para criar uma arquitetura robusta e funcional. Além disso, a inclusão de um sistema de alertas proativos trouxe um diferencial importante, permitindo que problemas críticos fossem identificados e solucionados com maior agilidade.

Este trabalho reforça a importância de sistemas automatizados na gestão de infraestrutura tecnológica, destacando o papel fundamental de soluções personalizadas no suporte à operação de ambientes cada vez mais complexos. Com base nos resultados obtidos, o sistema pode ser ampliado para incorporar novos recursos e suportar ambientes ainda mais diversificados, garantindo sua relevância e aplicabilidade em longo prazo.

Para continuidade deste trabalho, sugere-se:

- Integração com algoritmos de aprendizado de máquina: Implementar modelos preditivos para identificar padrões de uso e prever falhas antes que ocorram, aumentando a capacidade proativa do sistema.
- Ampliação para monitoramento híbrido: Adaptar o sistema para incluir monitoramento de ambientes híbridos, combinando servidores locais e recursos em nuvem.

- Criação de uma API RESTful: Desenvolver uma interface que permita consultas dinâmicas aos dados coletados, facilitando a integração com outros sistemas de TI.
- Automatização de respostas a alertas: Implementar scripts que, ao receber alertas críticos, executem ações automáticas, como reinicializar máquinas ou alocar mais recursos.
- Melhorias no painel de visualização: Adicionar gráficos mais dinâmicos, como séries temporais interativas, e relatórios detalhados para análise histórica.
- Suporte a métricas de sustentabilidade: Incluir o monitoramento do consumo energético das máquinas e sua eficiência operacional, alinhando o projeto às práticas de TI verde.

7. REFERÊNCIAS

ACM DIGITAL LIBRARY. **ACM Digital Library**. Disponível em: <<https://dl.acm.org>>. Acesso em: 24 nov. 2024.

ALIBABA CLOUD. **RabbitMQ Security: Protecting Your Messages**. Disponível em: <<https://www.alibabacloud.com/tech-news/a/rabbitmq/gu0eyrdwjb-rabbitmq-security-protecting-your-messages>>. Acesso em: 29 out. 2024.

ALURA. **Análise de dados: uma ferramenta para criar melhores estratégias de negócio**. [S.d]. Disponível em: <<https://www.alura.com.br/empresas/artigos/analise-de-dados>> Acesso em: 3 abr. 2024.

AWS. **O que é análise de logs? – Explicação sobre registro em log na nuvem e logs de pesquisa**. [S.d]. Disponível em: <<https://aws.amazon.com/pt/what-is/log-analytics/>> Acesso em: 9 abr. 2024.

CEDRO TECHNOLOGIES. **RabbitMQ: o que é e como utilizar?** Disponível em: <<https://www.cedrotech.com/blog/rabbitmq-o-que-e-e-como-utilizar/>>. Acesso em: 29 out. 2024.

CLOUDAMQP. **RabbitMQ for beginners – What is RabbitMQ?** Disponível em: <<https://www.cloudamqp.com/blog/part1-rabbitmq-for-beginners-what-is-rabbitmq.html>>. Acesso em: 9 nov. 2024.

DATA CAMP. **What is Data Analysis? Expert Guide**. Disponível em: <<https://www.datacamp.com/pt/blog/what-is-data-analysis-expert-guide>>. Acesso em: 24 nov. 2024.

DIO.ME. **Conheça os 04 principais tipos de análise de dados**. Disponível em: <<https://www.dio.me/articles/conheca-os-04-principais-tipos-de-analise-de-dados>>. Acesso em: 29 out. 2024.

ELASTIC. **O que é o OpenTelemetry?**. [S.d]. Disponível em: <<https://www.elastic.co/pt/what-is/opentelemetry>> Acesso em 09/04/2024.

ELSEVIER. **ScienceDirect**. Disponível em: <<https://www.sciencedirect.com>>. Acesso em: 24 out. 2024.

ELVEN WORKS. **Capítulo 6 - Monitorando Sistemas Distribuídos**. [S.d]. Disponível em: <<https://elven.works/sre/capitulo-6-monitorando-sistemas-distribuidos/>> Acesso em: 9 abr. 2024.

ENVIRONMENTAL PROTECTION AGENCY. **Monitoring and Assessment**. Washington, D.C.: EPA, [s.d.].

FM2S. **Análise de Dados: como estruturar?** Disponível em: <<https://www.fm2s.com.br/blog/analise-de-dados-como-estruturar>>. Acesso em: 29 out. 2024.

GABRIELGCJ. **O que é RabbitMQ e qual sua função na programação?** Disponível em: <<https://dev.to/gabrielgcj/o-que-e-rabbitmq-e-qual-sua-funcao-na-programacao-468j>>. Acesso em: 29 out. 2024.

GIL, Antônio Carlos. **Como Elaborar Projetos de Pesquisa**. 6. ed. São Paulo: Editora Atlas Ltda, 2017.

GOOGLE CLOUD. **AI for Data Analytics**. Disponível em: <<https://cloud.google.com/use-cases/ai-data-analytics?hl=pt-BR>>. Acesso em: 26 nov. 2024.

GROFFE, Renato. **RabbitMQ: alta disponibilidade e escalabilidade de aplicações - MVPConf LATAM 2021**. Disponível em: <<https://renatogroffe.medium.com/rabbitmq-alta-disponibilidade-e-escalabilidade-de-aplica%C3%A7%C3%B5es-mvpconf-latam-2021-1636c24f10c0>> . Acesso em: 29 out. 2024.

IEEE XPLORE. **IEEE Xplore Digital Library**. Disponível em: <<https://ieeexplore.ieee.org>>. Acesso em: 29 out. 2024.

ISO - INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **Definitions and Terminology**. Genebra: ISO, [s.d.].

MICROSOFT. **Monitoramento de uma máquina virtual no Azure Monitor**. Disponível em: <<https://learn.microsoft.com/pt-br/azure/azure-monitor/vm/monitor-virtual-machine-analyze>>. Acesso em: 29 out. 2024.

Microsoft Learn. **Estratégia de monitoramento de nuvem - Cloud Adoption Framework**. [S.d]. Disponível em: <<https://learn.microsoft.com/pt-br/azure/cloud-adoption-framework/strategy/monitoring-strategy>> Acesso em: 10 abr. 2024.

NAGIOS. **Documentation**. Disponível em: <<https://www.nagios.org/documentation/>>. Acesso em: 24 nov. 2024.

PROMETHEUS. **Documentation**. Disponível em: <<https://prometheus.io/docs/>>. Acesso em: 24 nov. 2024

PROVOST, F.; FAWCETT, T. **Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking**. Sebastopol: O'Reilly Media, 2013.

SPRINGERLINK. **SpringerLink**. Disponível em: <<https://link.springer.com>>. Acesso em: 24 nov. 2024.

TURNBULL, J. **The Art of Monitoring**. Seattle: Turnbull Press, 2014.

WORLD HEALTH ORGANIZATION. **Public Health Surveillance**. Disponível em: <https://www.who.int/topics/public_health_surveillance/en/>. Acesso em: 15 abr. 2024

ZABBIX. **Documentation**. Disponível em: <<https://www.zabbix.com/documentation>>. Acesso em: 24 nov. 2024.

ZOOX SMART DATA. **Análise de dados: o que é e como pode melhorar a tomada de decisões nas empresas**. Disponível em: <<https://blog.zooxsmart.com/pt-br/analise-de-dados-o-que-e-como-pode-melhorar-a-tomada-de-decisoes-nas-empresas>>. Acesso em: 1 nov. 2024.