UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS

PEDRO MIGUEL AGOSTINI CARRARO

MONITORAMENTO DE SERVIÇOS E ATIVOS DE TECNOLOGIA DE INFORMAÇÃO

CAXIAS DO SUL

2024

PEDRO MIGUEL AGOSTINI CARRARO

MONITORAMENTO DE SERVIÇOS E ATIVOS DE TECNOLOGIA DE INFORMAÇÃO

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Engenheiro de Computação na Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul.

Orientador: Prof. Dr. Daniel Luís Notari

CAXIAS DO SUL

2024

PEDRO MIGUEL AGOSTINI CARRARO

MONITORAMENTO DE SERVIÇOS E ATIVOS DE TECNOLOGIA DE INFORMAÇÃO

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Engenheiro de Computação na Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul.

Aprovado(a) em 27/11/2024

BANCA EXAMINADORA

_____

Prof. Dr. Daniel Luís Notari
Universidade de Caxias do Sul - UCS

_____

Prof. Me. Giovanni Ely Rocco
Universidade de Caxias do Sul - UCS

_____

Prof. Daniel Antonio Faccin
Universidade de Caxias do Sul - UCS

AGRADECIMENTOS

Com imensa gratidão, agradeço aos meus pais, que se dedicaram incansavelmente e ofereceram todo o suporte necessário para que eu pudesse alcançar esta importante conquista em minha vida.

Agradeço especialmente ao Professor Daniel Luis Notari, cuja orientação e expertise foram fundamentais para a realização deste trabalho, proporcionando-me o conhecimento e a confiança necessários para concluir este projeto.

Aos meus amigos e colegas, manifesto minha profunda gratidão pelo incentivo constante, pelas palavras de apoio e pela força que me deram nos momentos mais desafiadores desta jornada.

RESUMO

Este trabalho teve como objetivo identificar e implementar uma solução eficaz de monitoramento de serviços e ativos de Tecnologia da Informação (TI) para o grupo de empresas da Empresa X. Após a avaliação e teste de diversas plataformas, a *Statuspage*, da *Atlassian*, foi selecionada devido à sua capacidade de personalização, integração com os sistemas existentes e eficácia na comunicação de incidentes. A plataforma proporciona uma visão consolidada do estado atual e histórico dos serviços, facilitando auditorias de Acordos de Nível de Serviço (SLAs), notificações em tempo real e registros operacionais. Apesar da necessidade de integração com sistemas de monitoramento ativo para atualização de dados em tempo real, a *Statuspage* demonstrou alto desempenho na prática, contribuindo para a gestão de incidentes em sistemas críticos, monitoramento de qualidade de conexão e otimização da infraestrutura de rede. Assim, a solução mostrou-se eficiente na melhoria da observabilidade, redução de redundâncias e promoção de um gerenciamento mais proativo e eficiente do ambiente de TI.

Palavras-chave: Monitoramento. Sustentação de Serviços. Serviços. Ativos.

ABSTRACT

This study aimed to identify and implement an effective solution for monitoring IT services and assets for the Empresa X group of companies. After evaluating and testing several platforms, *Statuspage* by *Atlassian* was selected due to its customization capabilities, integration with existing systems, and efficiency in incident communication. The platform provides a consolidated view of the current state and history of services, facilitating SLA audits, real-time notifications, and operational records. Although it required integration with active monitoring systems for real-time data updates, *Statuspage* demonstrated high performance in practice, contributing to incident management in critical systems, connection quality monitoring, and infrastructure optimization. Thus, the solution proved to be efficient in improving observability, reducing redundancies, and promoting more proactive and efficient IT environment management.

Keywords: Monitoring. Service Support. Services. Assets.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AHP Processo Hierárquico Analítico
API Interface de Programação de Aplicações
AP Access Point
BD Bancos de Dados
CD Centros de Distribuição
CI/CD Integração Contínua e Entrega Contínua
CLP Controlador Lógico Programável
DC *Data Center*
DDoS *Distributed denial-of-service*
DNS Domain Name System
DR Recuperação de Desastre
EGR Estações de Gerenciamento de Rede
ER *Enterprise Replication*
ERP Planejamento de Recursos Empresariais
ES Entrada/Saída
GSTI Gerenciamento de Serviços de Tecnologia de Informação
ITIL *Information Technology Infrastructure Library*
JSON JavaScript Object Notation
k8s Kubernetes
LDAP Lightweight Directory Access Protocol
MIB Base de Informações de Gerenciamento
MES Sistema de Execução de Manufatura
OEE Eficiência Global de Equipamentos
OID Identificação do Objeto
PoC Prova de Conceito
RAA Rede de Área de Armazenamento
RCA Regras de Controle de Acesso
RLV Redes Locais Virtuais
SCADA Sistema de Supervisão e Aquisição de Dados
SGBD Serviço de Gerenciamento de Banco de Dados
SLAs Acordos de Nível de Serviço
SMS Serviço de Mensagem Curta

SNMP Protocolo Simples de Gerenciamento de Rede

SO Sistema Operacional

SSH Secure Shell

SSID Identificador de Conjunto de Serviços

TI Tecnologia da Informação

URL Localizador de Recursos Uniforme

VPN Rede Privada Virtual

WWW *World-Wide Web*

YAML *Yet Another Markup Language*

SUMÁRIO

1 INTRODUÇÃO

Um serviço na área de TI é, em essência, um componente projetado para sustentar um ou mais processos de negócios (WUNDER; HALBARDIER; WALTERMIRE, 2011). Estes serviços são fundamentais para o funcionamento eficiente e eficaz das organizações modernas, uma vez que suportam operações críticas e facilitam a realização de atividades empresariais. Desde a gestão de informações e comunicação interna até a automação de processos e suporte ao cliente, os serviços de TI abrangem uma ampla gama de funções vitais para a continuidade dos negócios. A gestão e governança eficazes desses serviços são cruciais para assegurar qualidade, segurança e resiliência, exigindo a aplicação de normas e padrões específicos. Sem uma gestão adequada, os serviços de TI podem falhar em atender às expectativas dos usuários e aos requisitos de negócios, resultando em interrupções, perda de dados e outros problemas que podem comprometer a operação e a competitividade da empresa (SERRANO *et al.*, 2021).

A implementação de boas práticas de governança e o uso de *frameworks* padronizados, como o *Information Technology Infrastructure Library* (ITIL), são essenciais para garantir que os serviços de TI estejam alinhados com os objetivos estratégicos da organização e sejam capazes de suportar suas demandas de forma confiável e eficiente. A gestão de serviços de TI envolve um conjunto de práticas e padrões voltados para a entrega e gerenciamento de serviços que atendam às necessidades do negócio. Um exemplo proeminente de tais padrões é o Gerenciamento de Serviços de Tecnologia de Informação (GSTI), um conjunto de definições com foco em alinhar os serviços de TI para garantir os objetivos do negócio (Atlassian, 2024). Estas normas definem o uso de ferramentas de gerenciamento e monitoramento em paralelo aos processos de TI, permitindo acompanhar e manter a configuração em ambiente produtivo, coletar métricas, identificar anomalias e implementar atualizações (JOHNSON *et al.*, 2007).

O principal serviço disponibilizado pela Empresa X é um sistema interno de Planejamento de Recursos Empresariais (ERP). Por definição, o ERP é uma solução abrangente de gestão composta por diversos softwares integrados, permitindo a comunicação e disponibilidade de dados para todas as áreas que compõem uma empresa, possibilitando maior visibilidade dos impactos e melhor tomada de decisão entre setores (KATUU, 2020). Sendo considerado o núcleo de uma organização, ele é responsável por acompanhar todos os processos empresariais e fabris em tempo real. Para isso, é composto por servidores de aplicação, que executam e processam o sistema, e Serviço de Gerenciamento de Banco de Dados (SGBD), que armazenam todas as informações da aplicação. Em sistemas menores, como lojas e centros de distribuição, ambos os softwares podem compartilhar o mesmo servidor e Sistema Operacional (SO).

Na infraestrutura de uma empresa, os equipamentos que disponibilizam a rede, capacidade de processamento ou espaço para armazenamento são considerados ativos, ou seja, itens inventariados que rendem valor agregado para a empresa (WUNDER; HALBARDIER; WALTER-

MIRE, 2011). Os ativos da Empresa X podem ser agrupados em três finalidades de uso principais: servidores utilizados para processamento e disponibilidade dos serviços e Bancos de Dados (BD), que podem ser máquinas físicas locais, virtuais em clusters locais ou processamento alocado em nuvem; switches gerenciados para distribuição da rede interna, definindo o roteamento de comunicações na topologia e implementação de Redes Locais Virtuais (RLV) para isolação de rede de servidores, equipamentos, computadores e telefones; e sistemas de armazenamento apresentados em uma Rede de Área de Armazenamento (RAA), que permitem armazenar, replicar e garantir alta disponibilidade de *backups*, arquivos sem estrutura, informações do BD e partições de inicialização dos servidores (MASICH *et al.*, 2022).

Grande parte do poder computacional é instalado em servidores de lâminas, com algumas aplicações isoladas em servidores de prateleira (*rack*). Servidores de lâminas utilizam uma estrutura modular para diminuir a ocupação de espaço físico e o consumo de energia. As lâminas (*blades*) são instaladas em um chassi (*enclosure*), que possui placas de rede, fontes de alimentação e portas de entrada e saída, deixando no módulo apenas os componentes de processamento e armazenamento, se necessário. Isso reduz a redundância de *hardware* e auxilia na organização de centros de processamento de alta densidade (THAKAR; BHATIA, 2022). A maioria das lâminas distribui seus recursos em clusters para virtualização dos servidores, permitindo uma grande quantidade de servidores de diferentes tamanhos e dinamismo na alocação de recursos entre diferentes aplicações.

Para gerenciar e controlar o tráfego de conexão e roteamento de pacotes, um *switch* de rede central é empregado. Este dispositivo atua como ponto de conexão entre o cabeamento horizontal e vertical dos data centers, integrando as operações das empresas regionais do grupo ou à internet. O *switch* é um equipamento de rede utilizado para a comunicação de outros equipamentos; caso seja um modelo gerenciado, também podem ser criadas RLV e suas interfaces para segmentação da rede, além de definições de Regras de Controle de Acesso (RCA) nas camadas 2, 3 ou 4 do Modelo OSI (PRIYADARSHI, 2024). Operando em paralelo ao *switch* central, há um servidor de *firewall* para realizar o controle e a moldagem das comunicações, internas e externas. O *firewall* é um *software* que pode possuir um servidor próprio, realizando a inspeção e filtragem de pacotes enviados em uma rede. Diferentes modelos e versões operam em diferentes camadas do Modelo OSI, aumentando a granularidade das regras e filtros (BRINGHENTI *et al.*, 2022; ANWAR; ABDULLAH; PASTORE, 2021).

1.1 PROBLEMA

Com enfoque no setor metalúrgico, a Empresa X possui fábricas, escritórios e centros de distribuição para atender diversas áreas do mercado. Ela atua predominantemente nos mercados de utensílios domésticos, jardinagem, eletrodomésticos, ferramentas manuais, entre outros. Para sustentar todo o processo de produção, distribuição e venda, utiliza diversas ferramentas agrupadas em um sistema interno. Com grande parte dos sistemas sendo locais, a disponibi-

lidade dos serviços depende da infraestrutura interna, que é composta por data centers com servidores, bancos de dados, *switches* e *storages*, entre outros equipamentos com resiliência, alta disponibilidade e redundância.

A transmissão de informação desempenha um papel fundamental como o eixo central que permite a colaboração de diferentes áreas, visando um objetivo comum. Na estrutura corporativa, isso se manifesta entre os diferentes departamentos que precisam se ajudar para a entrega de projetos ou comercialização de produtos. Existem várias maneiras de disponibilizar informações sobre os serviços de TI, como canais de comunicação, alertas automatizados, avisos por *e-mail* ou página de *status*.

Na Empresa X, as informações não são devidamente distribuídas e muitas informações incorretas são disseminadas. Transferir o problema de um setor para outro é comum, causando atraso na resolução de problemas, falta de cooperação entre departamentos e prejuízos à marca e ao produto. Isso ocorre pela falta de clareza e fácil acesso à situação dos serviços e ativos do departamento de tecnologia.

Com diversos ativos disponibilizados, cada um relacionado a um setor ou departamento, é importante definir em quais processos pode estar ocorrendo instabilidade e como isso pode afetar outros serviços. É necessário garantir que essas informações sejam protegidas para evitar contato direto com as equipes, o que pode causar transtorno e não auxiliar na resolução dos problemas.

As equipes contam com monitoramento e alarmes técnicos sobre os serviços, para auxiliar no acompanhamento, implantação e administração. Essas ferramentas possuem informações privadas que não podem ser expostas publicamente. Dessa forma, é necessária uma camada de processamento dessas informações para anonimização e simplificação dos dados, facilitando a comunicação e refletindo a experiência do usuário.

Por exemplo, durante uma atualização de rotina de um banco de dados ocorrida automaticamente pela manhã, um erro inesperado deixou alguns serviços indisponíveis. Os administradores de banco de dados foram avisados do erro e iniciaram a reparação da plataforma, mas nenhum cliente foi notificado. Para eles, o serviço estava fora do ar e começaram a enviar mensagens para a equipe de infraestrutura, em grupos gerais e para outras pessoas não relacionadas, causando confusão e atrapalhando os administradores que estavam normalizando a situação.

De forma similar, uma aplicação apresentou anomalias e encerramentos repentinos durante o turno da manhã. Sem outra opção, o cliente contatou os responsáveis pelo software questionando se algo poderia ter ocorrido. A equipe repassou para os administradores de infraestrutura e redes para validar o estado dos servidores e das conexões, que, após investigação, não identificaram anomalias no ambiente. Horas depois, descobriu-se que uma atualização realizada na mesma manhã possuía um caso de uso não mapeado que causava o problema. Pela falta de uma plataforma ou *site* para verificar o estado dos ativos, a solução foi atrasada em

algumas horas, o que em um ambiente de produção representa possível perda de faturamento e atraso no tempo de entrega.

1.2 OBJETIVOS

Testar e selecionar uma plataforma de monitoramento de ativos de TI para apoiar a tomada de decisão dos gestores da Empresa X.

Os objetivos específicos são:

1. Caracterizar os serviços de ativos e monitoramento de TI.
2. Descrever a infraestrutura de sistemas da Empresa X.
3. Descrever ferramentas de serviços de ativos e monitoramento de TI.
4. Avaliar ferramentas de serviços de ativos e monitoramento de TI.
5. Instalar, testar e avaliar uma ferramenta de serviços de ativos e monitoramento de TI.

1.3 ESTRUTURA DO TRABALHO

O trabalho está estruturado da seguinte maneira:

- O Capítulo 2 apresenta o referencial teórico, abordando os conceitos e ferramentas utilizados ao longo deste estudo.
- O Capítulo 3 descreve a aplicação dos conceitos pela empresa X e seu grupo, detalhando as aplicações, organização, estruturas e topologia.
- O Capítulo 4 discute os softwares de monitoramento de serviços e ativos, suas características e configurações.
- O Capítulo 5 detalha a escolha e os testes das soluções para determinar a que melhor atende aos requisitos, além de definir suas implementações e treinamentos.
- O Capítulo 6 apresenta o processo de contato com os fornecedores das soluções de monitoramento, a realização da prova de conceito para validação prática e as etapas envolvidas em sua implementação na Empresa X.

2 TOPOLOGIA DOS SERVIÇOS E ATIVOS DE TI

A topologia dos serviços e ativos de TI descreve a estrutura e o relacionamento entre os diversos serviços e ativos de TI que sustentam as operações empresariais. Neste capítulo, exploraremos os conceitos e as práticas relacionadas à topologia dos serviços e ativos de TI, destacando a importância dessa visão holística para a excelência operacional e estratégica das organizações modernas. A Seção 3.1 aborda o sistema de ERP desenvolvido e utilizado internamente pelas empresas do grupo, analisando a dependência dele com a infraestrutura disponibilizada, incluído o sistema disponibilizado parar acesso externo, como a Seção 3.1.1, Portal Interno para os colaboradores e parceiros.

Na Seção 3.2 é descrito o sistema de gestão e controle utilizado nas empresas fabris do grupo. Os ativos são definidos na Seção 3.3, Seção 2.4 e Seção 2.5, tratando de descrever e definir os equipamentos utilizados para processamento, roteamento, comunicação e armazenamento. Seu monitoramento é realizado em um servidor central configurado com o *software Zabbix*, como descrito na Seção 3.4.

2.1 SISTEMA DE PLANEJAMENTO DE RECURSOS EMPRESARIAIS

O ERP é uma plataforma de software que integra e automatiza as principais funções organizacionais, abrangendo áreas como finanças, vendas, recursos humanos, compras, produção e distribuição. Esta integração fornece uma visão consolidada e consistente dos processos de negócios, facilitando a tomada de decisões estratégicas e melhorando a eficiência das operações internas e externas (HAYEK; ODEH, 2020). A implementação de um sistema ERP é um projeto de grande escala que exige um compromisso significativo de tempo e recursos financeiros, devido à complexidade de integrar múltiplos sistemas de informação em uma única plataforma integrada (AL-JABRI; ROZTOCKI, 2015).

Um sistema ERP é projetado para centralizar as informações de diferentes departamentos de uma organização, promovendo uma maior eficiência e reduzindo a redundância de dados. A integração proporcionada por um ERP permite que as empresas respondam mais rapidamente às mudanças do mercado e aprimorem sua capacidade de prever demandas futuras. Além disso, a utilização de um ERP pode resultar em uma melhor coordenação entre as áreas de negócio, promovendo uma comunicação mais eficaz e um alinhamento estratégico entre as diversas unidades da empresa. No entanto, a implementação de um ERP não é uma tarefa trivial; envolve a reengenharia de processos de negócio, a formação dos usuários e a adaptação cultural da organização ao novo sistema (DAVENPORT, 1998).

Além disso, muitas implementações de ERP não alcançam o sucesso esperado por falta de planejamento adequado ou recursos insuficientes. Esses fracassos frequentemente ocorrem

quando o planejamento não se baseia em uma análise detalhada das necessidades organizacionais e não leva em consideração a adequação do sistema às características específicas da empresa. Portanto, para assegurar o sucesso de uma implementação de ERP, é crucial realizar uma análise minuciosa das necessidades da organização e desenvolver um plano estratégico bem elaborado que considere os recursos disponíveis e os objetivos de longo prazo da empresa.

A falta de sucesso em projetos de implementação de ERP pode ser atribuída a diversos fatores, incluindo a resistência à mudança por parte dos funcionários, a subestimação do tempo necessário para a implementação e a falta de treinamento adequado para os usuários finais (SOMERS; NELSON, 2001). A gestão de mudança organizacional desempenha um papel fundamental na adoção de um sistema ERP, pois envolve preparar e motivar os funcionários para aceitarem as novas tecnologias e processos. Além disso, é essencial garantir que haja um suporte contínuo e uma avaliação constante do desempenho do ERP após sua implementação, para identificar e corrigir possíveis problemas (WEI; WANG, 2004). Dessa forma, a implementação de um ERP pode trazer significativos benefícios para a organização, desde que seja conduzida com planejamento e execução rigorosos.

2.1.1 PORTAL INTERNO E PARCEIROS

Para aprimorar a colaboração e a produtividade, uma *intranet*, denominada Portal Interno, é empregada para facilitar o acesso remoto aos sistemas corporativos por parte dos colaboradores. Este portal é configurado como um *site* acessível via *internet*, permitindo acesso autenticado dos funcionários. O ambiente virtual inclui uma gama diversificada de ferramentas e recursos, que vão desde aplicações comuns, como atalhos para o *e-mail* corporativo e *websites* de empresas parceiras, até soluções especializadas para representantes que atuam em ambientes externos ou remotos (SULAIMAN; ZAILANI; RAMAYAH, 2012). Além disso, é possível integrar aplicações que não são compatíveis com o agente padrão da aplicação ou que foram desenvolvidas para operar em múltiplas plataformas. Essas aplicações podem ser adaptadas para funcionamento via navegador, possibilitando o acesso através de dispositivos móveis, como *tablets* e *smartphones*, diretamente do Portal Interno.

A implantação de um Portal Interno eficiente pode promover uma comunicação mais eficaz entre os colaboradores, facilitando o compartilhamento de informações e a colaboração em projetos. Ferramentas como fóruns de discussão, *wikis* corporativos e bibliotecas de documentos são frequentemente integradas ao portal, proporcionando um ambiente colaborativo robusto (MCAFEE, 2006). Além disso, a acessibilidade de aplicações críticas através de dispositivos móveis garante que os colaboradores possam manter-se produtivos mesmo fora do ambiente de escritório, respondendo rapidamente a demandas urgentes e mantendo o fluxo de trabalho contínuo (LEE; KIM, 2009). A segurança dos dados transmitidos e armazenados no Portal Interno também é um aspecto crucial, e medidas como autenticação multifator e criptografia de dados são implementadas para proteger informações sensíveis (BIDGOLI, 2010).

No contexto de sistemas distribuídos, como é o caso da infraestrutura descrita, a utilização de uma arquitetura baseada em microsserviços pode ser considerada para aumentar a resiliência e a escalabilidade do portal. Esse modelo permite que componentes individuais do sistema sejam atualizados, testados e escalados independentemente dos demais, facilitando a manutenção e a evolução contínua do sistema sem interrupções significativas do serviço. A abordagem modular dos microsserviços é crucial, especialmente em organizações que dependem de alta disponibilidade e desempenho constante de seus sistemas informatizados (NEWMAN, 2021). A arquitetura de microsserviços oferece a flexibilidade necessária para implementar melhorias e corrigir problemas de forma isolada, minimizando o impacto nas operações diárias e garantindo que o portal continue a atender às necessidades dos colaboradores de maneira eficiente e confiável.

A implementação de microsserviços permite que a organização responda rapidamente a mudanças nas demandas de negócios, escalando componentes específicos conforme necessário, sem a necessidade de redimensionar todo o sistema (DRAGONI *et al.*, 2017). Essa abordagem também facilita a integração de novas tecnologias e ferramentas, proporcionando uma base flexível e adaptável para a evolução contínua do portal. Além disso, a arquitetura de microsserviços pode melhorar a resiliência do sistema, permitindo a recuperação rápida de falhas em componentes individuais sem impactar significativamente o serviço geral (FOWLER; LEWIS, 2014). A combinação dessas vantagens torna os microsserviços uma escolha estratégica para a gestão de sistemas corporativos complexos e distribuídos.

2.2 SISTEMA DE SUPERVISÃO E AQUISIÇÃO DE DADOS

O Sistema de Supervisão e Aquisição de Dados (SCADA) é um sistema responsável por controlar e registrar as atividades das máquinas e das linhas de produção, integrando dados provenientes do ERP, além de coletar e distribuir dados para o sistema analítico (YADAV; PAUL, 2021). O sistema é composto por uma aplicação *web* que opera em paralelo com o Controlador Lógico Programável (CLP) das máquinas, responsável pelo gerenciamento e controle local, regulando todos os parâmetros e equipamentos (IOANNIDES, 2004). Nesta aplicação, realiza-se a autenticação dos operadores, verificando-se se suas permissões são adequadas para as máquinas em operação. As ordens de produção atribuídas são listadas, permitindo-se o início das mesmas. É possível também acessar informações técnicas sobre as peças, ajustes necessários e documentos relacionados, como manuais e certificados. Ao iniciar uma produção, o horário é registrado, juntamente com eventuais paradas para manutenção, troca de suprimentos ou intervalos, e o horário de conclusão. Além disso, apresenta um indicador em tempo real do Eficiência Global de Equipamentos (OEE) da máquina e outras métricas relevantes para o acompanhamento da produção. O OEE é um indicador de performance e eficiência de um equipamento, normalmente utilizado para acompanhar a produção e os processos manufaturados (HEDMAN; SUBRAMANIYAN; ALMSTRÖM, 2016).

A integração do SCADA com o ERP possibilita uma comunicação fluida entre os sistemas de gestão e a linha de produção, assegurando que todos os dados relevantes sejam compartilhados de maneira eficiente. Esse compartilhamento de informações permite uma visão holística da operação, facilitando a identificação de gargalos e a tomada de decisões informadas. A implementação do SCADA também envolve o uso de tecnologias de comunicação em tempo real, como protocolos de comunicação industrial, que garantem a transmissão rápida e precisa dos dados coletados pelas máquinas (GALLOWAY; HANCKE, 2012). Essas tecnologias são essenciais para manter a sincronização entre o sistema de controle e o sistema de gestão, promovendo uma operação mais coesa e eficiente.

Um BD pode ser compreendido, conforme definido por (ELMASRI *et al.*, 2020), como um conjunto de dados correlacionados, definidos ou não por uma mesma estrutura. Em maiores detalhes, os BD de CLP são projetados para registrar os valores obtidos das máquinas em operação, possibilitando a análise e cálculo de sua eficiência. Um exemplo disso é o indicador de OEE, que avalia interrupções na produção ou possíveis defeitos. Por outro lado, o BD SCADA é responsável por armazenar metadados do processo, que são exibidos no terminal para os clientes, além de conter informações sobre autenticações e todas as conexões necessárias. Para fins de acompanhamento e monitoramento, utiliza-se um *pod* denominado MES_Screens. Este *pod* emprega os dados coletados e processados no BD MES para publicar e manter páginas na *intranet*, que são exibidas em TVs e telas de monitoramento. Esta estrutura facilita a visualização em tempo real das operações e contribui para a gestão eficiente dos processos produtivos.

A arquitetura dos BD associados ao SCADA e ao CLP permite uma gestão robusta dos dados de produção, garantindo que informações críticas estejam disponíveis para análise e decisão. A utilização de *pods* como o MES_Screens é fundamental para a disseminação de dados em tempo real, permitindo que os gestores acompanhem o desempenho das operações de forma contínua (MCCLUSKY; PELTON, 2023). Além disso, a integração desses sistemas com a *intranet* da empresa facilita o acesso às informações por parte de todos os níveis hierárquicos, promovendo uma cultura de transparência e eficiência operacional. A capacidade de registrar e monitorar todos os aspectos da produção em tempo real ajuda a identificar rapidamente quaisquer desvios ou problemas, permitindo ações corretivas imediatas e a manutenção de um alto nível de desempenho.

2.3 SERVIDOR

Um servidor é um sistema de computação dedicado a fornecer serviços, recursos ou dados a outros computadores, conhecidos como clientes, em uma rede. Servidores podem variar amplamente em termos de tamanho e capacidade, desde sistemas simples para pequenas empresas até enormes data centers que atendem grandes organizações. A função principal de um servidor é responder a solicitações feitas por clientes, fornecendo acesso a recursos compartilhados, como arquivos, aplicações, bancos de dados e serviços de internet (SERVERWATCH,

2023; TECHOPEDIA, 2023).

Os servidores de lâminas, em conjunto com sua capacidade de otimizar espaço e eficiência energética, são uma solução ideal para ambientes que exigem alta densidade computacional com uma pegada mínima. O chassi não só fornece a infraestrutura necessária, como alimentação, refrigeração e conexões de rede, mas também permite que múltiplas lâminas compartilhem esses recursos críticos, maximizando a eficiência e reduzindo o custo total de operação. Essa estrutura é ideal para a elaboração de um *cluster* de computação agregada, que é a união dos recursos computacionais, de armazenamento e de conexão de redes (DATABANK, 2023). Nesse ambiente, os servidores são virtualizados, adicionando uma camada de refinamento nas suas configurações, e com chassis replicados entre DC, é possível mover os servidores logicamente entre cidades com basicamente nenhuma perda de conexão.

A capacidade de otimizar espaço e eficiência energética é notável nos servidores de lâminas, representando uma solução ideal para ambientes que demandam alta densidade computacional com mínimo impacto espacial. O chassi desses servidores não apenas provê a infraestrutura essencial, como alimentação, refrigeração e conexões de rede, mas também facilita o compartilhamento desses recursos críticos entre várias lâminas. Este compartilhamento maximiza a eficiência operacional e reduz os custos totais. Essa estrutura é ideal para a criação de um *cluster* de computação agregada, que integra recursos computacionais, de armazenamento e de rede em um ambiente virtualizado, permitindo refinamentos nas configurações dos servidores (VMware Inc., 2023). Além disso, com o chassi sendo implementado entre DC, é possível realizar a migração lógica de servidores entre diferentes localidades com mínima ou nenhuma interrupção na conectividade.

Com o objetivo de criar um ambiente de execução consistente e eficiente para aplicações, os contêineres representam uma abordagem em que a virtualização é desagregada em camadas, delegando ao servidor hospedeiro a responsabilidade pelo hardware e pelo *kernel*, enquanto uma imagem pré-configurada é disponibilizada para execução. Essa imagem é composta por um SO e camadas de configurações, que podem incluir desde modificações em arquivos até instalação de software. O processo de geração dessas imagens pode ser automatizado para facilitar atualizações e ajustes. Uma das principais vantagens dos contêineres é a velocidade de inicialização: um servidor físico pode demorar de dez a trinta minutos para ligar, uma máquina virtual pode levar de dois a cinco minutos para subir, enquanto um contêiner pode levar apenas segundos. Por conta de utilizar imagens prontas, o processo de inicialização se resume na criação de uma camada de interação temporária com o sistema, e, quando finalizado, a imagem original permanece inalterada (DOCKER, 2024).

2.3.1 CLUSTER

A virtualização, em conjunto a um ambiente de recursos compartilhados, não só proporciona uma maior granularidade no controle de recursos, mas também expande a eficiência

operacional de *clusters* de processamento. Estes *clusters* são essenciais na manutenção de ativos e recurso de alta disponibilidade, hospedando mais de 80% das aplicações e sistemas da Empresa X, oferecendo dinamismo para manutenção e gerência dos servidores (ALI *et al.*, 2015). A estrutura de um *cluster* de processamento permite não apenas a configuração de múltiplos servidores virtuais em um número limitado de máquinas físicas, mas também facilita a realocação dinâmica de recursos e a transferência de cargas de trabalho entre DC sem interrupções no serviço. Por não estar restrito ao *hardware* os servidores podem ser realocados em diferentes partes do *clusters*, incluindo partições em diferentes DC. Isso permite, em casos de desastres, recursos seja remanejado para os parques disponíveis, sendo essencial para aplicações de missão crítica ou de alta disponibilidade (STEINDER; WHALLEY; CHESS, 2008).

O uso de servidores dedicado (*baremetal*) está em declínio, pelo custo e restrições do uso de *hardware* isolado, sem camadas de virtualização. Acompanhando o movimento de infraestrutura como serviço, mesmo que disponíveis, o uso desse tipo de máquinas ficou isolado para aplicações críticas que podem sofrer por ruido do consumo de recursos de máquinas virtuais que dividem o mesmo *cluster*. A adoção da computação em nuvem permite que as organizações se beneficiem de uma escala mais flexível e de uma gestão de recursos mais eficiente. Este movimento em direção à nuvem reflete uma tendência crescente de otimização de infraestrutura e de redução de custos operacionais, ao mesmo tempo em que se maximiza o retorno sobre o investimento em tecnologia (RABETSKI; SCHNEIDER, 2013).

2.3.2 KUBERNETES

O principal uso de *clusters* de virtualização é a criação de máquinas virtuais, que tendem a serem mais persistentes, contudo, para os casos de serviços que necessitam que podem ser desligados e ligas dependendo da carga de uso, é possível utilizar um *cluster* de contêineres (ANDERSON, 2015). O Kubernetes (k8s) é uma solução para gerência de *pods*, tarefas (*jobs*) e contêineres, oferendo automações para balanço de carga, agendamento de tarefas, redirecionamento de conexões, entre outras funções. Esse ambiente de execução é composto por nós de gerência (*control plane*) para controle e de trabalhadores (*workers*) para recursos computacional e armazenamento (Kubernetes, 2024).

Os nós de gerência são responsáveis pela orquestração do tarefas e serviços em execução, como também por gerência dos trabalhadores. Nele é configurado e adicionado novos trabalhadores, gerenciado credenciais e variáveis, podendo ser executado *cluster* para interface gráfica do serviço (KUBERNETES, 2019; BURNS *et al.*, 2022). Cada nó deve ser configurado e instalado em um servidor próprio, sendo adicionado ao conjunto de servidores, mas não é hospedado nenhum contêiner de serviço neles, esses servidores não precisam de nenhuma característica ou condição especial e podem ser substituídos em caso de falha. Sendo o caso ideal para serem hospedados em um ambiente de virtualização, que combinado com *snapshots* e reinício automático, permite resiliência para incidentes, considerando que a seguinte regra está de

acordo, sempre possuir um número ímpar maior que um de coordenadores no ambiente, para assim considerando o número máximo de falhas não ultrapasse 50% dos servidores (Docker,).

Os nós trabalhadores são essenciais para gerenciar os recursos do ambiente, hospedando os contêineres e *pods* que podem ser realocados entre diferentes nós. Estes servidores são considerados não críticos; portanto, em caso de falhas, seus serviços são rapidamente reiniciados em outros trabalhadores (BURNS *et al.*, 2022). Além disso, volumes persistentes presentes em uma máquina que falha podem estar armazenados em um servidor de armazenamento externo, permitindo sua recuperação em outro nó. Como é crucial manter um ambiente virtualizado, é ideal que cada nó tenha seu próprio *hardware* e servidor dedicados, sem depender de uma camada adicional de virtualização.

A unidade fundamental gerenciada pelo ambiente do k8s é o *pod*, que representa um grupo de contêineres operando em conjunto. Dentro de um mesmo *pod*, recursos como armazenamento e rede são compartilhados entre os contêineres, facilitando a comunicação e a eficiência operacional. Além disso, um *pod* permite a definição de especificações detalhadas para a configuração desses serviços, otimizando a gestão de dependências e a configuração de ambiente (PHAN; KIM *et al.*, 2022). Para a criação e inicialização automatizada de um *pod*, utilizam-se arquivos de configuração escritos em *Yet Another Markup Language* (YAML), que facilita tanto o desenvolvimento quanto a documentação dos processos. Esses arquivos permitem aos desenvolvedores e administradores de sistemas descrever de maneira precisa a infraestrutura necessária, promovendo um desdobramento ágil e eficiente das aplicações dentro do *cluster* do k8s (NGUYEN *et al.*, 2020).

As aplicações são inicializadas a partir de imagens criadas durante o processo de *pipeline*, um componente central da prática de Integração Contínua e Entrega Contínua (CI/CD). Esta abordagem é projetada para não apenas entregar aplicações de forma eficiente aos usuários, mas também para garantir sua manutenção contínua e aprimoramento. Após a fase de desenvolvimento, a aplicação é automaticamente montada e compilada; subsequentemente, uma imagem contendo a versão mais recente do ambiente é gerada. Esta imagem é então destinada a substituir a versão anterior em produção (CHANDRAMOULI, 2022). Em cenários específicos, é possível implementar essa atualização sem interrupções. Isso é feito iniciando a nova versão da aplicação em paralelo à antiga. Uma vez que a nova versão esteja operando corretamente, as versões antigas são gradativamente desativadas. Esse método assemelha-se ao processo de balanceamento de carga, onde o tráfego é cuidadosamente deslocado da versão antiga para a nova sem afetar a disponibilidade do serviço. Essa estratégia de atualização sem tempo de parada é crucial para sistemas críticos, onde a continuidade do serviço é uma prioridade.

Para serviços acessados por múltiplos usuários, a questão das conexões simultâneas é crucial, pois pode gerar lentidão ou até mesmo indisponibilidade do serviço. Essa vulnerabilidade é frequentemente explorada por ataques do tipo *Distributed denial-of-service* (DDoS), que objetivam tornar um serviço indisponível ao saturar seu limite de tráfego ou capacidade

de processamento (MIRKOVIC; REIHER, 2004). A realocação de recursos conforme a demanda pode mitigar esse problema; contudo, os recursos de um único nó trabalhador eventualmente atingem seu limite, introduzindo a necessidade do balanceamento de carga. O balanceamento é efetuado através de recursos como *Service* e *Ingress*, que distribuem o tráfego de rede e as solicitações aos aplicativos entre múltiplos *pods*. O *Service* facilita o acesso aos *pods* por meio de *IPs* internos ou externos, permitindo que o serviço seja exposto em cada nó do *cluster*.

O *Ingress*, por sua vez, administra o acesso externo ao direcionar o tráfego para os serviços apropriados dentro do *cluster*, baseando-se em critérios como *URLs* e caminhos. Esses mecanismos não apenas otimizam a utilização dos recursos, distribuindo as cargas de forma equilibrada, mas também aumentam a disponibilidade e a resiliência dos aplicativos. Eles possibilitam a rápida redistribuição do tráfego para outros *pods* saudáveis em caso de falhas, garantindo a continuidade dos serviços sem interrupções perceptíveis aos usuários.

2.4 SWITCH

Para a comunicação eficaz entre todos os equipamentos, computadores e servidores de uma empresa, adota-se uma hierarquia de *switches* de rede, começando com um *switch* principal destinado aos gestores. O *switch* de rede é um dispositivo capaz de receber e enviar pacotes entre diversos aparelhos, incluindo computadores, servidores, roteadores e outros *switches* (Cloudflare,). Existem diversos modelos de *switches*, cada um projetado para atender demandas específicas. Um aspecto crucial desses dispositivos é determinar em qual camada do modelo OSI eles operam: os *switches* de camada 2, Enlace, apenas recebem e transmitem pacotes, sem a capacidade de roteamento; já os de camada 3, Rede podem rotear as conexões (ManageEngine,).

Os *switches* podem ser simples extensores de portas de rede, como os utilizados em ambientes domésticos, ou dispositivos altamente customizáveis para grandes infraestruturas empresariais com múltiplas redes distintas. De modo geral, esses equipamentos são classificados em dois tipos: modulares e fixos. Os *switches* modulares, embora mais caros, oferecem vantagens significativas em termos de expansibilidade e personalização, permitindo, por exemplo, a adição de fontes ou ventoinhas extras para redundância em caso de falha, além da escolha da quantidade, tipo e velocidade das portas. Por outro lado, os *switches* fixos são mais econômicos, porém limitam-se a configurações padrão (Cisco, a).

Para aplicações empresariais, é ideal que todos ou os principais *switches* sejam gerenciáveis, o que implica que possuem interfaces específicas para configuração. Essas interfaces podem ser baseadas em linha de comando, acessíveis diretamente através de uma conexão serial no dispositivo ou remotamente via configuração de rede. Alguns modelos também oferecem a opção de interfaces *web*, facilitando a gestão (Cisco, a). Predominantemente, esses equipamentos operam com um sistema operacional próprio, que inclui parâmetros de inicialização e um arquivo de configuração específico do *switch*. Este arquivo funciona de maneira similar a um

shell script, contendo todos os comandos necessários para a configuração das portas, definição das RLV, entre outros aspectos críticos do sistema (HAN *et al.*, 2015).

Existem diversas metodologias para monitorar e gerenciar um *switch* ou outro dispositivo de rede. A abordagem varia significativamente entre diferentes fabricantes e, para ambientes e sistemas heterogêneos, padrões e protocolos foram desenvolvidos para mitigar essas discrepâncias. Um exemplo notável é o Protocolo Simples de Gerenciamento de Rede (SNMP), que permite a coleta e configuração de informações a partir de uma ferramenta externa de monitoramento (MAURO; SCHMIDT, 2005). Este protocolo é crucial para garantir a interoperabilidade e eficiência na gestão de redes complexas.

O SNMP funciona como uma plataforma essencial para a administração de dispositivos em uma rede de computadores. Implementado sem a necessidade de um agente de software, o SNMP é compatível com uma variedade de sistemas operacionais e também com sistemas embarcados, frequentemente encontrado já integrado em equipamentos industriais de alto padrão. Este agente tem a função primordial de monitorar o estado do dispositivo em que está instalado, coletando dados operacionais que são armazenados em um banco de dados local, conhecido como Base de Informações de Gerenciamento (MIB) (CASE *et al.*, 1990).

Cada peça de informação armazenada no MIB, chamada de objeto, é unicamente identificada por um Identificação do Objeto (OID). Esses OID permitem que os administradores de rede acessem dados específicos de maneira eficiente e segura (CASE *et al.*, 1989). A gestão desses objetos é facilitada por uma Estação de Gerenciamento de Rede (Estações de Gerenciamento de Rede (EGR)), uma plataforma que se conecta ao agente SNMP para recuperar, configurar e controlar as informações de maneira centralizada.

A comunicação entre a EGR e o agente SNMP ocorre através do protocolo UDP. Dentro do modelo OSI, o SNMP se situa na camada de Aplicação, a sétima camada, que facilita a interação direta com o software de aplicação do usuário final. As operações fundamentais que o SNMP suporta incluem comandos *GET* para solicitar dados de um agente, e *SET* para modificar o valor de um objeto no MIB, permitindo aos administradores ajustar a configuração dos dispositivos conforme necessário. Esta arquitetura permite que redes complexas sejam gerenciadas de forma eficiente, fornecendo aos administradores as ferramentas necessárias para manter o desempenho ótimo, a segurança e a estabilidade da infraestrutura de TI.

2.5 STORAGE

O armazenamento de dados é um componente crítico em qualquer infraestrutura de TI, pois ele não só guarda informações valiosas, mas também assegura a continuidade e eficiência dos serviços. No contexto de sistemas de monitoramento de serviços e ativos de TI, os sistemas de armazenamento desempenham um papel essencial ao fornecer a capacidade de armazenar, gerenciar e proteger grandes volumes de dados.

O termo *storage* refere-se ao conjunto de tecnologias e dispositivos utilizados para armazenar dados digitalmente. Essas tecnologias variam desde discos rígidos (HDD) até unidades de estado sólido (SSD), e sistemas mais complexos como RAA (*Storage Area Networks* (SANs)). O principal objetivo de um sistema de *storage* é garantir que os dados estejam disponíveis, seguros e acessíveis quando necessários.

Uma RAA é uma arquitetura de rede projetada para conectar servidores a dispositivos de armazenamento de forma que os dados possam ser transferidos de maneira eficiente e com baixa latência. As RAA são frequentemente utilizadas em ambientes empresariais onde há uma necessidade crítica de desempenho, escalabilidade e resiliência. As RAAs utilizam tecnologias como *Fibre Channel* e *iSCSI* para estabelecer conexões rápidas e confiáveis entre servidores e dispositivos de armazenamento (WETHERALL; TANENBAUM, 2013).

O conceito de *boot over network* permite que sistemas operacionais sejam carregados a partir de uma imagem de disco localizada em um servidor de rede, eliminando a necessidade de armazenamento local em cada dispositivo cliente. Esse método pode ser implementado tanto sobre *IPv4* quanto utilizando conexões de fibra óptica. Em ambientes de alta performance, switches *Brocade* são comumente usados para facilitar a comunicação entre dispositivos de armazenamento e servidores (STALLINGS, 2015). O *boot over network* oferece várias vantagens, como a centralização da gestão de imagens de sistema e a possibilidade de manutenção e atualização simplificadas.

Os sistemas de *storage*, especialmente as RAA, desempenham um papel fundamental na infraestrutura de TI moderna, fornecendo a base necessária para o armazenamento seguro e eficiente de dados. O *boot over network* é uma funcionalidade avançada que complementa esses sistemas, proporcionando flexibilidade e facilidade de gerenciamento em grandes redes corporativas.

2.6 ZABBIX

O *Zabbix* é uma ferramenta *open-source* destinada ao monitoramento em tempo real de redes e ativos de TI. Desenvolvida inicialmente por *Alexei Vladishev* e mantida pela empresa *Zabbix SIA*, essa solução se destaca no cenário de monitoramento empresarial, oferecendo uma ampla gama de plataformas e agentes, além de capacidades de integração robustas para comunicação, manutenção e ativação de diversos ativos e equipamentos. Um de seus recursos mais notáveis é o sistema de notificações altamente customizável, que permite configurar alarmes e armadilhas para uma variedade de eventos e métricas (Zabbix,).

Para a comunicação e coleta de métricas, o *Zabbix* suporta o uso de agentes instalados diretamente nos equipamentos, além de protocolos como SNMP, IPMI, JMX e monitoramento *VMware*, ou ainda, configurações manuais integradas a outros sistemas. A configuração de um novo equipamento envolve ações em ambos os lados: no lado do agente, este é instalado e con-

figurado para se conectar ao servidor *Zabbix*; no lado da plataforma, é necessário configurar um novo objeto de inventário com dados básicos, como nome, *IP* e DNS. Para a implementação eficiente de itens de monitoramento e armadilhas, recomenda-se a criação de um *template* básico que possa ser aplicado aos inventários, especialmente para servidores de banco de dados, assegurando assim a continuidade do serviço (Zabbix, ; PEIXIAN *et al.*, 2020).

O acesso e a administração da plataforma são facilitados por uma interface *web*, que permite uma configuração gráfica e o acompanhamento das operações. Esta interface agrega uma camada de gestão de usuários e um *dashboard* personalizado, proporcionando um portal de acesso rápido e eficiente às últimas atualizações e problemas identificados (KALSIN; NOLAN, 2019). Para uma melhor visualização das informações das métricas coletadas, a utilização de ferramentas especializadas é recomendada. Por exemplo, a Empresa X emprega a *Grafana*, uma ferramenta *open-source* analítica e interativa para visualização de dados, que possui integração nativa com o *Zabbix*, facilitando o desenvolvimento de murais e painéis de acompanhamento da TI (LEPPÄNEN, 2021).

Com o monitoramento de itens no inventário do *Zabbix*, é possível configurar armadilhas (*triggers*) que, ao serem ativadas, geram um Problema caso o valor monitorado desvie de uma regra predefinida. Os Problemas representam anomalias ou falhas dentro do ambiente monitorado e são classificados em diferentes níveis de severidade, facilitando a gestão de incidentes e a resposta a eventos, conforme Figura 1.

Figura 1 – Escala de severidades de um problema.

Fonte: *Zabbix Docs* by *Zabbix SIA* (2001-2024).

A classificação inicia com a cor cinza, denominada Não Classificado (*Not classified*), utilizada para problemas que ainda não foram definidos ou cuja natureza é desconhecida. Em seguida, o nível azul claro, chamado Informativo (*Information*), é aplicado a situações que exigem clareza sobre o estado do ambiente, como alterações em arquivos ou dispositivos que foram reiniciados recentemente. O nível amarelo, conhecido como Aviso (*Warning*), é empregado para alertar sobre incidentes que indicam que, embora o sistema esteja operacional, algo não está conforme o esperado. O nível laranja, ou Médio (*Average*), destina-se a situações que requerem atenção, mas sem urgência imediata.

Para problemas mais sérios, como erros que afetam o funcionamento do sistema ou falhas em equipamentos, utiliza-se o nível vermelho claro, chamado Alto (*High*). Esse nível sinaliza que é imperativo verificar o que está ocorrendo. No topo da escala de severidade está o nível Desastre (*Disaster*), representado pela cor vermelha, que indica situações críticas como a paralisação de ambientes produtivos e a necessidade de ativação de planos de DR (Zabbix,). Essa estrutura hierárquica de classificação de problemas é fundamental para que os gestores

de TI possam priorizar as respostas a incidentes de maneira eficaz, garantindo a continuidade operacional e a integridade dos sistemas monitorados.

2.7 TRABALHOS RELACIONADOS

O estudo de Fung Po Tso em *"Network and server resource management strategies for data centre infrastructures: A survey"*, aborda estratégias de gerenciamento de recursos de rede e servidores em infraestruturas de data centers, destacando a crescente demanda por serviços de nuvem e as complexidades das operações contemporâneas. As técnicas de gerenciamento são divididas em categorias de rede e servidores, com foco em otimizar o tráfego, a largura de banda, a tolerância a falhas e o desempenho dos servidores através de métodos como a alocação de máquinas virtuais e o balanceamento de carga. O estudo ressalta a necessidade de uma abordagem integrada, que sincroniza o gerenciamento de rede e servidores para aumentar a eficiência, reduzir custos operacionais e aprimorar a entrega de serviços. Essas inovações apontam para um cenário futuro em que as operações de data centers serão cada vez mais automatizadas e otimizadas, atendendo às necessidades dinâmicas de usuários e aplicações com métodos gerenciais cada vez mais avançados (TSO; JOUET; PEZAROS, 2016).

No artigo *"On-Premise Server Monitoring with Prometheus and Telegram Bot"*, Rawoof, Favaz Mohammed e colaboradores abordam a relevância de monitorar servidores que armazenam informações críticas, como dados de usuários e registros financeiros. A pesquisa propõe uma solução utilizando o *Prometheus* para a coleta de métricas do servidor, um *software* similar ao *Zabbix*, combinada ao emprego de um *bot* do *Telegram* para o acesso remoto a esses dados. Essa estratégia possibilita aos administradores a gestão dos servidores tanto presencialmente quanto à distância, reduzindo a necessidade de verificações físicas ao essencial para a configuração inicial e manutenções periódicas. Além disso, a implementação dessa solução facilita a resposta rápida a incidentes e otimiza a eficiência operacional, fornecendo alertas em tempo real e uma plataforma de monitoramento contínuo, assegurando assim a integridade e a segurança das informações vitais (RAWOOF; TAJAMMUL; JAMAL, 2022).

O foco na aplicação do software *Zabbix* para monitoramento de dispositivos ambientais em centros de dados na Indonésia pode definir o trabalho escrito por Riyan Aji Nugroho e Perani Rosyani, *"Implementasi Monitoring Perangkat Environment Menggunakan Zabbix pada Data Center Pusat Data Sarana Informasi (PDSI)"*. A importância do monitoramento é ressaltada na seção introdutória do artigo, onde se destaca a capacidade do *Zabbix* em gerenciar a infraestrutura de TI e manter o controle sobre as operações e objetivos do centro de dados. O estudo ainda evidencia a relevância estratégica da gestão de dados e sistemas de informação no suporte às políticas de comunicação e informática no país. A metodologia adotada envolve a instalação e configuração do *Zabbix*, configurações de banco de dados e servidor, além do emprego de *templates* e SNMP para a gestão eficaz de dispositivos. O artigo conclui afirmando a eficácia do *Zabbix* em aumentar a confiabilidade e disponibilidade da infraestrutura, asse-

gurando o monitoramento e gestão em tempo real de condições críticas como temperatura e umidade. Assim, o trabalho não só detalha os processos técnicos de implementação, como também sublinha os benefícios em termos de custo-efetividade e eficiência operacional para centros de dados (NUGROHO; ROSYANI, 2023).

Em "A Implementação do *Zabbix* com Segurança: Um Estudo de Caso *Zabbix* Safely", os autores analisam a aplicação da ferramenta de monitoramento *Zabbix* em uma organização de gestão comercial voltada para o setor da construção em São Paulo, Brasil. O estudo de caso enfatiza a importância do monitoramento de rede para a prevenção de interrupções operacionais e a segurança da informação em várias infraestruturas de clientes. Inicialmente, o artigo aborda os desafios de gerenciar sistemas de clientes diversos e geograficamente dispersos, destacando a escolha do *Zabbix* devido à sua natureza de código aberto, documentação abrangente e prevalência na indústria, o que facilita a criação de um ambiente de monitoramento extensivo e seguro. No decorrer do estudo, são detalhados os componentes-chave da implementação, como a configuração de uma Rede Privada Virtual (VPN) para a troca segura de dados e a instalação do *Zabbix* em uma máquina virtual utilizando *Ubuntu*. Além disso, discute-se a configuração do *Zabbix* para monitoramento em tempo real e sua integração com o *Grafana* para melhorar a visualização de dados e a gestão de painéis. Os resultados obtidos comprovam a eficácia do monitoramento remoto em identificar e resolver proativamente problemas, aumentando a satisfação do cliente e a eficiência operacional. O artigo conclui ressaltando os benefícios do monitoramento de rede remoto e propõe a personalização das configurações de monitoramento para atender às necessidades específicas dos clientes, além de incentivar os clientes a monitorarem seus próprios sistemas de maneira independente, o que otimiza os custos operacionais e a alocação de recursos (SILVA; SILVA, 2024).

3 UTILIZAÇÃO DE SERVIÇOS E ATIVOS DE TI DA EMPRESA X

Sendo uma empresa metalúrgica que se expandiu para diversos setores do mercado, a Empresa X oferece soluções abrangentes para todo o fluxo de processos de um produto. Se há necessidade de serviços para auxiliar o processo de venda e faturamento, a Empresa X disponibiliza informações atualizadas sobre produtos, transportes e seus valores, além de registrar clientes e suas informações. Quando um pedido é faturado, os setores de compras e produção são notificados para organizar e disponibilizar os produtos para transporte na data prevista. A empresa também realiza o acompanhamento pós-venda, visando identificar melhorias e garantir a qualidade a longo prazo. Considerando que a Empresa X faz parte de um grupo internacional, além de processos mais complexos, existem também outros processos e subprocessos envolvidos, transformando a simples produção e venda em uma operação completa de análise e refinamento do produto, entregando qualidade e segurança ao cliente final.

3.1 SISTEMA DE PLANEJAMENTO DE RECURSOS EMPRESARIAIS

O sistema de ERP utilizado internamente pela Empresa X é desenvolvido e mantido internamente, baseado em uma linguagem de programão proprietária. A linguagem pode ser definida no conceito de *Low-Code, No-Code*, como descrito por Hurlburt (2021), um ambiente de baixo código requer algum conhecimento de programação, enquanto um ambiente sem código permite que os chamados "desenvolvedores cidadãos"montem aplicativos personalizados de Software como Serviço empresarial com pouca ou nenhuma habilidade de programação. Dessa forma, utilizando essa proposta de linguagem é possível escrever e entregar soluções e relatórios de forma mais rápida e dinâmica. A aplicação opera no modelo cliente-servidor, onde o processamento e execução ocorre no lado do servidor, e somente a interface gráfica é renderizada no equipamento do usuário. Isso significa que o servidor de aplicação é responsável por executar as operações principais do ERP, incluindo o processamento de dados, transações com o SGBD e a execução de cálculos e lógicas, reduzindo a necessitada de poder de processamento no lado do cliente. Os clientes locais dentro da empresa acessam o sistema através de agente instalado na máquina, que quando configurado com os parâmetros de acessos devido, estabelece uma conexão Secure Shell (SSH) com o servidor, que fornece uma maneira segura de comunicação com o servidor de aplicação (YLONEN; LONVICK, 2006).

O sistema é desenvolvido de forma altamente transacional, o que significa que todas as operações e manipulações são executadas em paralelo ou dependentes de transações no SGBD (HARIZOPOULOS *et al.*, 2018). Isso resulta em um baixo consumo de recursos e memória no servidor de aplicação, porém há um considerável aumento na carga de interações de Entrada/Saída (ES) do sistema no SGBD para leitura e atualização das informações. Essa

configuração cria a necessidade de uma rede extremamente resiliente e crítica, especialmente considerando que durante o horário de produção, entre cinco mil e dez mil usuários podem estar conectados simultaneamente ao servidor de aplicação. Além disso, nativamente, as aplicações não possuem um *pool* de conexões, o que pode resultar em falhas e erros caso um dos servidores ou a rede apresente instabilidade ou fique indisponível (FENG; YANG, 2007).

3.1.1 PORTAL INTERNO E PARCEIROS

Este sistema é operado a partir de um servidor dedicado de aplicação, projetado especificamente para isolar o tráfego de rede interno do acesso externo, evitando assim a exposição direta à internet e reduzindo conflitos de recursos entre as aplicações acessadas via agente local e aquelas operadas através do portal. O servidor executa o sistema continuamente em regime de serviço, o que permite a geração de processos dependentes para cada nova aplicação lançada. No entanto, caso haja necessidade de reiniciar o serviço, todas as aplicações ativas são encerradas e não são retomadas automaticamente, o que pode resultar em falhas operacionais temporárias. Tal comportamento necessita de monitoramento constante para assegurar a continuidade e a eficiência operacional, conforme os padrões de gestão de TI recomendam (KISTIJANTORO *et al.*, 2008).

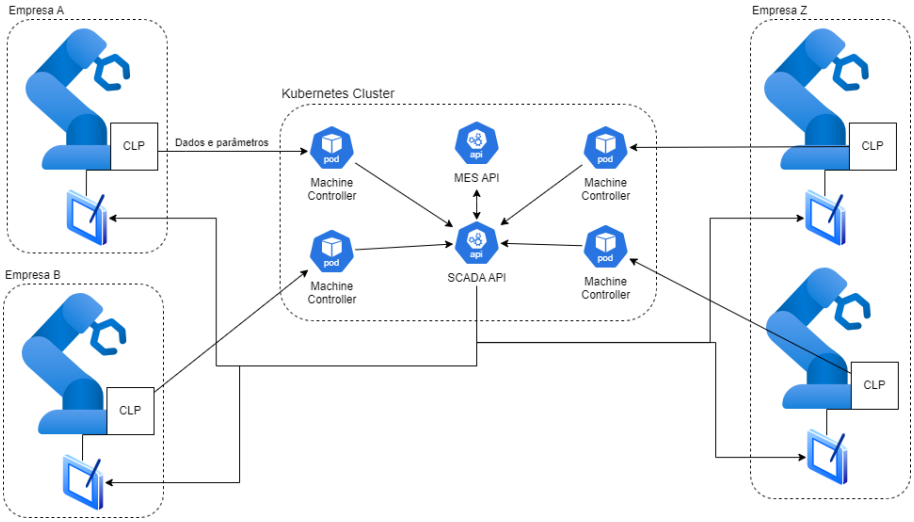
O sistema, que é disponibilizado para parceiros, representantes e terceiros, especialmente no departamento comercial, está integrado no Portal Interno com o objetivo de centralizar e facilitar o acesso. A liberação do acesso a esse portal requer um processo de autenticação rigoroso, que deve ser validado e devidamente registrado internamente para garantir a segurança e o controle de acesso. O portal é totalmente modular, permitindo a customização e a liberação de aplicativos e funcionalidades específicas durante o processo de cadastro. Isso assegura que o mesmo ambiente seja utilizado de forma eficiente tanto por usuários internos quanto por parceiros externos. Para os parceiros, em particular, são disponibilizadas ferramentas que permitem abrir ordens de vendas, realizar pesquisas em catálogos e acompanhar o *status* dos pedidos, facilitando assim a gestão comercial e a colaboração efetiva (TURRI; SANTINI, 2015).

3.2 SISTEMA DE SUPERVISÃO E AQUISIÇÃO DE DADOS

O SCADA disponibilizado para as empresas fabris integrantes do grupo Empresa X, tem suas informações e dados coletados são transmitidos ao sistema interno por meio da comunicação da aplicação local com o SGBD através de uma interface *web* e um *endpoint*. A aplicação utiliza o *IP* da máquina para identificação e estabelece conexão com um *endpoint* de Interface de Programação de Aplicações (API), gerenciado em um *cluster* Kubernetes; para mais detalhes, consulte a Seção 3.3.1. Cada CLP conta com um *endpoint* dedicado exclusivamente à transmissão de parâmetros e métricas da máquina, otimizando a coleta de informações tanto para monitoramento quanto para análises, conforme ilustrado na Figura 2. Pela definição

de (LÜBKE *et al.*, 2019), um *endpoint* de API é o extremo do lado do provedor de um canal de comunicação e uma especificação de onde os *endpoint* da API estão localizados para que possam ser acessados por clientes de API (também chamados de consumidores de API). Cada *endpoint* deve, portanto, ter um endereço único, como um Localizador de Recursos Uniforme (URL), como comumente usado na *World-Wide Web* (WWW), em *HTTP* baseado em *SOAP*, ou em *HTTP RESTful*. Cada *endpoint* de API pertence a uma API; uma API pode ter diferentes *endpoint* (SINGJAI *et al.*, 2021).

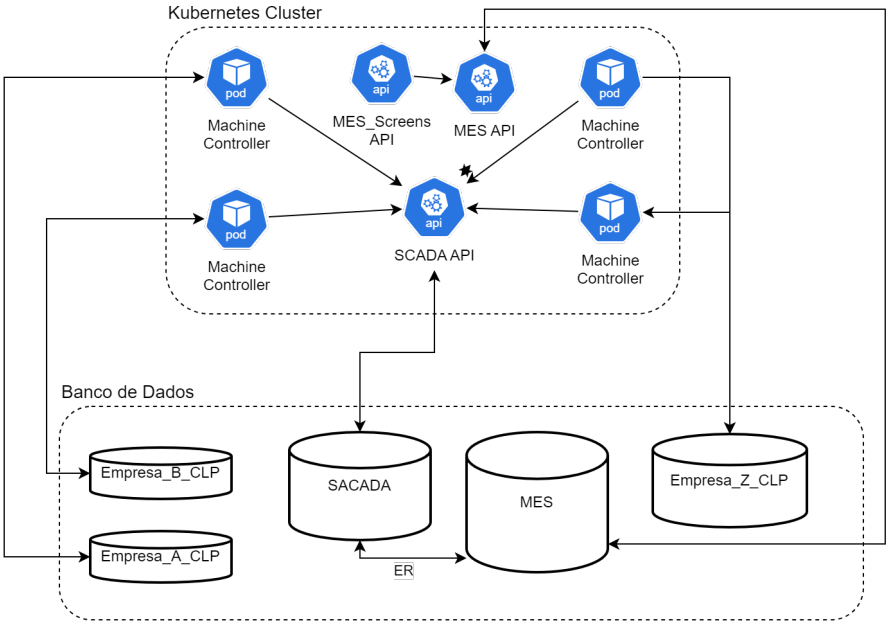
Figura 2 – Diagrama de conexão entre Máquinas e *Pods*



Fonte: Pedro Miguel Agostini Carraro (2024).

Esta API se comunica com o BD, e cada empresa do grupo mantém um BD separado, sendo identificado com o nome da empresa com o sufixo CLP, facilitando a segregação do consumo para uma cobrança de custos diferenciada, demonstrada na Figura 3. O BD *SCADA* armazena toda a estrutura e informações relacionadas ao ERP, como dados das máquinas, operadores, produtos, etc., sendo apenas um para toda a aplicação. Utilizando o *Enterprise Replication* (ER) certas tabelas e dados métricos são copiadas para o BD MES para processamento e uso analítico (IBM,).

Figura 3 – Diagrama de conexão entre *Pods* e o BD



Fonte: Pedro Miguel Agostini Carraro (2024).

3.3 SERVIDOR

Todos os sistemas próprios são mantidos em servidores locais. A capacidade de processamento é distribuída entre servidores alocados em *racks* e lâminas dentro de chassi, com todos os equipamentos apresentando redundância entre dois *Data Center* (DC), estruturas físicas onde todos os equipamentos de TI são instalados (Cisco, b). Os servidores locais permitem configurações flexíveis de memória e armazenamento, operando de forma totalmente independente ou com armazenamento gerenciado por um *storage*. Na configuração atual, as únicas máquinas configuradas dessa maneira são destinadas a aplicações críticas, tais como sistemas de diretórios ativos de domínio para a gestão e controle de usuários, grupos, computadores e objetos, bem como controladores de Access Point (AP) para a administração dos Identificador de Conjunto de Serviços (SSID) (RESENDE *et al.*, 2021).

Para as empresas e Centros de Distribuição (CD) do grupo, são disponibilizados *data centers* nas respectivas regiões de operação. No caso de operações nacionais, os *data centers* são distribuídos por região, enquanto para as operações internacionais, cada país conta com seu próprio *data center*. O tamanho e a capacidade desses *data centers* variam de acordo com a necessidade e infraestrutura local de cada empresa. No entanto, a padronização da família de produtos e modelos dos equipamentos, bem como suas configurações, é mantida para assegurar uniformidade e eficiência.

Em ambientes menores, como lojas e alguns CDs do exterior, os diversos serviços são consolidados em um único servidor. Esse servidor abriga múltiplos sistemas essenciais, como ERP, *DNS* e *firewall*, otimizando os recursos e simplificando a gestão da infraestrutura de TI.

Essa abordagem permite uma operação mais integrada e eficiente, garantindo que mesmo os ambientes com menores necessidades tenham acesso a uma infraestrutura tecnológica robusta e confiável.

3.3.1 KUBERNETES

No contexto de ambientes Kubernetes, a configuração dos nós é semelhante àquela dos servidores que compõem um ambiente de virtualização, mas sem a necessidade de criticidade e redundância exigidas para planos de Recuperação de Desastre (DR). Embora o ambiente ideal seja homogêneo, pequenas variações entre modelos e recursos dos nós não comprometem a funcionalidade do cluster. Essas divergências afetam apenas a capacidade de cada nó em suportar diferentes números de *pods*, permitindo que o Kubernetes ajuste automaticamente a distribuição das cargas de trabalho conforme as características específicas de cada nó, garantindo a eficiência e a resiliência do sistema como um todo.

3.4 ZABBIX

O monitoramento dos ativos do departamento de TI é realizado utilizando a ferramenta *Zabbix*. Para isso, são configurados agentes nos servidores e o protocolo SNMP nos equipamentos de rede, como *switches*. Em dispositivos sem suporte nativo, é configurada uma verificação básica de resposta à comunicação através do protocolo *Internet Control Message Protocol* (ICMP). Para cada ativo monitorado, são atribuídos *templates* específicos aos *hosts*, de acordo com o uso e a plataforma, permitindo a coleta de informações e métricas do sistema operacional, além do monitoramento de serviços e da conectividade.

Essa plataforma permite ao setor de operações da TI monitorar continuamente o ambiente e os ativos. Caso uma métrica apresente valores anômalos, um incidente é registrado nos murais e *dashboards* de acompanhamento, alertando os responsáveis e fornecendo informações técnicas sobre o problema. A hierarquia de problemas e dependências é configurada para exibir apenas o nível mais alto de criticidade, como no caso de uma falha em um *switch* que torna a comunicação com o servidor inacessível.

3.5 DATADOG

O *Datadog* é uma ferramenta amplamente utilizada pela Empresa X para o monitoramento de serviços e aplicações, desempenhando um papel fundamental na coleta e agregação de eventos relacionados ao desempenho e à saúde das soluções implementadas. Por meio de sua capacidade de integrar dados de diversas fontes, o *Datadog* oferece uma visão centralizada e abrangente do ambiente operacional, permitindo que a Empresa X mantenha um controle rigoroso sobre a qualidade de suas soluções. Esse monitoramento contínuo é essencial para garantir

que as aplicações estejam operando dentro dos parâmetros esperados, reduzindo assim o tempo de inatividade e melhorando a eficiência operacional.

Além disso, a utilização do *Datadog* permite à Empresa X configurar alertas personalizados que notificam imediatamente sobre falhas e problemas potenciais nas aplicações monitoradas. Esses alertas são configurados com base em métricas críticas que, quando ultrapassadas, indicam a necessidade de intervenção para evitar impactos negativos nos serviços oferecidos. Dessa forma, o *Datadog* não apenas auxilia na detecção precoce de problemas, mas também proporciona à Empresa X as ferramentas necessárias para uma resposta rápida e eficaz, alinhando-se ao princípio de monitoramento contínuo e assegurando que a qualidade dos serviços seja mantida em um nível elevado.

4 SOFTWARES DE MONITORAMENTO DE SERVIÇOS

Em um ambiente com um amplo conjunto de serviços e uma vasta base de clientes, o monitoramento constante torna-se essencial para identificar indisponibilidades ou falhas. No setor de TI, especificamente nas áreas de Desenvolvimento e Operações, é crucial acompanhar de perto a implementação e o lançamento dos serviços. Esse acompanhamento permite a identificação proativa de problemas, possibilitando sua rápida remediação e a contínua melhoria das soluções oferecidas. Para os clientes, a acessibilidade a métricas de disponibilidade é igualmente importante, pois garante a verificação da conformidade com os SLAs estipulados em contrato. Esses exemplos representam apenas uma parte das complexidades envolvidas. Para o grupo da Empresa X, existem muitos outros cenários críticos, especialmente relacionados a notificações e anúncios sobre indisponibilidades ou manutenções programadas. Assim, a comunicação eficaz e o monitoramento diligente são fundamentais para assegurar a qualidade e a continuidade dos serviços prestados.

A Seção 4.1 apresenta os conceitos e as funções dos sistemas de monitoramento de ativos e serviços, explicando suas estruturas e usos. A Seção 4.2.1 descreve os *softwares* utilizados para monitoramento de ativos e serviços. A Seção 4.3 discute o uso de uma plataforma de monitoramento como interface para o cliente, permitindo que ele acompanhe o ambiente e as soluções que o afetam. A Seção 4.4 descreve cinco soluções de plataformas de monitoramento para o cliente.

4.1 MONITORAMENTO DE SERVIÇOS E ATIVOS

O monitoramento de serviços e ativos no setor de TI é indispensável, sendo uma parte integral do processo de pós-implementação. Para o departamento de Desenvolvimento de Sistemas, após a disponibilização de uma nova versão ou ajustes, é crucial acompanhar novos cenários que possam surgir para o cliente final (DELOITTE, 2024). Para isso, são configuradas e implementadas ferramentas de agregação de eventos, com o objetivo de coletar e disponibilizar métricas dos serviços. Este acompanhamento é realizado pelo departamento de Operações de TI, que é responsável por configurar os agregadores de eventos, integrá-los aos softwares já utilizados e às novas aquisições.

O agregador de eventos desempenha o papel de receber novos eventos e métricas coletadas, processá-los e disponibilizar uma visão do estado dos serviços. A coleta de dados depende do nível de integração com as ferramentas existentes. Em alguns casos, pode haver uma integração pronta disponível na solução, requerendo apenas o mínimo de configuração necessária. Em outros casos, pode ser necessário realizar o processo manualmente, utilizando APIs e outros protocolos de comunicação, enviando dicionários ou pacotes de informações pré-processadas.

Além disso, alguns agregadores oferecem softwares próprios para integração, como agentes e bibliotecas para linguagens de programação, simplificando o processo de comunicação por meio da definição de uma chave de autenticação e ajustes nas mensagens de eventos ou métricas. No caso de agentes, podem ser definidas pastas a serem monitoradas de acordo com expressões regulares, cujos novos valores são enviados ao agregador.

Quando a informação não é pré-processada ou necessita de refinamentos finais, como indexação ou ajuste de prioridade, alguns agregadores de eventos possuem ferramentas específicas para isso. Esses fluxos de processamento podem incluir módulos para remapeamento de valores ou objetos, conversores de texto para objetos e indexadores para otimização e arquivamento. Utilizando as métricas centralizadas no agregador de eventos, é possível automatizar diversos serviços. O monitoramento automático permite configurar parâmetros e valores limites para objetos, que, em caso de anomalias e falhas, acionam sistemas de alerta por subscrição ou integrações.

De forma semelhante, o departamento de Infraestrutura de TI possui ferramentas para monitoramento dos ativos de TI. Os equipamentos são monitorados para identificar problemas e falhas, além de assegurar que serviços críticos, como SGBD e ERP, estão operando corretamente. Para esse monitoramento, é configurado um servidor específico juntamente com *dashboards* para visualização das métricas. Essa plataforma coleta métricas dos ativos e, conforme margens definidas, inicia e encerra incidentes relacionados. As métricas dos ativos são coletadas de maneira semelhante ao agregador de eventos, por meio de integrações nativas, agentes instalados nos equipamentos ou protocolos de comunicação como API e SNMP para equipamentos de rede.

No servidor de monitoramento, são configurados parâmetros e margens para os ativos, permitindo identificar serviços indisponíveis, sistemas operando com erro ou falhas totais do ambiente. Esses limites são validados com base nas métricas coletadas e, caso sejam acionados, podem executar ações configuráveis, como enviar mensagens por *e-mail* ou acionar outro *software* por API. Esta abordagem garante uma resposta rápida e eficiente a possíveis problemas, mantendo a integridade e a funcionalidade dos sistemas de TI.

4.2 SOFTWARES DE MONITORAMENTO DE SERVIÇOS E ATIVOS

4.2.1 SOFTWARES DE MONITORAMENTO DE SERVIÇOS

Uma plataforma de agregação de eventos é um sistema essencial para o monitoramento e gerenciamento dos serviços, visando seu crescimento e eficiência, a Empresa X recentemente migrou de uma solução *open-source* mantida localmente para o serviço de monitoramento da *Datadog*[1]. Sendo uma solução na nuvem para agregação e monitoramento de eventos e métricas,

[1] Disponível em datadoghq.com

o *Datadog* centraliza toda a gestão e controle de eventos, monitoramentos e alertas relacionados aos serviços da empresa.

A coleta de dados no *Datadog* é realizada principalmente através de integrações disponibilizadas em seu *Marketplace*, além de configurações manuais por meio de APIs e agentes instalados em servidores de produção. Esses agentes não apenas coletam métricas, mas também escaneiam e enviam arquivos de eventos. As informações coletadas que não estão formatadas em estruturas JavaScript Object Notation (JSON) ou que necessitam de refinamento podem ser analisadas e remapeadas, permitindo a identificação de níveis de prioridade, índices e outros metadados relevantes.

Além disso, o monitoramento de APIs, métricas e outros objetos no *Datadog* é facilitado por ferramentas nativas que possibilitam a definição de margens e limites para o acompanhamento dos valores lidos. Isso permite um controle mais preciso e automatizado dos serviços, garantindo que quaisquer anomalias ou falhas sejam rapidamente identificadas e tratadas. Com a migração para o *Datadog*, a Empresa X não apenas melhorou a eficiência do monitoramento de seus serviços, mas também obteve uma solução mais escalável e integrada, capaz de atender às demandas crescentes de seu ambiente de TI.

4.2.2 SOFTWARES DE MONITORAMENTO DE ATIVOS

O constante monitoramento de equipamentos é indispensável no setor de Infraestrutura de TI, pois é responsável não apenas pela sustentação dos ativos, mas também dos serviços que operam neles. Para a coleta e monitoramento dos ativos, utiliza-se a solução *Zabbix*[2], configurada em um servidor em conjunto com o software *Grafana*[3]. A coleta de informações é realizada por agentes instalados nos servidores, que obtêm dados sobre o consumo de recursos do sistema operacional, serviços em funcionamento e qualidade das conexões. Para equipamentos de rede, é utilizado o protocolo SNMP, que permite identificar configurações de portas, uso de recursos e qualidade da conexão.

A utilização do *Zabbix* e do *Grafana* proporciona uma visão detalhada e em tempo real do desempenho e do estado dos equipamentos de TI. Isso possibilita a definição de limites e margens para as métricas coletadas, o que é crucial para a detecção precoce de problemas. Quando os valores monitorados ultrapassam os limites predefinidos, o sistema gera incidentes automáticos. Essas regras são configuradas no formato de gatilhos, *triggers*, que abrem um incidente até que os valores sejam normalizados ou o problema seja resolvido manualmente.

Além da geração de incidentes, o sistema pode ser configurado para executar ações automáticas tanto no acionamento quanto no encerramento de um incidente. Isso inclui a abertura automática de chamados relacionados a problemas, envio de avisos aos responsáveis locais pelos equipamentos e, em casos específicos, a execução de ações de remediação automática, como

[2] Disponível em zabbix.com
[3] Disponível em grafana.com

a reinicialização de um serviço que apresentou falha. Essas medidas aumentam significativamente a eficiência da gestão de incidentes, minimizando o tempo de inatividade e melhorando a continuidade dos serviços.

Portanto, o monitoramento contínuo dos equipamentos de TI utilizando soluções como *Zabbix* e *Grafana* é essencial para manter a operacionalidade e a qualidade dos serviços prestados. A capacidade de detectar e responder rapidamente a problemas, junto com a possibilidade de automação de ações corretivas, contribui para a robustez e a confiabilidade da infraestrutura de TI, garantindo que os serviços permaneçam ativos e eficientes, mesmo diante de desafios técnicos.

4.3 INTERFACE DE MONITORAMENTO PARA O CLIENTE

O constante monitoramento do setor de TI é parte integrante de suas rotinas de trabalho e responsabilidades. No entanto, surge a questão: como essa vigilância é apresentada ao usuário final? A comunicação eficaz entre o cliente e o operador de TI é de vital importância, especialmente quando realizada de maneira madura e formal. Por exemplo, isso pode ocorrer através da requisição de reparos de defeitos por meio de chamados ou de notificações informando que uma determinada região ficará sem comunicação devido a uma falha que já está sendo investigada.

Sem uma troca de informações eficiente, o processo de aprimoramento do desenvolvimento pode se tornar reativo, resultando em respostas tardias e menos eficazes às demandas e problemas. A comunicação proativa, por outro lado, não apenas facilita a resolução rápida de problemas, mas também contribui significativamente para a construção de uma relação de confiança entre o setor de TI e seus usuários. Essa relação de confiança é fundamental, pois permite que os usuários se sintam seguros e satisfeitos com os serviços recebidos. Além disso, a comunicação proativa promove uma melhoria contínua e eficaz dos serviços oferecidos, garantindo que as necessidades e expectativas dos usuários sejam atendidas de maneira ágil e eficiente. Ao estabelecer uma comunicação aberta e constante, o setor de TI pode antecipar problemas, implementar soluções preventivas e, consequentemente, elevar a qualidade dos serviços prestados, criando um ambiente mais colaborativo e produtivo.

A comunicação inadequada entre o setor de TI e os clientes pode resultar em diversos problemas, como sistemas indisponíveis inesperadamente, manutenções não agendadas ou disponibilidade parcial dos serviços. Esses incidentes prejudicam a qualidade e a experiência de uso do cliente, que pode acabar procurando soluções alternativas fora do departamento de TI, configurando um dos piores cenários possíveis. Para evitar e mitigar esses problemas, é fundamental implementar um meio eficiente de transmitir ao usuário final informações sobre eventos em andamento e planejados, além de permitir a verificação em tempo real do estado operacional dos serviços. Contudo, é essencial evitar o uso de termos técnicos ou informações irrelevantes que não afetem a experiência do usuário, apresentando os serviços de forma simplificada e

acessível.

Uma plataforma para disponibilizar o estado dos serviços, incidentes e seus históricos representa uma solução eficaz para o problema de comunicação entre os departamentos de TI e os clientes finais. Ao centralizar as informações coletadas pelos sistemas de monitoramento da TI, essa plataforma organiza os dados em objetos visíveis e compreensíveis para os usuários. Isso não apenas melhora a observabilidade das soluções da Empresa X, mas também transforma e relaciona informações técnicas dos ativos e serviços em estados de operação do serviço principal, fornecendo uma visão macro do ambiente. Essa visão facilita a identificação de perdas de desempenho ou indisponibilidade, tornando o processo mais transparente e eficiente.

Os estados de operação de um serviço podem ser categorizados em diferentes níveis. Um estado normal indica que não há problemas identificados, enquanto um desempenho reduzido ocorre quando o consumo de recursos do serviço atinge seus limites, resultando em lentidão no ambiente. Em casos de falhas, a indisponibilidade parcial é registrada quando parte do ambiente e ativos estão fora de operação, afetando certas funcionalidades do serviço ou regiões específicas. A indisponibilidade total, por sua vez, é resultado de uma falha completa do ambiente, tornando os serviços inacessíveis. Esses estados ajudam a Empresa X a monitorar e responder prontamente a diferentes níveis de incidentes, garantindo uma gestão mais eficaz dos recursos e da infraestrutura.

Além disso, os serviços podem possuir um histórico de operação que oferece uma visão ao longo do tempo, identificando falhas constantes ou padrões de instabilidade. Esse histórico, acompanhado pelos SLAs, demonstra se as soluções operam de acordo com os contratos estabelecidos. A existência de registros de incidentes reportados ou em andamento permite aos clientes acompanhar o progresso da resolução dos problemas sem precisar acionar o suporte de TI. Isso reduz o volume de chamados redundantes durante falhas críticas ou indisponibilidades parciais, otimizando a eficiência do suporte e melhorando a satisfação do cliente.

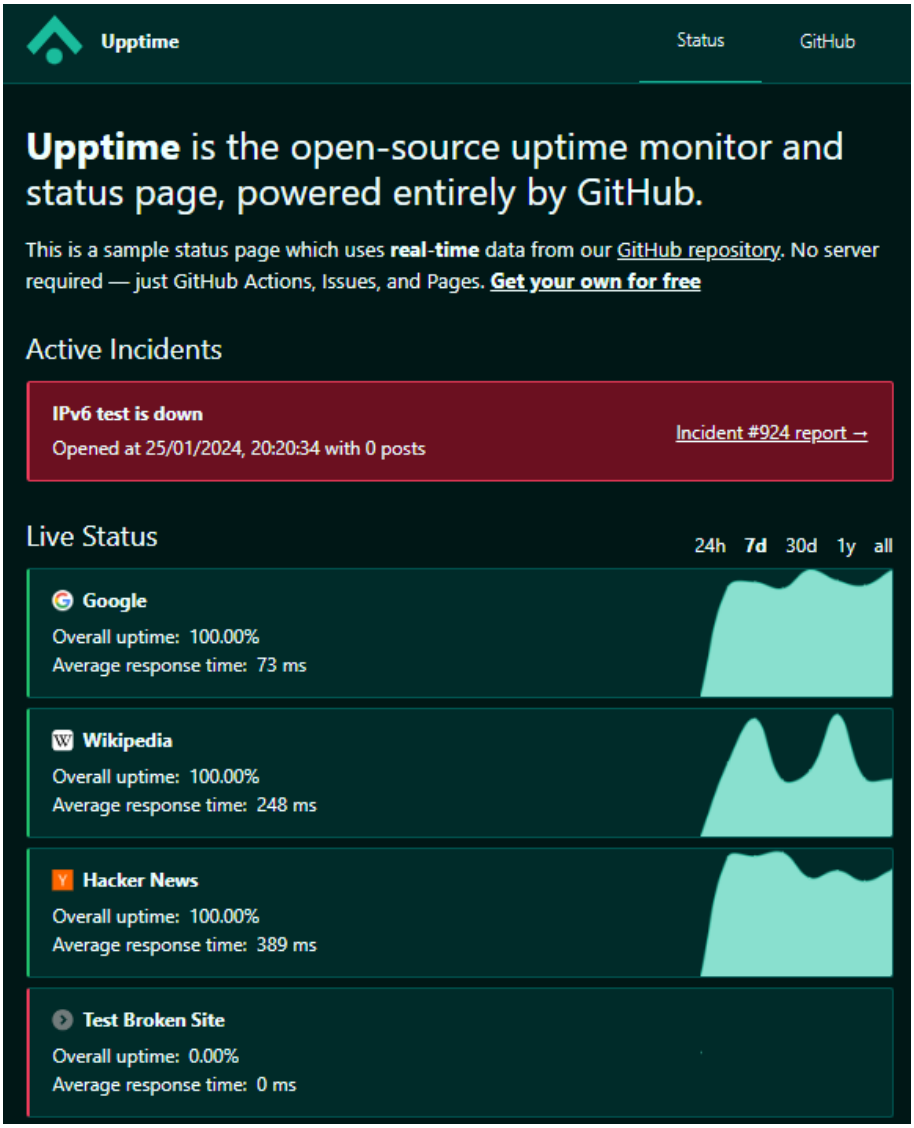
4.4 SOLUÇÕES DE MONITORAMENTO DE SERVIÇOS

Para garantir uma comunicação clara e eficaz com os usuários, especialmente em momentos de falhas ou manutenção, a implementação de uma página de *status* é essencial. Essa seção detalha diversas propostas de páginas de *status*, explicando suas características, benefícios e como podem ser personalizadas para atender às necessidades específicas de diferentes organizações. Abordaremos desde soluções prontas para uso até opções mais avançadas que permitem uma integração profunda com os sistemas internos, garantindo que as informações de *status* sejam sempre precisas e atualizadas.

4.4.1 UPPTIME

O *Upptime*[4] é uma solução de monitoramento de serviços desenvolvida com foco na disponibilidade e no desempenho de *websites* e *APIs*. Sua natureza *open-source* oferece uma alternativa acessível e transparente às ferramentas de monitoramento tradicionais, destacando-se por sua simplicidade e eficácia, como visível na Figura 4. A integração do *Upptime* com o *GitHub* permite que todos os dados de monitoramento sejam armazenados em repositórios, públicos ou privados, facilitando o acesso e a gestão dos dados. Esta transparência não só promove a confiança dos usuários, mas também proporciona uma visão contínua e clara das operações, essencial para a gestão eficaz de serviços online.

Figura 4 – Upptime - Página de Status



Fonte: Página de Status de Demonstração por CHOWDHARY

A plataforma abrange o monitoramento de uma ampla gama de serviços, incluindo *web* e *APIs*, registrando automaticamente incidentes e manutenções programadas. A detecção de

[4] Disponível em upptime.js.org

falhas aciona a criação automática de *Issues* no *GitHub*, detalhando os serviços afetados. Esta funcionalidade assegura que nenhuma interrupção passe despercebida e que as equipes possam agir rapidamente para resolver problemas. Além disso, o suporte para configuração de manutenção programada permite aos usuários preparar e ajustar o sistema para evitar falsos alarmes durante períodos de atualização planejada, mantendo a integridade do monitoramento.

Figura 5 – Upptime - Informações de Incidente

Fonte: Página de Status de Demonstração por CHOWDHARY

Quando ocorrem incidentes, o *Upptime* não se limita a registrar o problema; ele também notifica as equipes através de integrações configuradas com plataformas de comunicação como *Slack*, *Discord* e *Telegram*. Os *webhooks* personalizados permitem a implementação de ações automáticas, garantindo que as respostas aos incidentes sejam tanto rápidas quanto eficientes. Esta capacidade de resposta imediata é vital para minimizar o impacto dos problemas e garantir a continuidade dos serviços monitorados.

Em termos de implementação, o *Upptime* é totalmente hospedado em nuvem, utilizando as páginas do *GitHub* para armazenar todos os dados de monitoramento. Essa abordagem baseada em nuvem não apenas simplifica a infraestrutura necessária, mas também proporciona escalabilidade e facilidade de acesso de qualquer local. A configuração inicial requer apenas a definição dos serviços a monitorar, as frequências de verificação e as integrações desejadas. Isso permite que a plataforma seja ajustada para atender às necessidades específicas de cada organização.

Além disso, o *Upptime* suporta integrações com outras ferramentas de monitoramento como Zabbix e Datadog, facilitando uma visão mais abrangente e centralizada do desempenho dos serviços. Essas integrações, realizadas através de *APIs* e *webhooks*, permitem que os dados de monitoramento sejam enriquecidos e mais detalhados, oferecendo visibilidade mais profundos sobre o desempenho dos serviços. Além disso, a integração com sistemas de ges-

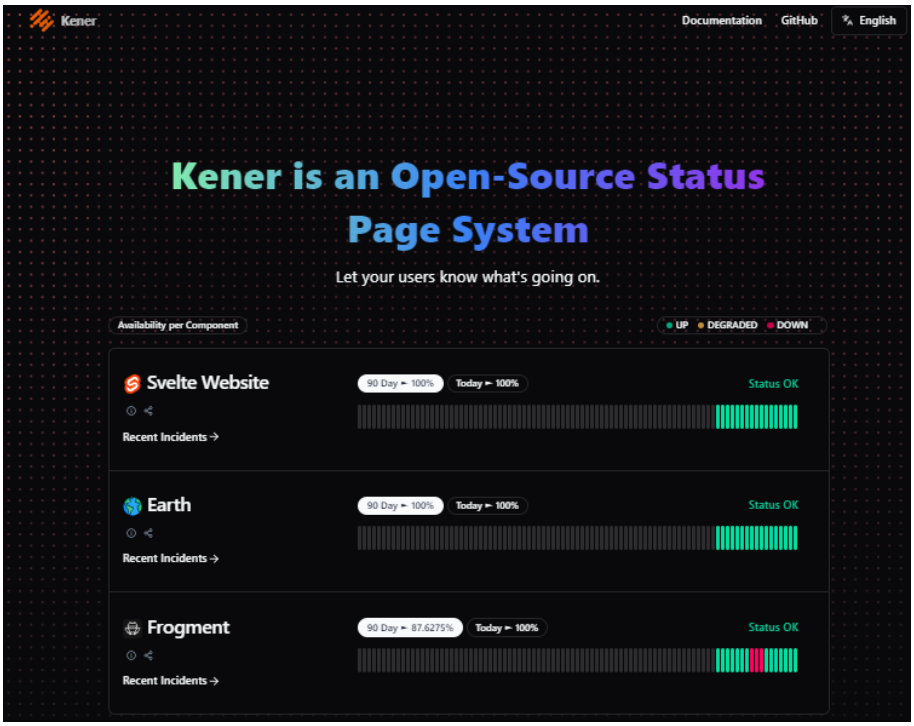
tão de chamado facilita o acompanhamento e a resolução eficaz de problemas, integrando o monitoramento com as operações de TI e suporte.

O *Upptime* representa uma evolução no monitoramento de serviços digitais, combinando automação, integração e acessibilidade. Sua capacidade de adaptação e a abordagem baseada em software livre tornam-no uma ferramenta valiosa para qualquer organização que busca melhorar a gestão e a visibilidade de seus serviços online. Com seu sistema robusto e fácil de configurar, o *Upptime* está configurado para ser um aliado fundamental na otimização do desempenho e na garantia da disponibilidade de serviços digitais críticos.

4.4.2 KENER

Sendo uma página de estados moderna, *Kener*[5], tenta trazer a experiencie e qualidade de serviços de monitoramento contratados, de forma local e auto sustável, como demonstrado na Figura 6. Inspirado na plataforma de monitoramento *Upptime*, descrito na Seção 4.4.1, *Kener* é desenvolvido em *Node.js* utilizando o *framework Svelte*, oferecendo rápido desenvolvimento e criação de componentes dinâmicos facilmente. Configuração da plataforma e dos monitores de serviços é gerenciado em arquivos YAML posicionados em pastas especificas. Com os objetos criados, ajustes e manutenções podem ser realizados requisições a API do *Kener*, tendo opções para gestão completa, exceto criação e remoção, de serviços e incidentes.

Figura 6 – Kener - Página de Status



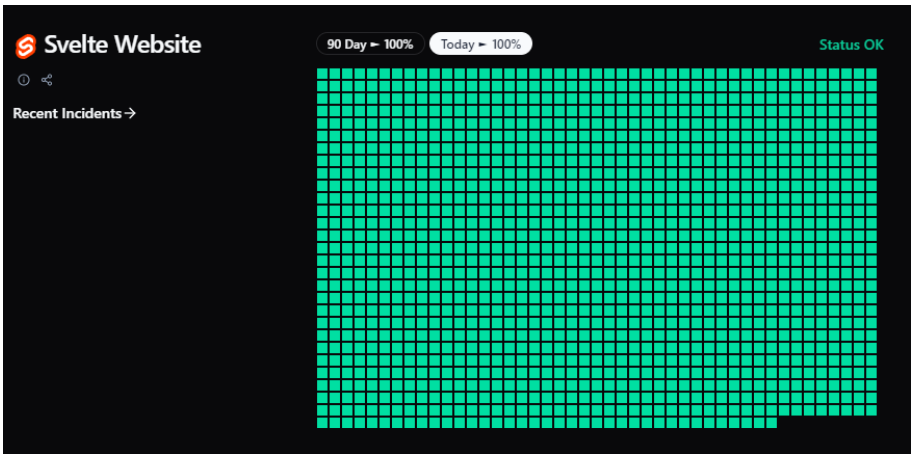
Fonte: Página de Status de Demonstração por SHARMA

[5] Disponível em kener.ing

A gestão de incidente é integrada com o sistema de *issues* no *Github*, que para serviços que possuam repositórios, é ideal, auxiliando o setor de desenvolvedores centralizar suas demandas, contudo para grupos maiores, como o da Empresa X, que utilizam outras soluções para gestão de chamados, essa opção não apresenta valor. Para trocar os serviços de gerenciamento dos incidentes é necessário desenvolver uma integração própria, e alterando o projeto original. Para soluções que não apresentam integração nativa com o sistema de chamado, como está, uma maneira de implementar a gestão de chamados para as ferramentas de monitoramento de ativos e serviços, e elas ao identificar um problema gerar um incidente e um chamado vinculados.

O monitoramento de serviços é realizado por meio de requisições personalizadas, que avaliam as respostas recebidas e são agendadas para execução periódica. Isso permite um controle completo sobre a validação do estado do serviço. As métricas coletadas são armazenadas e apresentadas em um resumo dos últimos 90 dias, bem como todas as leituras do último dia, como mostrado na Figura 7. Esses valores coletados são utilizados para a determinação dos SLAs dos serviços, proporcionando uma visão abrangente do desempenho e disponibilidade dos serviços monitorados. Dessa forma, é possível garantir que os serviços atendam aos níveis de qualidade acordados, permitindo uma gestão proativa e eficaz dos recursos tecnológicos.

Figura 7 – Kener - Histórico de Monitoramento



Fonte: Página de Status de Demonstração por SHARMA

Um sistema de subscrição e atualizações não está presente no *Kener*, por estar no início de seu ciclo de desenvolvimento. Essa solução não possui o lançamento de sua primeira versão oficial, por conta disso, alguns requisitos não atendidos no momento, mas estão no mapa de melhorias do projeto.

Um sistema de subscrição e atualizações ainda não está presente no *Kener*, pois a solução está no início de seu ciclo de desenvolvimento. A primeira versão oficial do *Kener* ainda não foi lançada, o que implica que alguns requisitos não foram atendidos até o momento. No entanto, esses requisitos já estão mapeados para melhorias futuras no projeto. À medida que o

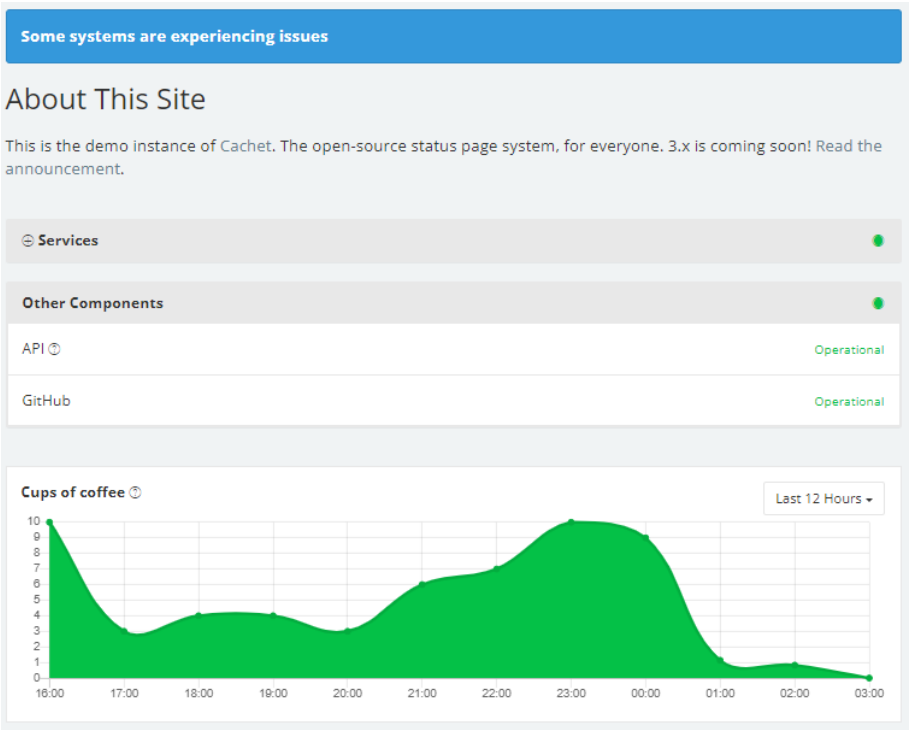
projeto avança, espera-se que essas funcionalidades sejam gradualmente introduzidas, melhorando a usabilidade e a eficiência do *Kener*.

Essa plataforma pode ser implementada, tanto em um servidor local manualmente, quanto utilizando a imagem de contêiner oficial. Todo o processo de configuração e o processo de instalação faz parte da documentação da ferramenta, sendo bem direto e simples de proceder. O contêiner poder ser gerado pelos arquivos fontes da última versão do repositório, ou ser utilizado uma imagem disponibilizada pelo autor, implementação em ambiente de processamento computacional em nuvem não possui documentação oficial.

4.4.3 CACHET

Uma página de *status open-source* desenvolvida em *PHP*, que não apenas oferece uma ampla variedade de opções, mas também apresenta visuais modernos e atraentes. O *Cachet*[6] é uma solução robusta para o monitoramento de serviços e suas métricas, baseada em integração por API ou por bibliotecas de funções que permitem a integração nas principais linguagens de programação. A plataforma oferece dois tipos principais de objetos para monitoramento: componentes, que representam os serviços e ativos, e métricas, que permitem o acompanhamento ao longo do tempo, como demonstrado na Figura 8. Os componentes podem ser organizados em categorias, o que facilita a visualização e o gerenciamento global das operações.

Figura 8 – Cachet - Página de Status



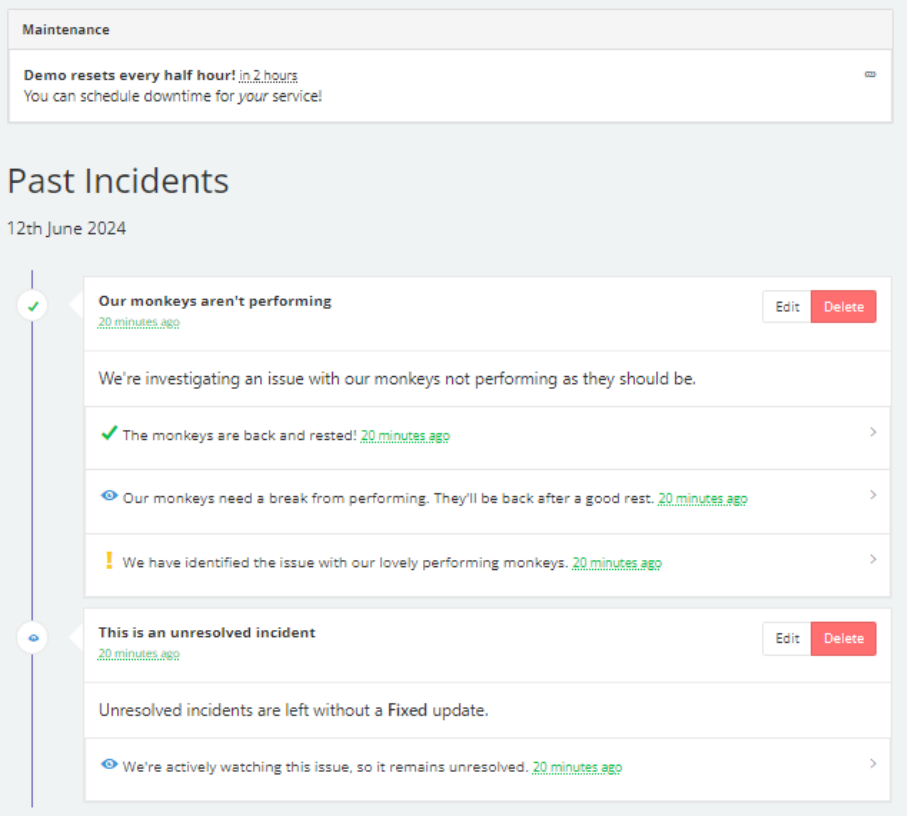
Fonte: Página de Status de Demonstração por LIMITED

[6] Disponível em cachethq.io

A plataforma, sendo baseada em integração, não possui um sistema de monitoramento nativo e deve ser configurada e atualizada manualmente. No entanto, através de integrações, é possível coletar dados e métricas de diversos serviços para alimentar o *Cachet*. Isso inclui monitoramento de *sites* e coleta de informações de outros serviços de monitoramento. O autor oficializou algumas integrações específicas, como para comunicação com o *Zabbix*, permitindo a criação de objetos e a coleta de informações. Para integrar com a plataforma do *Datadog* ou com sistemas de chamados, é necessário desenvolver *plugins* customizados. Essas customizações possibilitam a ampliação das capacidades de monitoramento e gerenciamento da plataforma, adaptando-a às necessidades específicas de cada ambiente.

A plataforma descrita adota um sistema de subscrição que permite atualizações abrangentes para todos os objetos ou componentes específicos, proporcionando aos usuários uma flexibilidade significativa na gestão de informações. No entanto, observa-se a ausência de opções para atualizações específicas sobre métricas, manutenção programada ou incidentes, o que pode limitar a capacidade de resposta proativa a esses eventos. As notificações, por padrão, são enviadas apenas por *e-mail*, o que, embora útil, pode não atender às necessidades de comunicação de todos os usuários.

Figura 9 – Cachet - Informação do Incidente



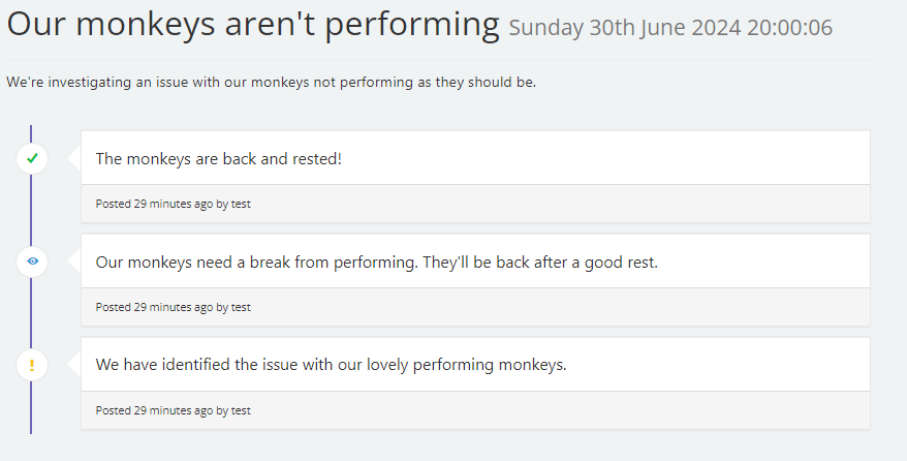
Fonte: Página de Status de Demonstração por LIMITED

Caso um serviço apresente estar sofrendo de degradação de desempenho ou instabilidade, um incidente pode ser aberto para reportar o caso. A opção de abertura automática poder

ser configurado ou por integrações de monitoramento, ou por ferramentas externas, contendo qual o serviço afetado, podendo ser apenas um, uma breve descrição dos casos e estado de progresso, como é possível observar na Figura 9. Um incidente pode estar em estado investigando, sendo o estado inicial, o estado identificado para quando o problema foi isolado e uma solução está sendo proposta. Com uma solução em andamento, o incidente pode estar no estado de observação, acompanhando se foi ajustado a falha, com o encerramento do incidente ele é passado para o estado de consertado. Cada atualização do incidente, possui uma breve descrição do progresso, e se necessários, retornar para um estado anterior.

Um histórico de incidentes passados está disponível ao final da página. Nele é possível ver todos os incidentes que ocorreram ao longo do tempo, os serviços afetados e em qual estado eles foram encerrados. Ao início dessa seção da página, é listado as manutenções programadas, incluído informações como tempo até a indisponibilidades e uma breve descrição com título, como mostrado na Figura 10. A manutenção pode ser programada, ou criado como registro histórico, definido um título, uma descrição do planejamento e para quando está agendado, podendo estar no estado de agendado, em progresso ou concluído.

Figura 10 – Cachet - Histórico de Incidentes



Fonte: Página de Status de Demonstração por LIMITED

Implementação dessa solução pode ser com um serviço em servidor local, ou como um contêiner em um ambiente *Kubernetes*. Todo o processo de instalação, configuração e preparação do ambiente está documentado, baseando-se em arquivos modulares de configuração. Uma imagem de contêiner já está disponível para preparo e inicialização, e as principais integrações oficializadas possuem também imagens próprias. Contudo, considerando o problema de hospedar localmente os serviços, não há documentação relacionada à implementação em ambientes de computação em nuvem, como *AWS* da *Amazon* ou *GCP* da *Google*.

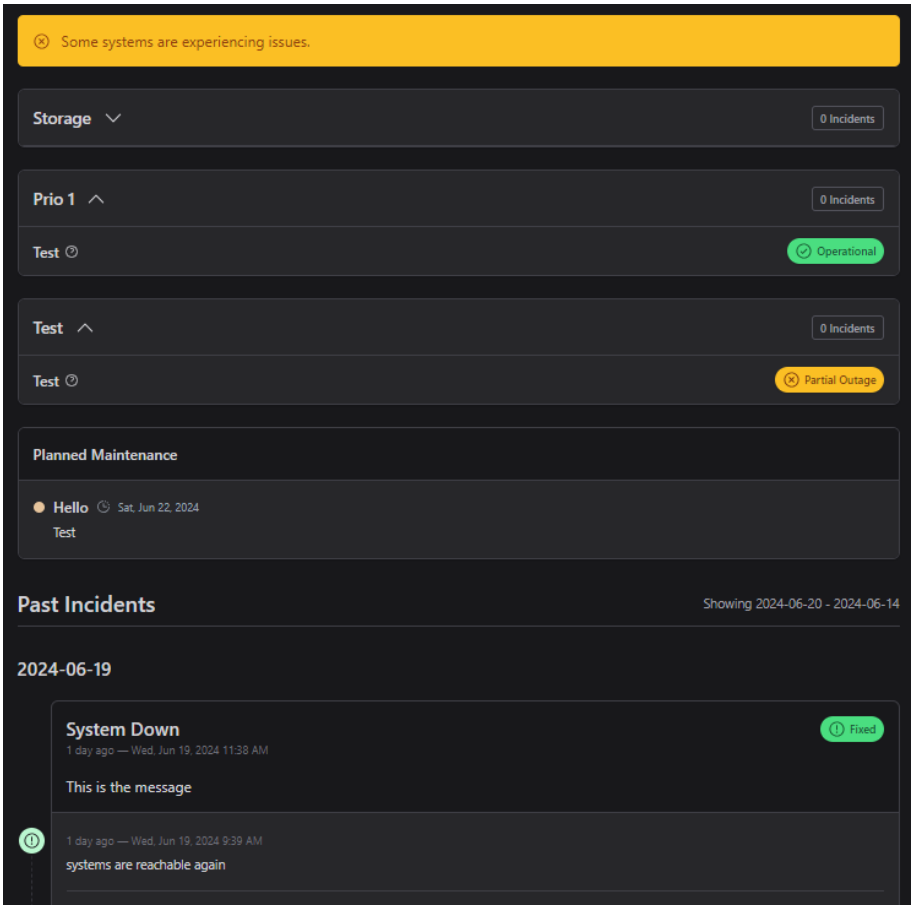
Para controle e manutenção da plataforma, o *cachet* possui uma página chamada *Dashboard*. Para uso administrativo, esse *site* oferece o controle dos objetos, incidentes e configurações, sendo acessado pela autenticação de usuário local, podendo habilitar autenticação de dois

fatores; não está prevista uma integração nativa para uso de Lightweight Directory Access Protocol (LDAP). Os usuários locais podem possuir acesso administrativo da plataforma, ou acesso de usuário, podendo manipular serviços e métricas, mas não usuários e outras configurações da plataforma.

4.4.3.1 CACHET 3.X

Na Seção 4.4.3 foi descrito as propriedades e funcionalidades da versão 2.4.1 LTS, lançada em 7 de novembro de 2023, essa versão foi desenvolvida e mantida por uma empresa externa que comprou a solução *Cachet* do autor original. Com a reaquisição, o autor está desenvolvendo a nova versão 3.x da plataforma; essa versão é a reestruturação de todo o projeto, reescrevendo a maior parte do código fonte em *Laravel* um *framework* de *PHP*. Com o lançamento dessa nova versão, o suporte para versões 2.x será finalizado e elas entrarão em final de vida.

Figura 11 – Cachet 3.x - Página de Status



Fonte: Página de Status de Demonstração por LIMITED.

Demonstrações digitais já estão disponíveis para a nova versão, Figura 11, apresentando quase todas as funcionalidades da versão anterior, com foco na melhoria dos visuais e na ampliação das opções de customização. Contudo, a principal diferença está na organização do projeto.

Na versão 2.x, a plataforma possui as estruturas de *backend* e *frontend* em um único projeto, o que não segue as boas práticas de engenharia de software. Na nova versão, o projeto foi segregado em seu núcleo, que contém as funcionalidades, API e processamento em *background*, e um projeto separado para a interface gráfica. Essa segregação entre *backend* e *frontend* oferece benefícios significativos, como maior modularidade, facilidade de manutenção e a possibilidade de evolução independente de cada componente.

4.4.4 STATUSPAGE

Statuspage[7] da *Atlassian* é uma plataforma poderosa projetada para manter seus usuários informados sobre o *status* dos seus serviços em tempo real. Esta solução oferece uma interface elegante e amigável, facilitando a comunicação de atualizações operacionais e incidentes para o seu público, sendo possível perceber na Figura 12. Sendo baseada em uma solução contratável de terceiros, essa plataforma está disponível em nuvem, que além de permitir o desacoplamento do monitoramento do ambiente local, também permite integrações nativas com outras soluções em nuvem.

Sendo capaz de monitorar componentes e serviços de terceiros, o *Statuspage* pode ter seus objetos organizados em grupos de serviços, ou valores isolados. Os grupos apresentam histórico médio de seus componentes, para uma visão mais macro, porém não é possível atribuir um grupo a outro grupo, criando uma organização mais complexa. Todos os serviços criados manualmente devem ser atualizados por uma ferramenta externa, por integrações com API, para monitoramento automático, pode ser utilizado integrações prontas soluções de terceiros, disponibilizadas no *marketplace*. Serviços de terceiros podem, não obrigatório, possuir histórico de operação, como seu estado de operação, e para alguns casos possuir segmentação entre funcionalidades e ferramentas da ferramenta, trazendo mais visibilidade na operação do serviço.

Um robusto sistema de alertas é essencial para manter os usuários atualizados sobre o estado do ambiente de serviços. Este sistema oferece configurações flexíveis, permitindo que os usuários se inscrevam em alertas para componentes e incidentes específicos ou sejam automaticamente notificados sobre novos problemas. A entrega de atualizações é altamente versátil, abrangendo várias formas de comunicação. *E-mail* é a opção mais comum e preferida para alertas, mas o sistema também suporta mensagens via Serviço de Mensagem Curta (SMS), *feeds RSS*, e integrações com sistemas de mensagens corporativas como o *Microsoft Teams* e o *Slack*. Além disso, a plataforma permite comunicações através de API ou *webhook*, facilitando a integração com diversas outras plataformas e maximizando a eficiência na gestão de incidentes e na resposta a eventos críticos.

Todos os incidentes são meticulosamente registrados no histórico de eventos ao final da página, conforme ilustrado na Figura 13. Cada registro inclui detalhes das atualizações, permitindo aos usuários acompanhar o progresso de um incidente desde seu início até a resolução.

[7] Disponível em atlassian.com/software/statuspage

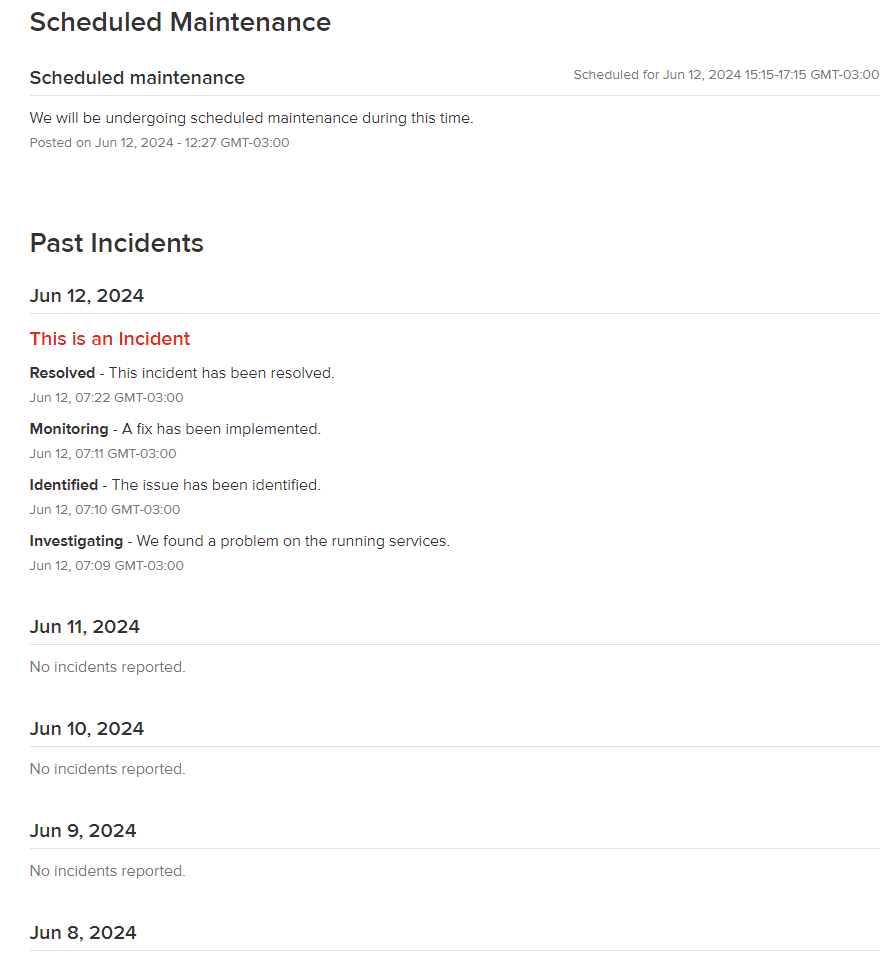
Figura 12 – Statuspage - Página de Estados

Fonte: Página de Status de Demonstração por ATLASSIAN

Além disso, é possível que um incidente esteja associado a um ou mais componentes ou grupos, facilitando a identificação e a gestão dos elementos afetados. Os usuários podem acessar e pesquisar no histórico para verificar se há eventos que impactam o estado operacional dos serviços. Esta funcionalidade assegura que todas as partes interessadas estejam cientes dos problemas em tempo real e possam responder apropriadamente.

Manutenções programadas são tratadas como eventos especiais e são posicionadas no início do histórico completo. Estes eventos interrompem temporariamente a coleta de dados e

Figura 13 – Statuspage - Histórico de Eventos



Fonte: Página de Status de Demonstração por ATLASSIAN

automações para permitir ajustes críticos e atualizações nos serviços. Cada manutenção possui um título, uma descrição detalhada, a data de início e a duração prevista. Elas podem ser agendadas para atuar sobre um ou mais objetos e seus respectivos grupos. Ademais, configurações de notificações permitem alertar os clientes no início da manutenção, enviar lembretes durante o procedimento e notificar ao concluir, além de manter a equipe de TI e os administradores atualizados com os progressos através de alertas regulares.

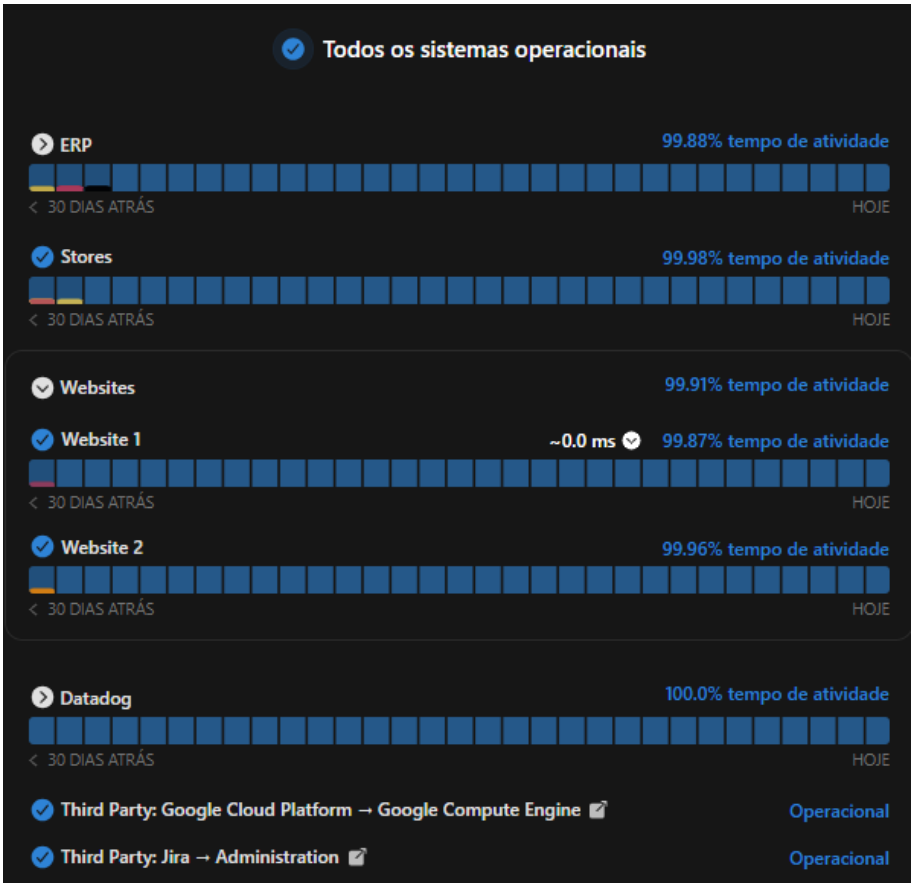
A plataforma opera na nuvem e sua eficácia depende das configurações e integrações estabelecidas no ambiente de TI. Um *website* administrativo está disponível para gerenciar a plataforma e ajustar suas configurações, onde os usuários podem configurar componentes manualmente ou utilizar opções de integração pré-definidas. Além disso, personalizações da interface gráfica podem ser realizadas para adaptar a plataforma às necessidades específicas da organização. O *site* também oferece módulos dedicados ao gerenciamento de incidentes e ao agendamento de manutenções, permitindo a atualização ou conclusão de casos abertos. Para integrações externas, pode-se utilizar a API da plataforma para atualizar objetos e incidentes, ou empregar o sistema de *webhook* configurado nas subscrições, permitindo uma comunicação

eficiente e bidirecional entre a plataforma de monitoramento e os serviços gerenciados.

4.4.5 INSTATUS

Sendo publicamente anunciada como uma alternativa mais rápida e econômica ao *Statuspage*, a plataforma *Instatus*[8] prometendo revolucionar o acompanhamento de serviços online. Com um design moderno e uma interface minimalista, *Instatus* não apenas simplifica a visualização das informações cruciais, mas também proporciona uma análise detalhada e profunda do estado dos serviços, como demonstrado na Figura 14. Esta abordagem permite aos usuários identificar rapidamente a origem e a natureza de quaisquer interrupções ou degradações de serviço, tornando-a uma ferramenta essencial para equipes de TI que precisam de soluções eficientes e acessíveis para a gestão de incidentes. Além disso, sua plataforma é projetada para ser intuitiva, facilitando a adoção por novos usuários que buscam implementar práticas de observabilidade em seus processos operacionais.

Figura 14 – Instatus - Página de Estados



Fonte: Página de Status de Demonstração por INSTATUS

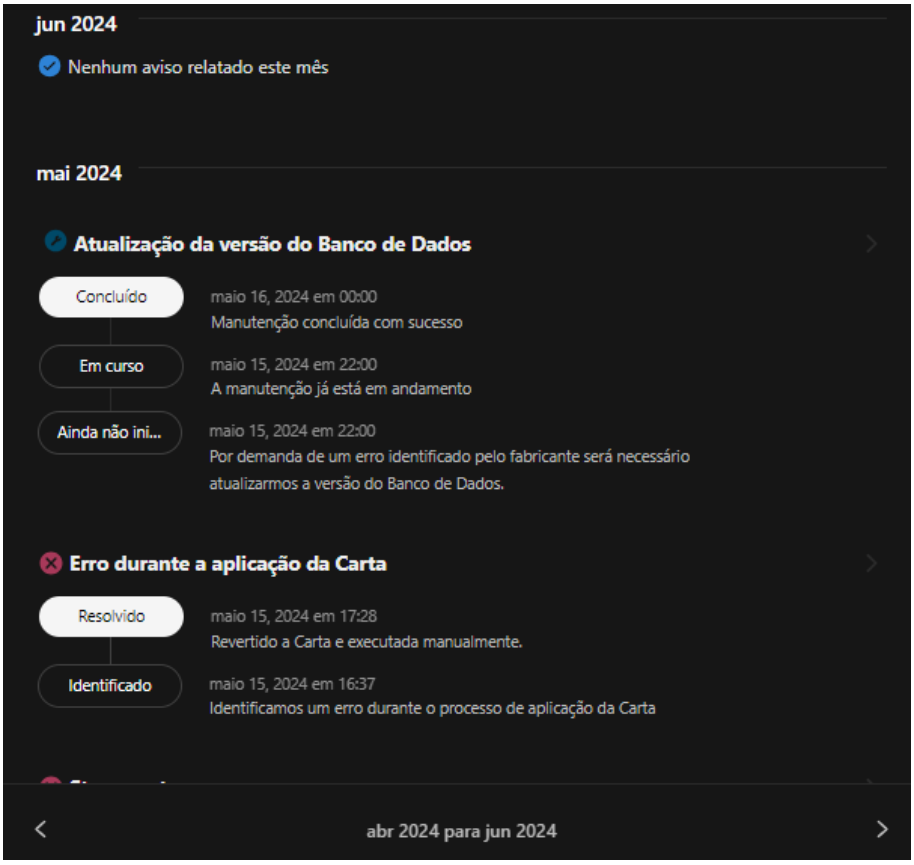
A plataforma proporciona um abrangente sistema de monitoramento de serviços, que pode ser configurado manualmente ou por meio de uma ampla coleção de integrações oficiais.

[8] Disponível em instatus.com/home

Os usuários têm a flexibilidade de atualizar o estado operacional dos serviços tanto manualmente quanto automaticamente, utilizando a API da plataforma. Adicionalmente, é possível configurar *webhooks* para facilitar o gerenciamento de incidentes em outras plataformas, tornando o processo mais integrado e automatizado. Esta funcionalidade destaca-se por sua adaptabilidade e eficiência, permitindo uma resposta rápida a problemas e uma gestão de incidentes mais fluida.

Além do monitoramento em tempo real, a plataforma oferece uma funcionalidade de histórico operacional que pode ser customizada para períodos específicos de 30, 60 ou 90 dias, ou até mesmo ser desativada conforme a necessidade do usuário. Notavelmente, o monitoramento de serviços de terceiros não inclui histórico ou métricas operacionais, uma limitação importante para usuários que dependem desses dados para análises detalhadas. No layout da plataforma, incidentes ativos e manutenções programadas são destacados no início das páginas, enquanto o histórico de eventos encerrados é acessível ao final, como ilustrado na Figura 15.

Figura 15 – Instatus - Histórico de Incidentes

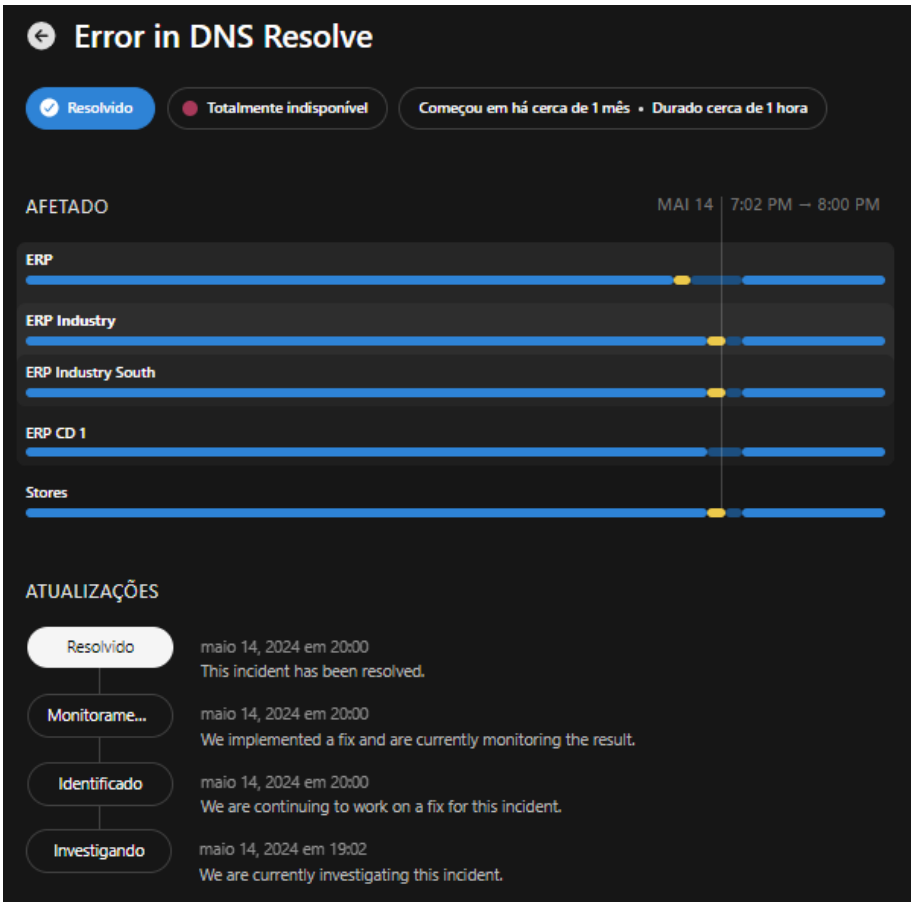


Fonte: Página de Status de Demonstração por INSTATUS

Os incidentes na plataforma são categorizados em diferentes estágios de resolução, tais como Identificando, Identificado, Monitorando e Resolvido. Da mesma forma, as manutenções seguem uma progressão de *status* que inclui Ainda Não Iniciado, Em Progresso e Concluído. Cada incidente ou manutenção registra seu progresso de resolução e histórico de atualizações

detalhadas. Ao acessar as informações de um incidente específico, a plataforma fornece uma lista dos serviços afetados, a duração do impacto e um resumo de quando e por quanto tempo o serviço ficou indisponível, como demonstrado na Figura 16. Essas informações são cruciais para entender o impacto operacional e tomar medidas corretivas de forma eficaz.

Figura 16 – Instatus - Informações do Incidente



Fonte:

A implementação da página de *status* é facilitada por ser uma solução baseada em nuvem, o que elimina a necessidade de instalações complexas e manutenção de hardware dedicado. Usuários podem simplesmente se registrar e configurar sua página de *status* através de uma plataforma administrativa online, que serve como o centro de controle para todas as operações relacionadas ao monitoramento de serviços. Esta abordagem baseada em nuvem não só economiza tempo e recursos, mas também garante que a plataforma possa ser acessada de qualquer lugar, oferecendo flexibilidade e conveniência para as equipes de TI.

Dentro desta plataforma administrativa, os usuários têm a capacidade de configurar integrações oficiais com facilidade. A integração nativa com o Datadog permite a incorporação automática de incidentes e o monitoramento de métricas, facilitando uma visão abrangente do desempenho dos serviços em tempo real. Para sistemas como Zabbix e plataformas de chamado, as integrações precisam ser configuradas por meio de *webhooks*, permitindo uma comunicação eficiente e personalizada entre sistemas. Essas configurações garantem que as operações de

monitoramento sejam robustas e totalmente adaptadas às necessidades específicas da infraestrutura de TI da organização, maximizando a eficácia da página de *status* na gestão e na resposta a incidentes.

5 PROPOSTA DE SOLUÇÃO

Neste capítulo, é apresentada a proposta de solução para o monitoramento de serviços e ativos de TI na Empresa X. A solução visa identificar, testar e selecionar a plataforma que melhor atende aos requisitos da organização, com foco em eficiência e integração com os sistemas existentes. Na Seção 5.1, são abordados os Requisitos da Solução, que incluem critérios de integração com sistemas de monitoramento, subscrição de atualização e visualização de métricas. A solução proposta deve comunicar-se com plataformas como Zabbix e Datadog, permitindo atualizações automáticas e agendamento de manutenções.

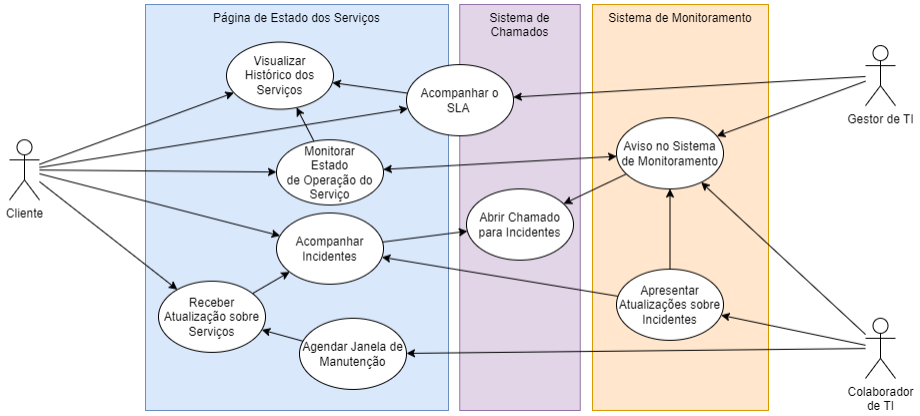
Em seguida, a Seção 5.2 apresenta a Escolha da Solução, onde é utilizado o Processo Hierárquico Analítico (AHP) para comparar alternativas e demonstrar quantitativamente qual solução melhor atende às demandas. A análise dos critérios levou à seleção da plataforma *Statuspage*, da *Atlassian*, como a mais compatível com os requisitos da empresa. Na Seção 5.3, são discutidas as Propostas de Soluções para Monitoramento, destacando as provas de conceito (PoCs) realizadas com *Statuspage* e *Instatus*. A Seção 5.3.1 detalha a Implementação da *Statuspage*, onde configurações de integração e mapeamento de ativos foram conduzidas para permitir atualizações automáticas e alertas em tempo real.

5.1 REQUISITOS DA SOLUÇÃO

Para atender às demandas da Empresa X, considerando seus serviços e integrações, a solução precisa atender aos requisitos de um sistema robusto de integração, customização e notificação, como demonstrado no diagrama da Figura 17. O monitoramento do departamento de TI está centralizado na utilização da plataforma de agregação de eventos *Datadog*, que monitora métricas, eventos e incidentes. Para o monitoramento dos serviços, os ativos são gerenciados pela plataforma *Zabbix*, que registra todas as suas métricas e outras configurações. Ambas as plataformas possuem APIs configuradas, permitindo a customização e a integração com outras ferramentas. Portanto, a solução de monitoramento deve ter a capacidade de comunicação por API para coleta de métricas e a possibilidade de ser atualizada por outras ferramentas.

Uma das principais funcionalidades previstas para essa plataforma é a reportagem de incidentes e seu acompanhamento. Para isso, é necessário poder iniciar e atualizar incidentes, tanto manualmente quanto baseados nas métricas coletadas. Esses incidentes devem estar integrados com o sistema de chamados do departamento de TI, para não depender da identificação do problema por parte do cliente, permitindo que todos os usuários afetados possam acompanhar o progresso da correção e outras atualizações. Isso reduz chamados redundantes e permite fácil acesso a informações e atualizações. Uma subcategoria de incidentes é o agendamento de manutenções programadas, avisando com antecedência o cliente sobre a indisponibilidade.

Figura 17 – Diagrama de Caso de Uso da Plataforma



Fonte: Elaborado pelo autor (2024)

O sistema de alertas e subscrições é crítico para avisar automaticamente gestores e clientes sobre problemas ou falhas. A subscrição deve permitir que os clientes configurem quais serviços desejam monitorar, sendo notificados apenas sobre os serviços que os afetam. Caso um incidente ocorra, o cliente pode se inscrever para receber atualizações. Se um serviço tiver seu desempenho degradado ou ficar indisponível, os usuários inscritos receberão atualizações sobre o evento. Essas notificações podem ser enviadas por vários canais de comunicação, sendo o principal o *e-mail*, seguido por integrações com sistemas de mensagens ou outras plataformas.

Além disso, a disponibilização do histórico de operação dos serviços e seus grupos é importante para a observabilidade do ambiente. Esse histórico permite que a equipe de TI analise tendências, identifique padrões e previna futuros incidentes. A solução deve oferecer ferramentas para a visualização de dados históricos de forma clara e compreensível, possibilitando uma melhor gestão dos recursos e serviços da empresa.

5.1.1 DESENVOLVIMENTO INTERNO

O desenvolvimento interno de uma plataforma para monitoramento apresenta diversos benefícios, incluindo o controle total sobre a configuração e personalização do sistema. Sendo desenvolvido pelo departamento de TI, a solução pode ser ajustada para atender a todos os requisitos específicos da empresa, de acordo com a demanda. Isso permite uma maior flexibilidade e adaptabilidade às necessidades internas, garantindo que a plataforma funcione exatamente como esperado e possa ser modificada conforme as necessidades mudam.

Contudo, desenvolver uma plataforma completamente do zero pode resultar em uma entrega mais demorada e custosa em comparação com soluções prontas disponíveis no mercado. Este processo exige um departamento de desenvolvimento maduro e bem estruturado, capaz de lidar com os desafios técnicos e operacionais. Além disso, o planejamento, desenvolvimento e implantação de uma solução interna podem não ser viáveis devido aos altos custos e ao tempo necessária para sua conclusão. A manutenção e melhoria contínua da plataforma também podem

se tornar um fardo significativo a longo prazo, não atendendo às perspectivas de eficiência e economia da Empresa X. Portanto, considerando esses fatores, a proposta de desenvolvimento interno é descartada como uma solução válida.

5.2 ESCOLHA DE SOLUÇÃO

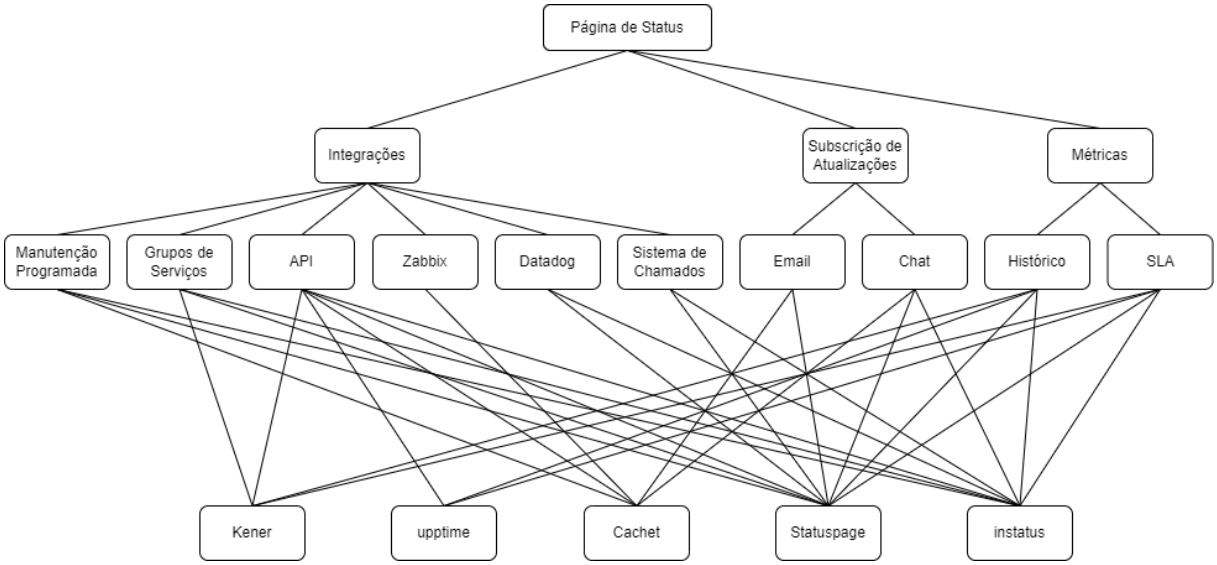
Neste capítulo, é apresentado a proposta de solução para o monitoramento de serviços e ativos na área de TI. A solução é desenvolvida com o objetivo de integrar diversas ferramentas e tecnologias para garantir maior eficiência e confiabilidade no processo de monitoramento. Os requisitos propostos não possuem o mesmo peso em sua premissa; por exemplo, integrações com os sistemas de monitoramento e de chamados são as demandas mais importantes a serem atendidas. Para isso, foi adotada a técnica de AHP para demonstrar quantitativamente qual das soluções melhor atende os requisitos da Empresa X.

O modelo AHP pode ser iniciado pela elaboração de um diagrama que apresente os critérios, alternativas e o objetivo a ser alcançado. A estrutura do diagrama começa com a definição clara do objetivo principal, que se encontra no topo da hierarquia. A partir desse objetivo, são apresentados os critérios que devem ser atendidos para alcançá-lo. Esses critérios podem ser divididos em grupos e subgrupos, operando como demandas e requisitos específicos que, juntos, contribuem para o alcance do objetivo ideal. Essa divisão é organizada em uma estrutura de árvore invertida, onde cada nível subsequente representa uma desagregação mais detalhada dos requisitos até alcançar sua forma mais atômica. Na base da árvore, são listadas as alternativas de soluções, que são conectadas aos critérios correspondentes, indicando quais critérios cada alternativa atende. Esse diagrama proporciona uma visão clara e estruturada dos elementos envolvidos no processo de decisão, facilitando a análise e comparação das diferentes soluções.

A análise do AHP consiste na atribuição de pesos em pares entre requisitos, avaliando a importância relativa de cada um. Essa análise retorna um peso de importância para cada requisito. Em seguida, são atribuídos valores aos requisitos atendidos por cada solução, demonstrando o quão bem cada requisito é cumprido. Esses valores são combinados com os pesos dos requisitos, e uma porcentagem é calculada para definir quanto cada solução atende às demandas. Essa abordagem permite uma análise objetiva e quantitativa, facilitando a escolha da melhor solução. Para mais informações e cálculos utilizados na determinação da AHP, favor referenciar o Apêndice A.

Os requisitos da solução podem ser divididos em três grupos principais: Integrações, Subscrição de Atualização e Métricas. As integrações incluem a comunicação e configuração com ferramentas externas para automação de processos e organização dos serviços. Isso envolve a comunicação com plataformas de monitoramento, como *Zabbix* e *Datadog*, que permitem a atualização automática do estado de operação dos serviços ou ativos, a iniciação dinâmica de incidentes com base em métricas externas e o agendamento de manutenções que afetam os

Figura 18 – Diagrama do Processo Hierárquico Analítico



Fonte: Elaborado pelo autor (2024)

serviços. A integração com a plataforma de chamados para atualizações de incidentes ao cliente é crucial e pode ser configurada diretamente na solução ou gerenciada por ferramentas externas utilizando API. Portanto, este grupo abrange integrações com *Zabbix*, *Datadog* e o Sistema de Chamados, destacando a importância de uma API para comunicação e configuração externa. A manutenção programada é fundamental para a visibilidade do cliente e a organização do departamento de TI, enquanto grupos de serviços são desejáveis para organização, mas não essenciais, especialmente se não houver histórico resumido dos serviços.

O critério de subscrição de atualizações enfoca a comunicação e observabilidade disponibilizada ao cliente, visando oferecer meios para que o cliente monitore incidentes e anomalias no sistema por meio de avisos, reduzindo a identificação reativa de problemas. O principal critério deste grupo é a comunicação por *e-mail*, permitindo que o cliente acompanhe o ambiente completo, apenas os serviços que o afetam, ou atualizações de incidentes e manutenções. A configuração de *webhooks* para automações externas é importante, especialmente para o departamento deTI da Empresa X e para os setores de TI das empresas do grupo, embora essa demanda tenha sido integrada ao critério de API no grupo de integrações. Anúncios automáticos em grupos de sistemas de comunicação também são uma opção requisitada, ajudando a evitar acúmulos de *e-mails* redundantes e informando grandes grupos de colaboradores e clientes. Portanto, este grupo de critérios inclui a configuração de subscrição e alertas por *e-mail*, essenciais para o objetivo da plataforma, e a comunicação e alertas por sistemas de mensagens, uma opção adicional muito útil.

Para aumentar a observabilidade do ambiente e dos serviços, o acompanhamento de métricas e estatísticas é essencial. O grupo de critérios de Métricas abrange demandas para o monitoramento do estado de operação dos serviços e seus ambientes, assegurando que cumpram

os acordos de disponibilidade com o cliente. Este grupo inclui o requisito de histórico dos serviços e incidentes, permitindo identificar padrões ou sistemas falhos e visando a criação de um sistema mais estável e robusto. Também é necessário o acompanhamento das SLAs dos serviços para validar o cumprimento dos acordos e melhorar a qualidade do atendimento ao cliente.

Com os grupos, seus critérios e suas importâncias definidos, é possível quantificar essas relações e determinar sua prioridade. A Tabela 1 apresenta a comparação por pares (*Pairwise comparison*) dos critérios, onde a linha possui uma importância tanto maior que a coluna. Como as demandas possuem o mesmo peso que elas mesmas, a diagonal principal é unitária, e outras células são populadas pela importância relativa de um requisito (linha) em relação a outro (coluna), quantificando a relação. Por exemplo, o valor na coluna três e linha dois demonstra que a Manutenção Programada é de grande importância quando comparada aos Grupos de serviços, gerando a relação 5/1.

Tabela 1 – Comparação por Pares dos Critérios

	Manut.	Grupos	API	Zabbix	Datadog	Chamado	Email	Chat	Histórico	SLA
Manut.	1	5/1	2/1	1/3	1/3	1/3	1/2	2/1	3/1	2/1
Grupos	1/5	1	1/3	1/5	1/5	1/5	1/3	1/2	1/3	1/2
API	1/2	3/1	1	1/2	1/2	1/2	1/2	3/1	1/2	3/1
Zabbix	3/1	5/1	2/1	1	1	1	3/1	5/1	2/1	2/1
Datadog	3/1	5/1	2/1	1	1	1	3/1	5/1	2/1	2/1
Chamado	3/1	5/1	2/1	1	1	1	3/1	5/1	2/1	2/1
Email	2/1	3/1	2/1	1/3	1/3	1/3	1	5/1	5/1	7/1
Chat	1/2	2/1	1/3	1/5	1/5	1/5	1/5	1	1/4	1/3
Histórico	1/3	3/1	2/1	1/2	1/2	1/2	1/5	4/1	1	2/1
SLA	2/1	2/1	1/3	1/2	1/2	1/2	1/7	3/1	1/2	1

Fonte: Elaborado pelo autor (2024)

Com as relações estabelecidas, é possível determinar as prioridades dos critérios. Na Tabela 2, os critérios estão ordenados em ordem crescente de importância. Observa-se que a integração com o sistema de monitoramento, o sistema de chamados e o sistema de alertas para o cliente possuem a mesma prioridade e são de extrema importância para a solução. A seguir, a Manutenção Programada destaca-se como necessária para melhorar a comunicação e organização do departamento de TI e a interação com o cliente. Em continuidade, o histórico e as SLAs para análise de entrega e sustentação dos serviços são considerados relevantes na escolha.

Os critérios restantes possuem menor importância, não afetando significativamente as soluções, mas oferecendo um diferencial. A API é relevante e útil para comunicação com *softwares* externos. No entanto, as principais integrações são critérios próprios, deixando essa opção principalmente para automações e outros casos de uso, alinhado com o objetivo da Empresa X de minimizar o desenvolvimento interno de integrações devido aos custos e tempo associados.

Tabela 2 – Prioridade dos Critérios

Critério	Prioridade
Zabbix	16,4%
Datadog	16,4%
Sistema de Chamados	16,4%
Email	14,0%
Manutenção Programada	9,1%
Histórico	8,1%
API	7,6%
SLA	6,6%
Chat	3,0%
Grupo de Serviços	2,6%

Fonte: Elaborado pelo autor (2024)

A partir das análises das soluções nas seções anteriores, foram definidos os critérios que cada alternativa deve atender. Na Tabela 3, são transcritas as ligações apresentadas na Figura 18, que mostram como cada alternativa atende a esses critérios. A tabela permite uma visualização clara de como cada solução proposta se alinha com os critérios estabelecidos, facilitando a avaliação comparativa e a escolha da melhor opção. Dessa forma, fica evidente quais alternativas cumprem mais adequadamente os requisitos prioritários, oferecendo uma base sólida para a tomada de decisão.

Tabela 3 – Critérios atendidos pelas Alternativas.

	upptime	Kener	Cachet	Statuspage	instatus
Manutenção Programada	1	0	1	1	1
Grupo de Serviços	0	1	1	1	1
API	1	1	1	1	1
Zabbix	0	0	1	0	0
Datadog	0	0	0	1	1
Sistema de Chamados	0	0	0	1	0
Email	1	0	1	1	1
Chat	0	0	0	0	1
Histórico	1	1	0	1	1
SLA	1	1	0	1	1

Fonte: Elaborado pelo autor (2024)

Com a importância dos critérios calculada e o atendimento deles pelas alternativas listado, é possível calcular e apresentar, na Tabela 4, a porcentagem de atendimento das soluções propostas em relação às demandas da Empresa X para a plataforma de monitoramento de ativos para o cliente. Analisando os resultados, a solução *Statuspage* da *Atlassian* é a que mais atende aos critérios, com uma pontuação de 80%, seguida pelo *Instatus*, com 67%. Outras alternativas serão desconsideradas por ficarem abaixo de 50%.

Portanto, ao utilizar o modelo de Processo Hierárquico Analítico para determinar a importância dos critérios e quantificar o escopo das alternativas, as opções para a escolha de uma solução de plataforma de monitoramento foram reduzidas a duas: *Statuspage* e *Instatus*. Ambas

Tabela 4 – Resultado do Modelo AHP.

Alternativo	Pontuação
Statuspage	80,64%
instatus	67,28%
Cachet	49,60%
upptime	45,32%
Kener	24,89%

Fonte: Elaborado pelo autor (2024)

são soluções em nuvem oferecidas por empresas externas no modelo de serviço, cada uma com seus próprios SLAs.

5.3 PROPOSTA DE SOLUÇÕES PARA MONITORAMENTO DE SERVIÇOS

A escolha de uma solução não foi realizada pela gerência na Empresa X, por esse fato, serão realizados provas de conceitos do uso das plataformas. Esses testes consistem na implementação temporária das soluções no ambiente, em detalhes na Seção 5.3.1 e Seção 5.3.2, integrando os principais serviços, e testando os principais casos de uso. Ambos alternativos oferecem licenças de testes sem custo, sendo a limitação o oferecimento de recursos avançados. A escolha da solução a ser implementada ainda não está definida, mais informações na Seção 5.5.

5.3.1 IMPLEMENTAÇÃO DA PLATAFORMA DO STATUSPAGE

A criação inicial da plataforma é simples e direta. Ao acessar o *site* com uma conta autenticada, pode-se iniciar a criação de uma página. As configurações iniciais incluem opções visuais, definição dos objetos de tela e métricas. Esses passos garantem que a página seja personalizada de acordo com as necessidades específicas do usuário, proporcionando uma interface intuitiva e eficaz. Este processo inicial é projetado para ser acessível, mesmo para aqueles com pouca experiência técnica.

A plataforma não fornece dados ou métricas de demonstração, oferecendo apenas integrações prontas de serviços de terceiros. Para a prova de conceito, é necessário realizar uma integração parcial com o ambiente, migrando os principais objetos e aqueles utilizados para testes. No monitoramento de ativos no servidor *Zabbix*, os ativos devem ser mapeados para os serviços que possuem, incluindo servidores, *ILOs*, *firewalls*, *Switches* e *storages*. Para cada *host* será mapeado por variáveis o identificador do serviço atrelado, essa informação será utilizada em ações acionadas pelas *triggers* disparadas, enviando um pacote a API do *Statuspage*, que atualizará os componentes relacionados, e em caso de problemas altos ou desastres, também é iniciado um incidente.

O monitoramento dos eventos dos serviços requer apenas a atualização dos monitoramentos e alertas configurados, para refletir o estado atual das informações. Incidentes serão iniciados por chamadas da API como resposta aos monitoramentos configurados, e na resolução do problema, uma resposta de encerramento automático será enviada. Essa configuração para incidente é manual, e serão avaliados os principais serviços que demandam, sendo integrado no procedimento de criação e manutenção dos monitoramentos do *Datadog*. A plataforma possui a função de *Downtimes*, que são manutenções programadas, elas podem ser integradas por API para serem criadas na plataforma de monitoramento.

Tanto o agendamento de manutenções programadas, quanto incidentes em progresso, podem ser gerenciados pela plataforma de administração do *Statuspage* quanto pelo *Zabbix* ou *Datadog*. Utilizando a configuração de *webhooks*, é possível realizar essa integração, como a atualização de mensagem ou avisos nos serviços de monitoramento, baseada em resposta na plataforma. Por limitado acesso à documentação, e somente estar disponível para planos empresariais, não foi validado o uso da plataforma de chamados como um gerenciador de alertas. Nessa estrutura, os serviços de monitoramento, ao identificar um problema ou anomalia, iniciam um chamado, sendo as atualizações de informação e progresso do incidente atualizadas apenas nele, e replicadas para as outras plataformas. Nesse formato, os serviços de monitoramento indicam o início da anomalia, o chamado acompanha e atualiza informações até sua conclusão, centralizando o local para atualizações e informações.

A plataforma permite o agrupamento de serviços com até um nível de profundidade, apresentando o histórico e os SLAs resumidos do grupo. Isso facilita a gestão e a visualização do desempenho e do histórico de serviços, assegurando uma administração eficiente e organizada. Com essas funcionalidades, a plataforma não só melhora a visibilidade do status dos serviços, mas também oferece ferramentas para uma gestão proativa e informada.

5.3.2 IMPLEMENTAÇÃO DA PLATAFORMA DO INSTATUS

Se tratando de uma plataforma oferecida como serviço, o processo de configuração é simplificado e acessível, sendo necessário apenas a criação de uma conta no *site* da ferramenta, permitindo o início imediato das configurações e personalizações. A configuração inicial dedica-se principalmente à aparência da plataforma, incluindo a criação e personalização dos objetos iniciais necessários para o funcionamento do sistema. Além disso, as métricas do sistema de agregação de eventos dos serviços são configuradas de maneira eficiente através do vínculo da chave da API e alguns parâmetros específicos. Este processo inicial garante que a plataforma esteja adaptada às necessidades do usuário desde o começo, promovendo uma experiência de uso otimizada e integrada.

A plataforma não oferece dados ou métricas de demonstração, somente integrações prontas de serviços de terceiros. Para a prova de conceito é necessário realizar a integração parcial com o ambiente, migrando os principais objetos e objetos utilizados para testes. No mo-

nitoramento de ativos no servidor *Zabbix*, deve ser mapeado os ativos com os serviços que eles afetam, incluindo servidores, *switches*, *storages*, O mapeamento será realizado por variáveis (macros) nos *hosts*, indicando o componente relacionado e sendo utilizado em eventos ou problemas. A replicação de incidentes e manutenções é feito por requisições na API da plataforma, sendo problemas iniciado es replicado do *Zabbix* para a plataforma de monitoramento por ações nas *triggers*, com o *webhook* configurado para atualizações em ambos os lados.

Para os serviços de agregação de eventos e métricas, é configurado de forma similar utilizando chamados API e *webhooks*. Os monitoramentos são configurados para reportar as mensagens a plataforma de monitoramento, e atualizado de acordo, até o envio de mensagem de estado restaurado. A plataforma do *Datadog* possui a função para manutenção de serviços, chamado de *Downtime*, porém essa opção não está sendo utilizada no fluxo de processo e uso da ferramenta, contudo, ela opera similar ao sistema de monitoramento, baseado em mensagem e *webhooks*.

O sistema de chamados pode ser integrado pela configuração de um *webhook* no sistema de subscrição da plataforma, enviando requisição de acordo com novos incidentes abertos. No sistema de chamado, é configurado para atualizar esses incidentes pelo uso da API. Uma alternativa mais simples é migrar a gerência de incidentes e chamados para a plataforma do *Zabbix*. Por ele, é possível atualizar tanto o sistema de chamado quanto o *Instatus* por API para problemas relacionados com os ativos, para serviços, é necessária a mesma configuração na *Datadog*. Outra opção para essa configuração é utilizar o sistema de chamados como plataforma de gerenciamento de alertas, alterando o fluxo para os serviços de monitoramento iniciar um chamado; porém, as atualizações e progresso permanecem centralizados no chamado, sendo replicados para as outras plataformas.

Os grupos de serviços não possuem limite de profundidade para sua organização e apresentam um histórico resumido de seus componentes, com exceção de serviços de terceiros, que não possuem históricos. Dessa forma, a flexibilidade da plataforma permite uma adaptação rápida e eficiente às necessidades específicas de cada usuário, garantindo uma gestão eficaz dos serviços e incidentes.

5.4 TREINAMENTO

O propósito dessa plataforma é oferecer uso e visibilidade ao cliente, tornando o treinamento de uso crucial. A Empresa X possui três níveis de suporte de TI. Membros do grupo possuem colaboradores locais próprios, para oferecer o suporte de nível um, auxiliando no uso diário e identificação de problemas. A partir da triagem básica, e o problema demandar conhecimento técnico das soluções, um chamado deve ser aberto a Empresa X, onde o setor de suporte nível dois opera, auxiliando com configurações, ajustes técnicos e dúvidas de uso avançado. Caso o problema demandar ajuste ou modificação dos serviços, ou liberação e auxílio dos

ativos, o chamado pode ser redirecionado ao suporte de nível três, sendo os departamentos de Desenvolvimento de Sistemas de TI ou de Infraestrutura e Operações de TI da Empresa X.

O cliente será treinado para uso da plataforma por meio da documentação e guias desenvolvidos internamento, e em caso de dúvidas mais especificas, deve contatar o suporte nível um, seguindo o fluxo de auxílio ao cliente, e escalando os chamados, de acordo com a demanda. Os operadores de TI do suporte de nível um, receberam treinamento remoto do uso da plataforma, realizando testes, simulações, aprendendo a configurar subscrição para alertas, e como navegar entres as páginas. O nível dois vai receber treinamento de uso avançado, como acesso a plataforma administrativa, aprendendo opções avançadas, visualização de histórico, verificar em que serviços o cliente está subscrito, e acesso ao sistema interno de monitoramento da TI, para validar ou identificar as informações presentes na plataforma. Também vai ser apresentado a estrutura utilizada no sistema de chamados, para a replicação entre as plataformas, e automações realizadas.

O nível três participara no treinamento do nível dois, e também vai revisar o fluxo de abertura do chamado até sua conclusão, realizando atualizações nos componentes, trocando o progresso do incidente. Considerando o objetivo da plataforma de monitoramento, o uso para o suporte de nível três não vai fazer parte do fluxo de resolução de chamados, não tendo acesos a plataforma administrativa, nem controle de componentes. A gestão da plataforma vai estar sobre o setor de Sustentação de Serviços de TI, com auxiliares em outros setores do suporte de nível três, para configuração inicial da plataforma, e acesso ao gestor do departamento de TI para configuração de perfis de acesso e designação de função.

5.5 AVALIAÇÃO

Com a definição de duas propostas que melhor atendem os requisitos avaliados, inicia-se a prova de conceitos no ambiente da Empresa X. Primeiramente, serão avaliadas a simplicidade de integração da plataforma com outros serviços e a identificação de possíveis incompatibilidades com o ambiente atualmente utilizado. Este processo inclui a revisão do fluxo de trabalho do departamento de TI, focando nos monitoramentos e métricas essenciais. Em seguida, será validada a facilidade de uso por parte do cliente, garantindo que a clareza dos componentes e sua influência no ambiente sejam compreensíveis. Além disso, os sistemas de comunicação e subscrição para alertas serão testados exaustivamente para assegurar sua eficácia. Durante o período de avaliação, todos os resultados das simulações e incidentes reais serão documentados e catalogados.

Após a fase de testes, os dados coletados serão apresentados à gerência da Empresa X. Esta apresentação visa determinar qual das soluções testadas melhor atende às necessidades e se adéqua ao grupo da empresa. A decisão final será baseada na análise detalhada dos resultados obtidos durante o período de avaliação. O método a ser utilizado para escolher a solução que

melhor agrega valor ao grupo da Empresa X ainda não está definido. Estão sendo avaliados diversos *frameworks* para tomada de decisão, incluindo DECIDE (PREECE; SHARP; ROGERS, 2013), SWOT (FINE, 2009), formulários de pesquisa, entre outros.

O *framework* DECIDE é uma ferramenta sistemática projetada para auxiliar na tomada de decisões complexas. Ele segue seis etapas principais: (1) Definir o problema; (2) Estabelecer os critérios; (3) Considerar as alternativas; (4) Identificar os melhores critérios; (5) Desenvolver a decisão; e (6) Avaliar a decisão. Cada etapa é cuidadosamente elaborada para garantir que todas as opções sejam consideradas e que a decisão final seja a mais informada possível, baseando-se em uma análise detalhada e criteriosa dos dados e das opções disponíveis. Seu uso pode ser demonstrado na escolha de um ambiente virtual de aprendizagem no trabalho "Análise de utilização de ambiente virtual de aprendizagem por estudantes e professores na educação básica"(FANTIN; NOTARI; SILVA, 2021).

O *framework* SWOT (*Strengths*, *Weaknesses*, *Opportunities*, *Threats*) é uma ferramenta essencial para a tomada de decisões estratégicas, permitindo que organizações identifiquem e avaliem seus pontos fortes e fracos internos, bem como as oportunidades e ameaças externas. Este método proporciona uma visão abrangente da situação atual da empresa, facilitando a elaboração de estratégias informadas e eficazes. Ao analisar os quatro componentes do SWOT, os tomadores de decisão podem maximizar os pontos fortes, minimizar as fraquezas, explorar oportunidades e se preparar para ameaças, promovendo assim um planejamento estratégico mais robusto e direcionado.

6 IMPLEMENTAÇÃO DA SOLUÇÃO

Neste capítulo é detalhado o processo de implementação da solução de monitoramento selecionada, enfatizando as etapas envolvidas para sua integração completa na infraestrutura de TI da Empresa X. Esse processo inclui desde o contato com fabricantes até a realização de provas de conceito e a estruturação final dos componentes essenciais da solução de monitoramento. Com isso, busca-se garantir que o sistema atenda aos requisitos de eficiência, escalabilidade e segurança previamente identificados.

O processo implementação é iniciado pelo contato com os fabricantes, Seção 6.1, que descreve as interações iniciais com os fornecedores das plataformas de monitoramento, discutindo características, suporte e customizações para adequação às necessidades da Empresa X. Em seguida, a prova de conceito, Seção 6.2 apresenta os testes controlados realizados para validar o desempenho e a confiabilidade das soluções em cenários simulados, o que contribuiu para a escolha final da plataforma. Por fim, a estrutura da implementação, Seção 6.3 detalha a arquitetura técnica e a configuração da solução escolhida, abordando a integração com a infraestrutura existente e as medidas de segurança para garantir o monitoramento eficiente e seguro dos ativos e serviços de TI.

6.1 CONTATO COM FABRICANTES

A escolha da plataforma de monitoramento adequada é uma etapa crucial para garantir o desempenho, a segurança e a disponibilidade de serviços e ativos em ambientes de TI. Nesse contexto, a análise das características técnicas de diferentes soluções de monitoramento, aliada ao suporte fornecido pelos fabricantes, torna-se essencial para a tomada de decisão informada. Os fabricantes não apenas oferecem as ferramentas e funcionalidades necessárias, mas também desempenham um papel fundamental no fornecimento de suporte técnico, atualizações de software e treinamentos, contribuindo diretamente para o sucesso da implementação.

Nesta seção, serão apresentados os principais canais de contato disponibilizados pelos fabricantes das plataformas de monitoramento analisadas. Informações sobre suporte técnico, horários de atendimento e documentações oficiais serão abordadas para facilitar o acesso a recursos essenciais e garantir a comunicação eficiente entre os usuários e as equipes responsáveis pelo desenvolvimento e manutenção das plataformas. Esse contato direto com os fabricantes pode ser determinante para resolver problemas críticos e implementar melhorias contínuas nas operações de monitoramento.

Durante o processo de testes das plataformas propostas, a pesquisa por outras opções e soluções de monitoramento foi mantida de forma contínua. Esse esforço adicional teve como objetivo garantir uma cobertura abrangente das alternativas disponíveis no mercado, possibi-

litando uma comparação mais completa entre as ferramentas e verificando se outras soluções poderiam melhor atender aos requisitos técnicos e funcionais estabelecidos. Essa abordagem permitiu uma avaliação criteriosa de novas ferramentas que surgiram ou de opções que inicialmente não haviam sido consideradas, mas que poderiam agregar valor ao projeto.

Além disso, foi realizado um levantamento detalhado sobre a capacidade das novas soluções em cumprir os requisitos necessários, como a escalabilidade, compatibilidade com diferentes ambientes de TI e facilidade de integração com sistemas legados. Através dessa análise, buscou-se identificar plataformas que pudessem oferecer diferenciais competitivos, tanto em termos de funcionalidades quanto de custo-benefício, garantindo que a solução final atendesse plenamente às demandas de monitoramento, suporte e flexibilidade exigidas pelo ambiente em que seria implementada.

Por tanto, na sequência serão abordadas três soluções de monitoramento que se destacam no mercado atual: *Instatus* (Seção 6.1.1), *Incident.io* (Seção 6.1.2) e *Statuspage* (Seção 6.1.3). Cada uma dessas plataformas apresenta características distintas que podem ser vantajosas dependendo das necessidades específicas de monitoramento e suporte de diferentes organizações. A seguir, serão exploradas as funcionalidades e vantagens de cada uma dessas opções, com foco em como elas atendem aos requisitos de escalabilidade, integração e suporte contínuo, aspectos essenciais para garantir a eficácia do monitoramento em ambientes de TI.

6.1.1 INSTATUS

O contato inicial com o fabricante da plataforma de monitoramento foi realizado por meio de *e-mails* enviados ao suporte oficial disponibilizado pela empresa. O objetivo desse contato era obter informações detalhadas sobre as opções de implementação e o suporte técnico oferecido no Brasil, uma vez que a escolha de parceiros de implementação e a proximidade do suporte podem influenciar significativamente a adoção de uma nova tecnologia. A comunicação inicial ocorreu de maneira adequada, e as primeiras respostas atenderam a algumas das dúvidas levantadas, proporcionando uma visão preliminar sobre as operações da plataforma.

Durante as interações, questões específicas foram levantadas, incluindo as políticas de segurança aplicadas pela plataforma no tratamento e proteção dos dados dos clientes. Este ponto é crítico, especialmente considerando a necessidade de conformidade com regulamentações como a Lei Geral de Proteção de Dados (LGPD). Além disso, foi solicitado um orçamento detalhado para o plano executivo da plataforma, com o intuito de avaliar a relação custo-benefício e compreender as vantagens associadas à contratação desse serviço. No entanto, as respostas fornecidas foram incompletas e pouco esclarecedoras, o que dificultou a obtenção de uma visão clara sobre a solução.

À medida que o contato evoluiu, surgiram dificuldades significativas na comunicação. A troca de informações foi inicialmente conduzida por um representante de pré-vendas, mas

posteriormente foi redirecionada para o suporte geral da empresa, tornando o atendimento mais lento e menos eficiente. A mudança no ponto de contato resultou em respostas inconsistentes, e muitas questões críticas continuaram sem solução. Esse cenário causou insegurança em relação à qualidade do suporte e ao nível de comprometimento do fabricante em atender às necessidades da Empresa X.

Além disso, ficou evidente que o fabricante não possuía parceiros implementadores no Brasil e oferecia suporte técnico limitado, restrito a canais automatizados e *e-mails* de atendimento. Essa falta de suporte local foi um fator decisivo na avaliação da plataforma, pois comprometeu a confiança na capacidade do fabricante de oferecer assistência adequada durante e após a implementação da solução.

Por fim, o baixo nível de interesse demonstrado pelo fabricante em apresentar sua solução de maneira proativa, somado à dificuldade de comunicação e à qualidade insatisfatória do atendimento, gerou insegurança quanto ao suporte futuro. Diante dessas limitações, a plataforma foi desconsiderada como uma opção viável para o projeto, reforçando a importância de soluções que ofereçam suporte técnico eficiente, atendimento responsivo e parceiros locais para garantir uma implementação bem-sucedida.

6.1.2 INCIDENT.IO

A plataforma *Incident.io* é uma solução relativamente nova no mercado de monitoramento, lançada em 2019. Ela se destaca por oferecer uma gama de funcionalidades avançadas e opções de configuração que prometem otimizar o gerenciamento de incidentes em ambientes corporativos. A descoberta da plataforma ocorreu durante a análise de casos de clientes listados no *site* da *Statuspage* da *Atlassian*, onde a um caso foi removido, e depois de pesquisa, foi identificado que o cliente trocou a plataforma pela a do *Incident.io* . Esse fator despertou interesse na investigação da viabilidade de uso da ferramenta, especialmente por seu foco em automação e integração com sistemas de comunicação modernos, características essenciais para a otimização do monitoramento em tempo real.

Para iniciar os testes com a *Incident.io*, foi necessário criar uma conta de demonstração. No entanto, a plataforma exigia o uso de um *e-mail* corporativo e uma licença ativa do *Slack*, o que impediu a realização de testes anônimos e livres. Essa abordagem visa garantir a segurança e o controle de acesso aos testes, mas também restringe o acesso a empresas que já utilizam ferramentas específicas de comunicação. Devido à falta de mais informações públicas detalhadas sobre a plataforma no *site*, um contato direto por *e-mail* foi estabelecido com a equipe de suporte da *Incident.io* para esclarecer dúvidas e solicitar uma apresentação personalizada da solução.

Após alguns *e-mails* trocados, foi agendada uma apresentação virtual da plataforma, na qual foram detalhadas suas funcionalidades e esclarecidas as dúvidas da Empresa X. Durante

a demonstração, foi destacada a necessidade de um ambiente ativo no *Slack* para que as automações e integrações oferecidas pela *Incident.io* pudessem funcionar de maneira eficiente. A solução depende diretamente dessa ferramenta de comunicação para a execução de suas principais funções, como o monitoramento de incidentes e a coordenação das equipes envolvidas na resolução. Esse requisito pode limitar a adoção da plataforma por empresas que não utilizam o *Slack* como parte de seu ecossistema de comunicação.

Além do *Slack*, foi mencionada a possibilidade de integração com o *Microsoft Teams* como uma alternativa, o que ampliaria a variedade de opções para empresas que utilizam essa ferramenta. No entanto, a Empresa X não faz uso de nenhuma dessas duas plataformas de mensagens, o que representa um obstáculo significativo para a adoção da *Incident.io*. A dependência dessas ferramentas externas torna a solução inadequada para organizações que operam com diferentes sistemas de comunicação ou que não estão dispostas a adaptar seus fluxos de trabalho a essas plataformas específicas.

Por fim, um ponto adicional que influenciou na decisão de não adotar a *Incident.io* foi a ausência de suporte técnico nativo no Brasil. Embora a solução seja promissora e apresente características inovadoras, a falta de suporte local dificulta a resolução de problemas e a implementação personalizada para o mercado brasileiro. Assim, apesar de seu potencial, a *Incident.io* não se alinhava com as necessidades e o ambiente de trabalho da Empresa X, levando à sua exclusão do processo de seleção de plataformas de monitoramento.

6.1.3 STATUSPAGE

A Empresa X já possui implementações anteriores de outras soluções da *Atlassian*, o que motivou a escolha pela mesma implementadora para a análise e possível adoção da *Statuspage*. O contato foi feito diretamente com eles, que facilitou a comunicação e possibilitou uma integração mais fluida com as soluções já existentes. A familiaridade prévia com o ecossistema da *Atlassian* contribuiu para que o processo fosse conduzido com mais eficiência, reduzindo o tempo necessário para entendimento e configuração inicial.

Durante a interação com a equipe da *Statuspage*, a plataforma foi ativada para que uma demonstração prática pudesse ser realizada. A implementadora apresentou as principais funcionalidades do serviço, como a criação de páginas de *status*, o gerenciamento de incidentes e as opções de notificações para os usuários. Além disso, foi explorada a capacidade de integração com outras ferramentas de comunicação e monitoramento já utilizadas pela Empresa X, como forma de proporcionar uma visão unificada e mais eficiente dos *status* dos serviços.

Foi também discutida a estrutura de preços e os benefícios de cada um dos planos disponíveis. A *Atlassian* oferece uma gama de opções que variam conforme o número de serviços monitorados e a complexidade das necessidades de comunicação. Durante a conversa, foram abordados pontos como suporte técnico, opções de customização e integração com outras so-

luções de TI. A equipe se mostrou flexível e disposta a adaptar a oferta conforme as demandas específicas da Empresa X.

O contato com a *Statuspage* foi facilitado pela integração já existente de outras soluções da *Atlassian* na infraestrutura e sustentação da Empresa X. Isso possibilitou que a demonstração fosse realizada de forma mais rápida e que o valor da solução fosse evidenciado dentro de um contexto já familiar. A experiência foi positiva, destacando a importância de contar com um fornecedor único para diversas soluções, promovendo um ecossistema mais coeso e otimizado.

6.2 PROVA DE CONCEITO

A escolha das soluções para a Prova de Conceito (PoC) envolveu uma análise criteriosa de diferentes fatores técnicos e operacionais. Durante esse processo, surgiram desafios como a dificuldade de comunicação com algumas fabricantes, a ausência de suporte técnico adequado e a incompatibilidade de certas plataformas com o ambiente já implementado na Empresa X. Tais dificuldades poderiam causar atrasos significativos no cronograma estabelecido para a entrega da PoC. Dessa forma, optou-se por focar inicialmente na solução da *Statuspage* da *Atlassian*, visando minimizar riscos e garantir que os prazos sejam cumpridos.

A *Statuspage* foi selecionada como a primeira opção por apresentar uma arquitetura compatível com os requisitos técnicos da Empresa X e por se integrar de maneira simplificada com o ambiente de TI já existente. Além disso, a solução possui uma base consolidada no mercado, sendo amplamente utilizada por empresas de diversos setores para a comunicação de incidentes e *status* de serviços. A experiência da *Atlassian* em fornecer ferramentas robustas e confiáveis contribuiu para que a decisão fosse tomada com mais segurança, reduzindo o tempo gasto com testes preliminares e ajustes.

Outro fator decisivo para a escolha da *Statuspage* foi a confiança gerada pela sua adoção por grandes empresas de referência no mercado. Essa validação por parte de organizações que lidam com altos volumes de tráfego e exigem comunicação de incidentes em tempo real reforçou a ideia de que a solução é capaz de atender a demandas complexas de maneira eficiente. Ao seguir essa estratégia, a Empresa X se posiciona para validar rapidamente a funcionalidade e a eficácia da ferramenta em um ambiente de produção, permitindo uma tomada de decisão mais informada para futuras implementações.

A flexibilidade da *Statuspage* e a experiência prévia da equipe da *Atlassian* no suporte à Empresa X garantiram um início mais ágil para a prova de conceito. Esse enfoque inicial possibilitará um melhor entendimento sobre como a solução pode ser expandida e adaptada às necessidades específicas de comunicação e monitoramento da Empresa X. A escolha visa não apenas cumprir os prazos estabelecidos, mas também proporcionar um caminho claro para futuras evoluções, permitindo comparações com outras soluções à medida que a PoC avança e novas funcionalidades são avaliadas.

6.3 ESTRUTURA DE IMPLEMENTAÇÃO

A configuração inicial da plataforma *Statuspage* foi realizada com base nas informações específicas da Empresa X, assegurando que a solução refletisse sua identidade visual e facilitasse a comunicação com usuários internos e externos. Personalizações incluíram a URL do domínio utilizado para acessar a página de *status*, além da adição de *banners* e ícones representativos da marca. Também foram incorporados *links* para suporte e documentação técnica, promovendo uma navegação intuitiva e voltada para a resolução de problemas. No entanto, foram observadas limitações na configuração nativa dos *e-mails* de subscrição, como a restrição no tamanho dos nomes de componentes e grupos, que eram truncados com reticências, gerando confusão para o cliente. Como alternativa, pode-se adotar a customização de *CSS* e *HTML*, embora isso adicione complexidade e demanda de trabalho extra.

Os componentes na *Statuspage* foram organizados para abranger todos os principais serviços críticos da Empresa X. Entre eles, o ERP interno recebeu destaque como um grupo específico, no qual foram detalhados os componentes correspondentes aos ERPs das principais fábricas e CD. Essa estrutura facilita o monitoramento de cada unidade operacional de maneira independente, possibilitando que incidentes específicos sejam rapidamente identificados e isolados sem afetar a visualização do estado geral do ERP. Além disso, serviços de infraestrutura, como servidores de impressão e *links* de comunicação (internos e externos), foram devidamente categorizados, permitindo um acompanhamento granular e segmentado por região e tipo de serviço.

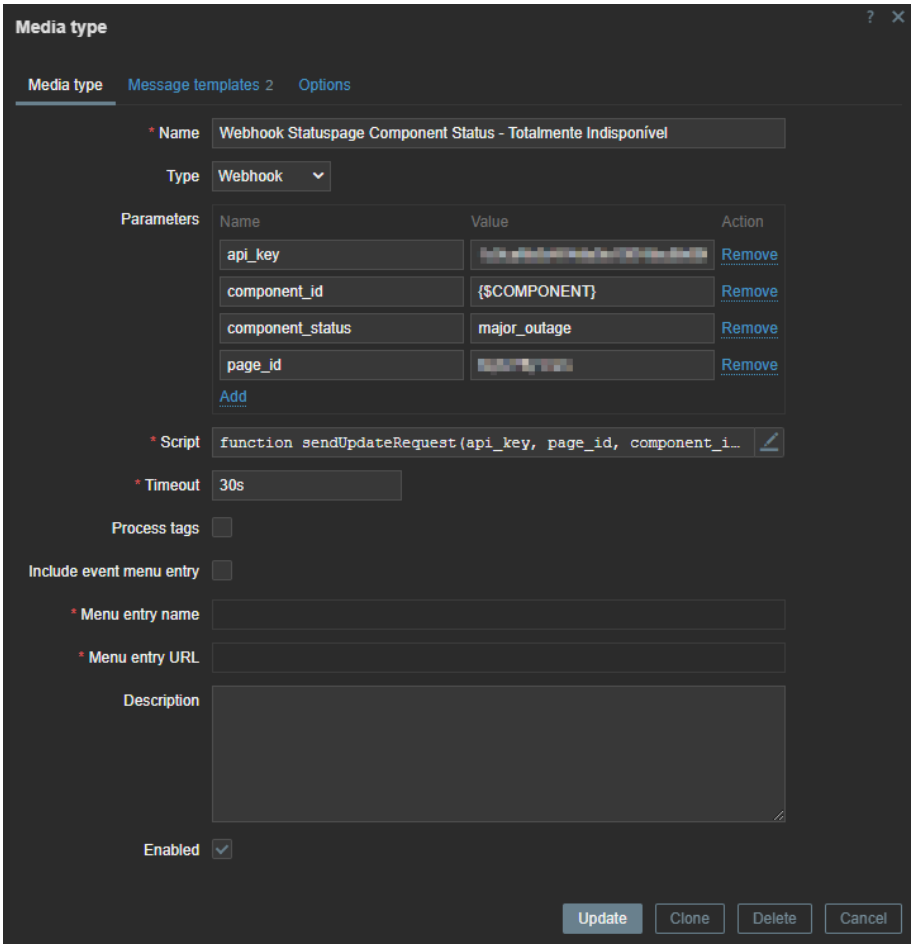
O estado das lojas, tanto nacionais quanto no exterior, foi tratado como um componente separado, dada a importância estratégica de garantir a continuidade operacional e a experiência de compra dos clientes. Essa categorização permite que a equipe responsável visualize rapidamente quais localidades estão funcionando normalmente e quais podem estar passando por algum tipo de instabilidade. Além disso, serviços *web* que suportam as operações de *e-commerce* e comunicação com clientes foram mapeados na plataforma, garantindo que qualquer problema com esses componentes seja rapidamente comunicado, minimizando impactos na experiência do usuário final.

A configuração foi pensada não apenas para fornecer uma visão clara da disponibilidade, mas também para monitorar manutenções planejadas e incidentes imprevistos de forma proativa. Assim, cada componente possui *status* que variam entre Operacional, Em Manutenção, Performance Degradada, Parcialmente Indisponível e Totalmente Indisponível, o que possibilita que a equipe de TI e os gestores tenham uma visão atualizada da saúde dos serviços. Esse nível de detalhamento permite, ainda, a emissão de relatórios e análises posteriores, ajudando na tomada de decisão para a alocação de recursos e no planejamento de melhorias para o ambiente de TI da Empresa X.

O monitoramento de ativos utilizando a plataforma *Zabbix* apresenta uma abordagem

robusta para a gestão de incidentes e serviços em ambientes distribuídos. No cenário descrito, as ações são executadas com base nos problemas identificados no sistema, como a perda de conexão com o *firewall* de uma loja, que ocasiona uma operação parcial do sistema local. Esse comportamento impede a comunicação com o sistema central, afetando o desempenho e a integridade das operações. Em casos críticos como este, é fundamental que as ações de resposta sejam automatizadas para garantir a continuidade dos serviços e a mitigação de impactos.

Figura 19 – Zabbix - Media Type Webhook

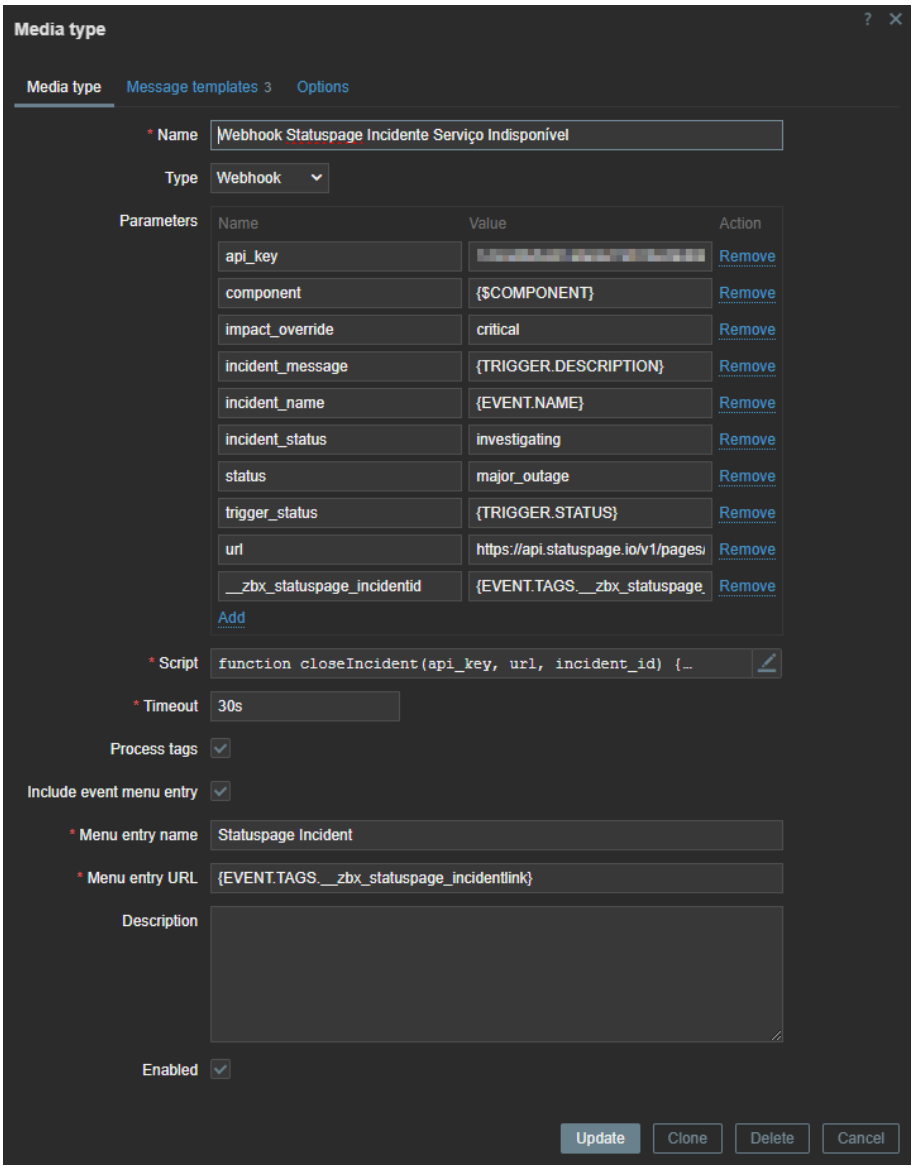


Fonte: Webhook configurado pela Empresa X

Para aumentar a eficiência das ações automatizadas, inicialmente foi proposta uma abordagem baseada no envio de *e-mails*. Essa solução consistia em configurar o Zabbix para disparar notificações para cada componente monitorado, utilizando os alertas para acionar ou atualizar o estado de operação do componente na plataforma de monitoramento. Embora funcional em teoria, essa abordagem apresentou limitações devido à arquitetura de organização de variáveis do Zabbix. A ferramenta não permite o uso de *macros* nos campos de destinatário, exigindo a configuração manual de um ou mais endereços de *e-mail* para cada componente monitorado. Essa limitação tornaria o processo de configuração e manutenção extremamente trabalhoso, aumentando a probabilidade de erros e a complexidade da solução.

Diante dessas limitações, optou-se por adotar *webhooks* [1], uma abordagem mais robusta e flexível para integração do Zabbix com outras plataformas de monitoramento e automação. Diferentemente da solução por *e-mail*, os *webhooks* permitem a comunicação direta entre sistemas, enviando alertas automaticamente quando eventos específicos ocorrem. Além disso, os *webhooks* suportam o uso de *macros* no nível de *hosts*, o que elimina a necessidade de configurar cada componente individualmente. Essa funcionalidade foi demonstrada na Figura 19, onde os *webhooks* foram configurados para atualizar páginas de monitoramento, além de abrir, atualizar e encerrar incidentes de forma integrada. Nesse contexto, foi definida uma *macro* personalizada, *{$COMPONENT}*, atribuída ao *host*, simplificando o processo de configuração e aumentando significativamente a eficiência operacional.

Figura 20 – Zabbix - Media Type Webhook



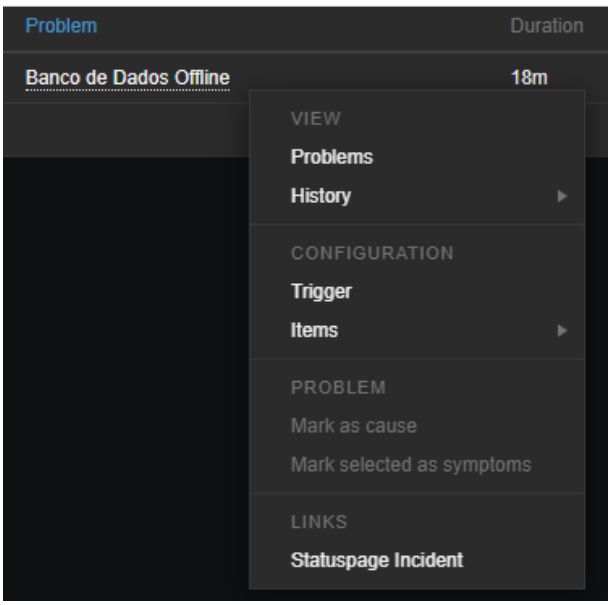
Fonte: Webhook configurado pela Empresa X

[1] Webhook: Método de comunicação em que um servidor envia dados automaticamente para outro, via HTTP, quando um evento ocorre

Uma das principais vantagens dos *webhooks* no Zabbix é a possibilidade de reutilizar configurações de maneira dinâmica por meio de variáveis associadas aos *hosts*. Essa flexibilidade permite que, ao alterar o *host*, o mesmo *webhook* seja aplicado a diferentes componentes, sem necessidade de ajustes adicionais. Contudo, houve situações específicas em que essa abordagem padrão não foi suficiente, como no monitoramento de *links* de internet e interfaces de rede pertencentes ao mesmo *host*, que no caso da Empresa X, é o *firewall*. Para resolver esse desafio, foi desenvolvido um *webhook* personalizado, capaz de mapear os componentes monitorados para suas respectivas interfaces de rede, utilizando informações como o nome do alerta ou as *tags* atribuídas aos problemas. Essa solução garantiu uma correspondência precisa dos alertas, permitindo ações rápidas e direcionadas, especialmente em situações críticas.

Para alertas de alta prioridade ou serviços críticos, foi projetado outro *webhook* com o objetivo de gerenciar incidentes de forma automatizada, demonstrado no Figura 20. Esse *webhook* cria automaticamente um incidente ao início do problema, com o link do incidente disponível no menu de contexto do problema no Zabbix, conforme ilustrado na Figura 21. Além disso, as atualizações e encerramentos dos problemas registrados no Zabbix são refletidos diretamente na plataforma *Statuspage*. Esse recurso assegura que, mesmo em situações em que os problemas sejam resolvidos automaticamente ou a equipe técnica não consiga atualizar o sistema por restrições de tempo, os clientes continuem recebendo informações atualizadas e precisas sobre o status dos serviços. Esse nível de integração melhora significativamente a transparência e a comunicação com os clientes, promovendo maior confiança e satisfação com os serviços oferecidos.

Figura 21 – Zabbix - Menu de Contexto



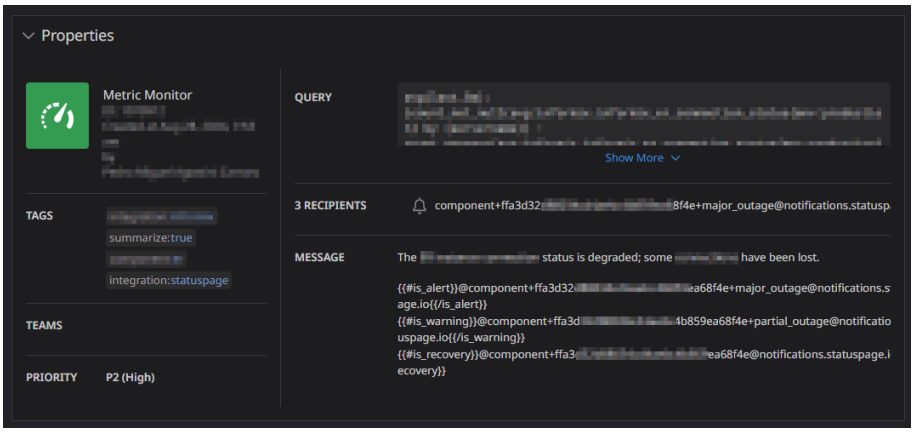
Fonte: Problema na plataforma Zabbix da Empresa X

No entanto, uma limitação identificada na plataforma é a ausência de integração nativa

com a gestão de manutenções programadas, o que exige um registro manual dessas atividades no sistema. Isso pode resultar em uma lacuna de controle e aumento de esforços operacionais para garantir a consistência dos dados. Para solucionar esse problema, sugere-se o desenvolvimento de um aplicativo externo que complemente a integração entre o Zabbix e outras plataformas de gerenciamento de manutenção. Dessa forma, seria possível automatizar o fluxo de informações e garantir que as mudanças planejadas sejam refletidas de maneira eficiente no ambiente de monitoramento, minimizando falhas e a necessidade de intervenções manuais.

A configuração de monitoramento no *Datadog* utiliza uma abordagem simplificada e eficiente baseada em automações via *e-mail* para gerenciar o *status* dos componentes monitorados, demonstrado na Figura 22. Cada elemento de infraestrutura possui um endereço de *e-mail* dedicado, onde a alteração de *status* é feita de maneira automatizada ao interpretar palavras-chave específicas no assunto da mensagem. Para indicar a disponibilidade de um componente, o termo *UP* é utilizado para marcar o estado operacional, enquanto *DOWN* é aplicado quando o serviço se encontra totalmente indisponível. Essa abordagem permite uma rápida atualização dos monitores com base nas mensagens enviadas, tornando a gestão de alertas mais ágil e menos suscetível a falhas manuais.

Figura 22 – Datadog - Monitors



Fonte: Monitor configurado pela Empresa X

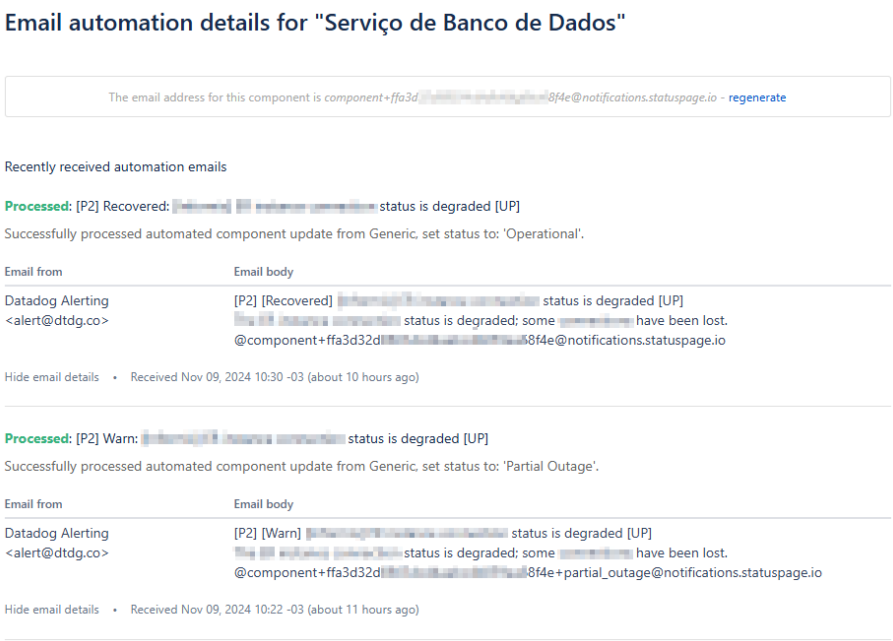
Para tornar as automações ainda mais refinadas, a *Statuspage* oferece uma configuração opcional que adiciona uma flexibilidade adicional ao processo. Ao incluir o símbolo de adição (“+”) seguido do estado de operação ao final do usuário no endereço de *e-mail*, é possível especificar diferentes níveis de problema, como "performance degradada"e "parcialmente indisponível", demonstrado no campo *message* da Figura 22, onde foi adicionado *major_outage* para o *e-mail* "componente+ffa[...]+major_outage@notifications.status.io". Isso facilita a categorização dos incidentes e ajuda a priorizar as ações corretivas com base na gravidade de cada evento. No entanto, apesar dessa capacidade de detalhamento, não é possível configurar o estado operacional utilizando esse método alternativo. A resolução de alertas continua dependente do envio de um *e-mail* padrão com o assunto *UP*, o que mantém a consistência na atualização

de *status*.

Para implementar automações utilizando *e-mails*, foi inicialmente planejado o uso de uma estrutura condicional para diferenciar os estados *UP* e *DOWN* nos alertas enviados pelo monitor. Nesse modelo, o mesmo assunto seria utilizado para notificar tanto a ocorrência quanto a resolução de um problema, com o estado sendo distinguido por uma variável no título do alerta. Contudo, essa abordagem apresentou uma limitação significativa: ao adicionar as informações de estado ao título do alerta, que também serve como assunto do *e-mail*, o campo se tornava poluído e menos intuitivo, dificultando a identificação rápida dos incidentes.

Durante a revisão da documentação da *Statuspage*, foi descoberto um método alternativo que resolveu esse problema. Ao adicionar o símbolo de adição ("+") seguido do estado de operação ao endereço de *e-mail* do destinatário, tornou-se possível especificar diretamente o nível de severidade do problema sem alterar o assunto da mensagem. Por exemplo, para categorizar um incidente como *major outage*, o endereço de *e-mail* poderia ser configurado como *componente+major_outage@notifications.status.io*. Essa técnica, ilustrada no campo *message* da Figura 22, permite que o estado do componente seja definido no corpo do *e-mail*, mantendo o assunto simplificado com apenas *UP* ou *DOWN*.

Figura 23 – Datadog - Monitors Alerts



Fonte: Monitor configurado pela Empresa X

Esse método oferece uma solução elegante ao evitar que o título do alerta fique sobrecarregado com informações desnecessárias, ao mesmo tempo em que possibilita um controle refinado sobre os estados intermediários de operação. Apesar disso, algumas limitações permanecem: o estado operacional *UP* ainda precisa ser utilizado tanto para resolução de problemas quanto para a restauração do status normal do monitor. Essa uniformidade no termo *UP* ga-

rante a consistência na atualização dos *status*, mas restringe a personalização dos estados de resolução.

Com base nessa abordagem, o fluxo de trabalho para alertas no Datadog foi adaptado para enviar mensagens utilizando o formato "+estado"no endereço de *e-mail*, permitindo diferenciar entre falhas parciais e totais durante um incidente. Ao término do evento, é enviado um *e-mail* com o assunto contendo apenas o termo *UP* para indicar que o componente voltou ao estado normal, como mostrado na Figura 23. Essa configuração simplificada melhora a comunicação entre os sistemas de monitoramento e a *Statuspage*, facilitando a gestão de incidentes com clareza e precisão, além de reduzir a complexidade na análise posterior dos eventos registrados.

Além disso, serviços de terceiros que possuem integração nativa com a plataforma foram adicionados diretamente na plataforma. Essa integração facilita o acompanhamento da disponibilidade e do desempenho dos ambientes monitorados, oferecendo visibilidade em tempo real para equipes e usuários finais. Através dessas integrações, tanto o Datadog quanto o Zabbix oferecem uma visão abrangente do estado dos serviços, possibilitando uma resposta rápida e eficaz em casos de indisponibilidade ou incidentes.

6.4 CASOS DE TESTES DAS INTEGRAÇÕES

Os testes executados envolveram diversas funcionalidades cruciais para garantir a eficácia da plataforma de monitoramento e sua capacidade de oferecer respostas rápidas e automáticas a incidentes e manutenções programadas. Focando nos principais serviços e ativos para o negócio, visando trazer melhor visibilidade da sustentação realizada nos componentes.

Um dos testes realizados foi o de abertura e encerramento automático de incidentes, configurado para atuar em casos de problemas críticos no BD do ERP, monitorado pelo Zabbix. Durante o teste, foi simulado um erro crítico que levou ao desligamento automático do BD do ERP, resultando em uma parada do sistema. Imediatamente, a plataforma de monitoramento alertou a equipe de TI, que acionou o time responsável para restaurar o ambiente à normalidade. Enquanto isso, os gestores de negócios puderam acompanhar as informações e atualizações sobre o incidente diretamente na plataforma, o que demonstrou a eficácia do sistema para manter as partes interessadas informadas em tempo real, minimizando a necessidade de comunicação direta com o time de TI.

Outro teste importante foi a programação de manutenção em um CD, com o objetivo de coordenar a manutenção entre as equipes de TI e as operações de negócios. A manutenção foi planejada e comunicada pela plataforma, que atualizou automaticamente o status do CD, permitindo que tanto a equipe de TI quanto a área de negócios acompanhassem o progresso e recebessem notificações até a liberação do ambiente. Esse teste demonstrou a utilidade da plataforma para sincronizar informações entre setores distintos, melhorando a gestão de manutenções e a transparência.

Por fim, foi realizada a validação do SLA de entrega dos links de internet e outros serviços de terceiros, utilizando o monitoramento de desempenho para avaliar o cumprimento dos acordos de nível de serviço. Durante o teste, um dos links de internet apresentou desempenho abaixo do esperado, e o sistema de monitoramento identificou a falha. Essa informação possibilitou que a equipe de TI entrasse em contato com o provedor responsável, cobrando melhorias e restabelecendo a qualidade esperada. Esse teste evidenciou como a plataforma auxilia na análise de desempenho de fornecedores externos, fornecendo dados concretos para ajustes de contrato e otimização do serviço.

Alguns testes, no entanto, não foram executados devido a limitações de tempo e recursos. A Integração com a Plataforma de Chamados não foi realizada pela falta de tempo disponível da equipe responsável. Além disso, a Integração com o Sistema de Fábricas SCADA e MES também não foi possível, pois esses sistemas ainda não contam com monitoramento contínuo do serviço, sendo monitorada apenas a infraestrutura física. Esses casos pendentes ressaltam pontos futuros para expandir o monitoramento e a integração da plataforma.

7 CONSIDERAÇÕES FINAIS

O presente estudo teve como objetivo principal testar e selecionar uma plataforma de monitoramento de serviços e ativos de TI que apoiasse os gestores no processo decisório. Os resultados obtidos evidenciaram o êxito dessa empreitada, satisfazendo as necessidades identificadas e oferecendo maior clareza sobre o estado e o desempenho da infraestrutura de TI da organização.

Inicialmente, os serviços e ativos de TI foram caracterizados e analisados em sua totalidade. Esse processo, abordado nos capítulos iniciais, permitiu entender a relevância de cada componente e sua relação com os processos de negócios, evidenciando a importância de uma solução robusta de monitoramento. Em seguida, a infraestrutura de sistemas foi descrita em detalhes, possibilitando uma visão clara dos requisitos técnicos e operacionais necessários para suportar a solução escolhida.

Uma avaliação detalhada das ferramentas disponíveis no mercado foi realizada, considerando suas funcionalidades práticas, integração e custo-benefício. A análise levou à escolha da *Statuspage* como a solução mais apropriada, devido à sua personalização, integração com sistemas existentes e eficiência na comunicação de incidentes. O processo de seleção baseou-se em testes e avaliações técnicas, assegurando que a solução implementada estivesse em consonância com os objetivos estratégicos da empresa.

A implementação da *Statuspage* incluiu etapas como o contato com fornecedores, configuração inicial, integração com ferramentas de monitoramento, como *Zabbix* e *Datadog*, além de testes extensivos para validar sua eficácia. O treinamento das equipes de TI e a apresentação da solução aos gestores e clientes foram passos fundamentais para assegurar a adoção e o pleno aproveitamento da ferramenta.

Por fim, o projeto alcançou seus objetivos específicos ao oferecer uma plataforma que centraliza e simplifica o monitoramento, promovendo maior transparência e eficiência na comunicação de incidentes e no gerenciamento dos ativos de TI. A solução implementada não apenas resolve os problemas de comunicação identificados no início do trabalho, mas também estabelece uma base sólida para expansões futuras, como o mapeamento de novos serviços a serem monitorados.

O cronograma detalhado das atividades realizadas durante o projeto é apresentado na Tabela 5. Este cronograma destaca o planejamento e a execução das etapas essenciais para garantir o sucesso do trabalho.

Tabela 5 – Cronograma do Projeto

Data de Início	Data de Término	Descrição
02/08/24	10/09/24	Realização de contatos com a *Instatus* por *e-mail*.
05/09/24	05/09/24	Decisão final, com a gestão da Empresa X, para cancelar a PoC da *Instatus*.
02/08/24	16/08/24	Contato da equipe responsável com a implementadora da *Atlassian* para tratar da solução *Statuspage*.
14/08/24	30/08/24	Liberação e configuração inicial da página da *Statuspage*.
02/09/24	05/09/24	Configuração da integração dos componentes com o *Datadog Monitor*.
02/09/24	05/09/24	Configuração da integração dos componentes com o *Zabbix* utilizando *e-mail*.
05/09/24	18/09/24	Configuração da integração dos componentes com o *Zabbix* utilizando *webhooks*.
18/09/24	27/09/24	Configuração da integração para gerenciamento de incidentes no *Zabbix* por *webhooks*.
27/09/24	02/10/24	Realização de testes de incidentes no *Zabbix*.
30/09/24	06/10/24	Validação do histórico e dos SLAs dos componentes integrados ao *Datadog* e ao *Zabbix*.
07/10/24	07/10/24	Apresentação da solução à gestão da Empresa X.
08/10/24	08/11/24	Mapeamento de novos serviços a serem monitorados na solução.
08/10/24	16/10/24	Treinamento interno do departamento de TI para operação da plataforma.
21/10/24	31/10/24	Apresentação da solução final para os clientes da Empresa X.

Fonte: Elaborado pelo autor (2024).

Assim, este estudo proporciona uma contribuição valiosa para aumentar a visibilidade da TI perante o cliente e para apoiar a gestão de TI em decisões mais proativas, eficientes e baseadas em dados. A experiência acumulada durante este projeto também serve como um exemplo para organizações que lidam com desafios semelhantes no controle de seus serviços e recursos de TI, sublinhando a importância de soluções convergentes e transparentes na administração tecnológica.

REFERÊNCIAS

AFSHARI, A. R. **Simple Additive Weighting Approach to Personnel Selection Problem**. 2024. Scientific Figure on ResearchGate. Disponível em: <https://www.researchgate.net/figure/SAATYS-1-9-SCALE-OF-PAIRWISE-COMPARISONS_tbl1_285828294>.

AL-JABRI, I. M.; ROZTOCKI, N. Adoption of erp systems: Does information transparency matter? **Telematics and Informatics**, v. 32, n. 2, p. 300–310, 2015. ISSN 0736-5853. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0736585314000653>.

ALI, E. *et al.* Optimizing server resource by using virtualization technology. **Procedia computer science**, Elsevier, v. 59, p. 320–325, 2015.

ANDERSON, C. Docker [software engineering]. **Ieee Software**, IEEE, v. 32, n. 3, p. 102–c3, 2015.

ANWAR, R. W.; ABDULLAH, T.; PASTORE, F. Firewall best practices for securing smart healthcare environment: A review. **Applied Sciences**, MDPI, v. 11, n. 19, p. 9183, 2021.

Atlassian. **ITIL: An Essential Guide to IT Service Management**. 2024. <https://www.atlassian.com/itsm/itil>.

ATLASSIAN. **Statuspage Demo**. 2024. <https://acmecorp2.statuspage.io/>.

BIDGOLI, H. **The handbook of technology management, supply chain management, marketing and advertising, and global management**. [S.l.]: John Wiley & Sons, 2010. v. 2.

BRINGHENTI, D. *et al.* Automated firewall configuration in virtual networks. **IEEE Transactions on Dependable and Secure Computing**, IEEE, v. 20, n. 2, p. 1559–1576, 2022.

BURNS, B. *et al.* **Kubernetes: up and running**. [S.l.]: "O'Reilly Media, Inc.", 2022.

CASE, J. *et al.* **A Simple Network Management Protocol (SNMP)**. 1990. <https://datatracker.ietf.org/doc/html/rfc1157>.

CASE, J. D. *et al.* **Simple network management protocol (SNMP)**. [S.l.], 1989.

CHANDRAMOULI, R. **Implementation of DevSecOps for a Microservices-based Application with Service Mesh**. Gaithersburg, MD, 2022. Available free at <https://doi.org/10.6028/NIST.SP.800-204C>.

CHOWDHARY, A. **Upptime Demo**. 2024. <https://demo.upptime.js.org>.

Cisco. **Understanding the Different Types of Network Switches**. <https://www.cisco.com/c/en/us/solutions/small-business/resource-center/networking/understanding-the-different-types-of-network-switches.html>. Accessed: 2024-05-05.

____. **What is a Data Center?** <https://www.cisco.com/c/en/us/solutions/data-center-virtualization/what-is-a-data-center.html>. Accessed: 2024-05-05.

Cloudflare. **What is a Network Switch?** <https://www.cloudflare.com/pt-br/learning/network-layer/what-is-a-network-switch/>. Accessed: 2024-05-05.

DATABANK. **Powering The Future: The Move To High Performance Computing Data Centers**. 2023. Acesso em: 18 maio 2024. Disponível em: <https://www.databank.com>.

DAVENPORT, T. H. Putting the enterprise into the enterprise system. **Harvard Business Review**, Harvard Business Publishing, v. 76, n. 4, p. 121–131, 1998.

DELOITTE. **Digital Utility Asset Management: Power and Utilities**. 2024. <https://www2.deloitte.com/us/en/insights/industry/power-and-utilities/digital-utility-asset-management.html>.

Docker. **Add Manager Nodes for Fault Tolerance**. <https://docs.docker.com/engine/swarm/admin_guide/#add-manager-nodes-for-fault-tolerance>. Accessed: 2024-05-05.

DOCKER. **Docker Documentation**. 2024. Accessed: April 16, 2024. Disponível em: <https://docs.docker.com/>.

DRAGONI, N. *et al.* Microservices: yesterday, today, and tomorrow. **Present and ulterior software engineering**, Springer, p. 195–216, 2017.

ELMASRI, R. *et al.* Fundamentals of database systems. In: SPRINGER NATURE. **Advances in Databases and Information Systems: 24th European Conference, ADBIS 2020, Lyon, France, August 25–27, 2020, Proceedings**. [S.l.], 2020. v. 12245, p. 139.

FANTIN, K.; NOTARI, D. L.; SILVA, S. A. e. Análise de utilização de ambiente virtual de aprendizagem por estudantes e professores na educação básica. **Revista Educação e Cultura Contemporânea**, 2021. Disponível em: <http://dx.doi.org/10.5935/2238-1279.20210005>.

FENG, G.-l.; YANG, L.-h. A new method in improving database connection pool model. **World Academy of Science, Engineering and Technology**, Citeseer, v. 29, p. 246–249, 2007.

FINE, L. G. **SWOT Analysis: A Tool for Making Better Business Decisions**. [S.l.]: Kick It, LLC, 2009. ISBN 978-1-4392-3884-8.

FOWLER, M.; LEWIS, J. Microservices: A definition of this new architectural term. **ThoughtWorks**, ThoughtWorks, 2014.

GALLOWAY, B.; HANCKE, G. P. Introduction to industrial control networks. **IEEE Communications surveys & tutorials**, IEEE, v. 15, n. 2, p. 860–880, 2012.

HAN, J. H. *et al.* Blueswitch: Enabling provably consistent configuration of network switches. In: IEEE. **2015 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)**. [S.l.], 2015. p. 17–27.

HARIZOPOULOS, S. *et al.* Oltp through the looking glass, and what we found there. In: **Making Databases Work: the Pragmatic Wisdom of Michael Stonebraker**. [S.l.: s.n.], 2018. p. 409–439.

HAYEK, W. Y. A.; ODEH, R. A. A. Cloud erp vs on-premise erp. **International Journal of Applied Science and Technology**, v. 10, n. 4, 2020.

HEDMAN, R.; SUBRAMANIYAN, M.; ALMSTRÖM, P. Analysis of critical factors for automatic measurement of oee. **Procedia Cirp**, Elsevier, v. 57, p. 128–133, 2016.

IBM. **Enterprise Replication**. <https://www.ibm.com/docs/en/informix-servers/14.10?topic=replication-enterprise>. Accessed: 2024-05-05.

INSTATUS, I. **Instatus Demo**. 2024. <https://acmecorp.instatus.com/>.

IOANNIDES, M. G. Design and implementation of plc-based monitoring control system for induction motor. **IEEE transactions on energy conversion**, IEEE, v. 19, n. 3, p. 469–476, 2004.

JOHNSON, M. W. *et al.* Evolving standards for it service management. **IBM Systems Journal**, IBM, v. 46, n. 3, p. 583–597, 2007.

KALSIN, V.; NOLAN, T. **How to Install and Configure Zabbix to Securely Monitor Remote Servers on CentOS 7**. [S.l.]: DigitalOcean, 2019. <https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-zabbix-to-securely-monitor-remote-servers-on-centos-7>. Accessed: 2024-05-05.

KATUU, S. Enterprise resource planning: past, present, and future. **New Review of Information Networking**, Taylor & Francis, v. 25, n. 1, p. 37–46, 2020.

KISTIJANTORO, A. I. *et al.* Enhancing an application server to support available components. **IEEE Transactions on Software Engineering**, IEEE, v. 34, n. 4, p. 531–545, 2008.

Kubernetes. **Kubernetes Documentation**. 2024. Accessed: 2024-04-29. Disponível em: <https://kubernetes.io/docs/>.

KUBERNETES, T. Kubernetes. **Kubernetes. Retrieved May**, v. 24, p. 2019, 2019.

LEE, S.; KIM, B. G. Factors affecting the usage of intranet: A confirmatory study. **Computers in Human Behavior**, Elsevier, v. 25, n. 1, p. 191–201, 2009.

LEPPÄNEN, T. Data visualization and monitoring with grafana and prometheus. 2021.

LIMITED., A. T. S. **Cachet 3.x Demo**. 2024. <https://v3.cachethq.io/>.

LIMITED, A. T. S. **Cachet Demo**. 2024. <https://demo.cachethq.io/>.

LÜBKE, D. *et al.* Interface evolution patterns: Balancing compatibility and extensibility across service life cycles. In: **Proceedings of the 24th European Conference on Pattern Languages of Programs**. New York, NY, USA: Association for Computing Machinery, 2019. (EuroPLoP '19).

ManageEngine. **Layer 2 vs. Layer 3 Switch: What's the difference?** <https://www.manageengine.com/products/oputils/tech-topics/layer2-vs-layer3-switch.html>. Accessed: 2024-05-05.

MASICH, I. S. *et al.* Prediction of critical filling of a storage area network by machine learning methods. **Electronics**, MDPI, v. 11, n. 24, p. 4150, 2022.

MAURO, D.; SCHMIDT, K. **Essential SNMP: Help for System and Network Administrators**. [S.l.]: "O'Reilly Media, Inc.", 2005.

MCAFEE, A. P. Enterprise 2.0: The dawn of emergent collaboration. **MIT Sloan Management Review**, MIT Sloan Management Review, v. 47, n. 3, p. 21–28, 2006.

MCCLUSKY, K.; PELTON, S. Integrating sql databases with scada and plc systems for efficient production data management. **Industrial Automation Journal**, v. 45, n. 2, p. 215–230, 2023.

MIRKOVIC, J.; REIHER, P. A taxonomy of ddos attack and ddos defense mechanisms. **ACM SIGCOMM Computer Communication Review**, ACM New York, NY, USA, v. 34, n. 2, p. 39–53, 2004.

NEWMAN, S. **Building microservices**. [S.l.]: "O'Reilly Media, Inc.", 2021.

NGUYEN, T.-T. *et al.* Horizontal pod autoscaling in kubernetes for elastic container orchestration. **Sensors**, MDPI, v. 20, n. 16, p. 4621, 2020.

NUGROHO, R. A.; ROSYANI, P. Implementation of environmental device monitoring using zabbix in the pusat data sarana dan informasi (pdsi). **OKTAL: Jurnal Ilmu Komputer dan Sains**, v. 2, n. 07, p. 1846–1873, 2023.

PEIXIAN, C. *et al.* Research on cluster monitoring and prediction platform based on zabbix technology. In: IOP PUBLISHING. **IOP Conference Series: Earth and Environmental Science**. [S.l.], 2020. v. 512, n. 1, p. 012155.

PHAN, L.-A.; KIM, T. *et al.* Traffic-aware horizontal pod autoscaler in kubernetes-based edge computing infrastructure. **IEEE Access**, IEEE, v. 10, p. 18966–18977, 2022.

PREECE, J.; SHARP, H.; ROGERS, Y. **Interaction Design: Beyond Human-Computer Interaction**. [S.l.]: Wiley, 2013.

PRIYADARSHI, P. T. R. N. Understanding computer networks: A comprehensive overview of types, configurations, and the osi model. **arXiv preprint arXiv:2403.11296**, 2024.

RABETSKI, P.; SCHNEIDER, G. Migration of an on-premise application to the cloud: Experience report. In: SPRINGER. **European Conference on Service-Oriented and Cloud Computing**. [S.l.], 2013. p. 227–241.

RAWOOF, F. M.; TAJAMMUL, M.; JAMAL, F. On-premise server monitoring with prometheus and telegram bot. **International journal of scientific research in engineering and management**, v. 6, n. 04, 2022.

RESENDE, J. S. *et al.* Towards a modular on-premise approach for data sharing. **Sensors**, MDPI, v. 21, n. 17, p. 5805, 2021.

SERRANO, J. *et al.* An it service management literature review: Challenges, benefits, opportunities and implementation practices. **Information**, MDPI, v. 12, n. 3, p. 111, 2021.

SERVERWATCH. **What is a Server? | Definition, Types, and Features**. 2023. Acesso em: 5 maio 2024. Disponível em: <https://www.serverwatch.com>.

SHARMA, R. N. **Kener Demo**. 2024. <https://kener.ing/>.

SILVA, R. A.; SILVA, W. J. A implementação do zabbix com segurança: Um estudo de caso zabbix safely. **Revista Foco**, v. 17, n. 4, p. e4851–e4851, 2024.

SINGJAI, A. *et al.* Patterns on designing api endpoint operations. 2021.

SOMERS, T. M.; NELSON, K. G. The impact of critical success factors across the stages of enterprise resource planning implementations. **Journal of Management Information Systems**, Taylor Francis, v. 17, n. 4, p. 103–116, 2001.

STALLINGS, W. **Foundations of modern networking: SDN, NFV, QoE, IoT, and Cloud**. [S.l.]: Addison-Wesley Professional, 2015.

STEINDER, M.; WHALLEY, I.; CHESS, D. Server virtualization in autonomic management of heterogeneous workloads. **ACM SIGOPS Operating Systems Review**, ACM New York, NY, USA, v. 42, n. 1, p. 94–95, 2008.

SULAIMAN, F.; ZAILANI, S.; RAMAYAH, T. Intranet portal utilization: monitoring tool for productivity-quality and acceptance point of view. **Procedia-social and behavioral sciences**, Elsevier, v. 65, p. 381–386, 2012.

TECHOPEDIA. **What is a Server? Definition, Form Factors, Components and Uses**. 2023. Acesso em: 27 abril 2024. Disponível em: <https://www.techopedia.com>.

THAKAR, A.; BHATIA, R. Converged rack architecture (cra) for data center. 2022.

TSO, F. P.; JOUET, S.; PEZAROS, D. P. Network and server resource management strategies for data centre infrastructures: A survey. **Computer Networks**, Elsevier, v. 106, p. 209–225, 2016.

TURRI, F.; SANTINI, M. Extranet and web portal for the management of the supply chain processes: an empirical analysis. Politecnico di Milano, 2015.

VMware Inc. **VMware vSphere Documentation**. [S.l.], 2023. Acessado em 15 de abril de 2024. Disponível em: <https://docs.vmware.com/en/VMware-vSphere/index.html>.

WEI, C.-C.; WANG, M.-J. A comprehensive framework for selecting an erp system. **International Journal of Project Management**, v. 22, p. 161–169, 02 2004.

WETHERALL, D. J.; TANENBAUM, A. S. **Computer networks**. [S.l.]: Pearson Education, 2013.

WUNDER, J.; HALBARDIER, A.; WALTERMIRE, D. **Title of the Report**. Gaithersburg, MD, 2011.

YADAV, G.; PAUL, K. Architecture and security of scada systems: A review. **International Journal of Critical Infrastructure Protection**, Elsevier, v. 34, p. 100433, 2021.

YLONEN, T.; LONVICK, E. C. **The Secure Shell (SSH) Connection Protocol**. [S.l.], 2006. SSH Communications Security Corp and Cisco Systems, Inc. Disponível em: <https://www.rfc-editor.org/rfc/rfc4254>.

Zabbix. **Zabbix Documentation**. <https://www.zabbix.com/documentation/current/en/manual/>. Accessed: 2024-05-05.

APÊNDICE A – PROCESSO HIERÁRQUICO ANALÍTICO

O modelo AHP consiste em definir um objetivo, seus critérios e alternativas propostas. Os critérios são comparados em pares para determinar sua importância, Tabela 6, utilizando uma escala de 1 a 9 de importância, demonstrado na Figura 24.

Figura 24 – Escala de Importância na Comparação por Pares

Intensity of importance	Definition	Explanation
1	Equal Importance	Two activities contribute equally to the objective
2	Weak or Slight	
3	Moderate Importance	Experience and judgment slightly favor one activity over another
4	Moderate Plus	
5	Strong Importance	Experience and judgment strongly favor one activity over another
6	Strong Plus	
7	Very Strong	An activity is favored very strongly over another
8	Very, very Strong	
9	Extreme Importance	The evidence favoring one activity over another is of the highest

Fonte: *SAATY'S 1-9 SCALE OF PAIRWISE COMPARISONS* (AFSHARI, 2024)

Tabela 6 – Comparação por Pares dos Critérios

	Manut.	Grupos	API	Zabbix	Datadog	Chamado	Email	Chat	Histórico	SLA
Manut.	1	5/1	2/1	1/3	1/3	1/3	1/2	2/1	3/1	2/1
Grupos	1/5	1	1/3	1/5	1/5	1/5	1/3	1/2	1/3	1/2
API	1/2	3/1	1	1/2	1/2	1/2	1/2	3/1	1/2	3/1
Zabbix	3/1	5/1	2/1	1	1	1	3/1	5/1	2/1	2/1
Datadog	3/1	5/1	2/1	1	1	1	3/1	5/1	2/1	2/1
Chamado	3/1	5/1	2/1	1	1	1	3/1	5/1	2/1	2/1
Email	2/1	3/1	2/1	1/3	1/3	1/3	1	5/1	5/1	7/1
Chat	1/2	2/1	1/3	1/5	1/5	1/5	1/5	1	1/4	1/3
Histórico	1/3	3/1	2/1	1/2	1/2	1/2	1/5	4/1	1	2/1
SLA	2/1	2/1	1/3	1/2	1/2	1/2	1/7	3/1	1/2	1

Fonte: Elaborado pelo autor (2024)

Os valores da Tabela 6 são trocados de suas formas fatoradas para decimais, Tabela 7, e os valores das colunas são somados, Tabela 8, para normalizar os valores, Tabela 9. Assim, é calculado a média simples das linhas na tabela de valores normalizado, Tabela 10, que indica a prioridade do critério.

Tabela 7 – Forma Decimal dos Pesos da Comparação por Pares

	Manut.	Grupos	API	Zabbix	Datadog	Chamados	Email	Chat	Histórico	SLA
Manut.	1,00	5,00	2,00	0,33	0,33	0,33	0,50	2,00	3,00	2,00
Grupos	0,20	1,00	0,33	0,20	0,20	0,20	0,33	0,50	0,33	0,50
API	0,50	3,00	1,00	0,50	0,50	0,50	0,50	3,00	0,50	3,00
Zabbix	3,00	5,00	2,00	1,00	1,00	1,00	3,00	5,00	2,00	2,00
Datadog	3,00	5,00	2,00	1,00	1,00	1,00	3,00	5,00	2,00	2,00
Chamados	3,00	5,00	2,00	1,00	1,00	1,00	3,00	5,00	2,00	2,00
Email	2,00	3,00	2,00	0,33	0,33	0,33	1,00	5,00	5,00	7,00
Chat	0,50	2,00	0,33	0,20	0,20	0,20	0,20	1,00	0,25	0,33
Histórico	0,33	3,00	2,00	0,50	0,50	0,50	0,20	4,00	1,00	2,00
SLA	2,00	2,00	0,33	0,50	0,50	0,50	0,14	3,00	0,50	1,00

Fonte: Elaborado pelo autor (2024)

Tabela 8 – Soma dos Pesos da Comparação por Pares

Critério	Soma
Manutenção Programada	15,53
Grupo de Serviços	34,00
API	14,00
Zabbix	5,57
Datadog	5,57
Chamados	5,57
Email	11,88
Google Chat	33,50
Histórico	16,58
SLA	21,83

Fonte: Elaborado pelo autor (2024)

Tabela 9 – Pesos Normalizado da Comparação por Pares

Manut.	0,0644	0,1471	0,1429	0,0599	0,0599	0,0599	0,0421	0,0597	0,1809	0,0916	0,0908
Grupos	0,0129	0,0294	0,0238	0,0359	0,0359	0,0359	0,0281	0,0149	0,0201	0,0229	0,0260
API	0,0322	0,0882	0,0714	0,0898	0,0898	0,0898	0,0421	0,0896	0,0302	0,1374	0,0761
Zabbix	0,1931	0,1471	0,1429	0,1796	0,1796	0,1796	0,2526	0,1493	0,1206	0,0916	0,1636
Datadog	0,1931	0,1471	0,1429	0,1796	0,1796	0,1796	0,2526	0,1493	0,1206	0,0916	0,1636
Chamados	0,1931	0,1471	0,1429	0,1796	0,1796	0,1796	0,2526	0,1493	0,1206	0,0916	0,1636
Email	0,1288	0,0882	0,1429	0,0599	0,0599	0,0599	0,0842	0,1493	0,3015	0,3206	0,1395
Chat	0,0322	0,0588	0,0238	0,0359	0,0359	0,0359	0,0168	0,0299	0,0151	0,0153	0,0300
Histórico	0,0215	0,0882	0,1429	0,0898	0,0898	0,0898	0,0168	0,1194	0,0603	0,0916	0,0810
SLA	0,1288	0,0588	0,0238	0,0898	0,0898	0,0898	0,0120	0,0896	0,0302	0,0458	0,0658

Fonte: Elaborado pelo autor (2024)

Com a importância dos critérios, é calculado pontuação dos alternativos. Para isso, é listado quais critérios são atendidos pelas soluções, Tabela 11, e a o valor é multiplicado pela importância que foi calculada para o critério, gerando a pontuação demonstrada na Tabela 12.

Tabela 10 – Prioridade dos Critérios

Critério	Prioridade
Zabbix	0,1636
Datadog	0,1636
Sistema de Chamados	0,1636
Email	0,1395
Manutenção Programada	0,0908
Histórico	0,0810
API	0,0761
SLA	0,0658
Chat	0,0300
Grupo de Serviços	0,0260

Fonte: Elaborado pelo autor (2024)

Tabela 11 – Critérios atendidos pelas Alternativas.

	upptime	Kener	Cachet	Statuspage	instatus
Manutenção Programada	1	0	1	1	1
Grupo de Serviços	0	1	1	1	1
API	1	1	1	1	1
Zabbix	0	0	1	0	0
Datadog	0	0	0	1	1
Sistema de Chamados	0	0	0	1	0
Email	1	0	1	1	1
Chat	0	0	0	0	1
Histórico	1	1	0	1	1
SLA	1	1	0	1	1

Fonte: Elaborado pelo autor (2024)

Tabela 12 – Resultado do Modelo AHP.

Alternativo	Pontuação
Statuspage	0,8064
instatus	0,6728
Cachet	0,4960
upptime	0,4532
Kener	0,2489

Fonte: Elaborado pelo autor (2024)