

UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

EDICLEIDE ALVES DOS SANTOS, 2003531
JULIANA NELIA DO NASCIMENTO CORREA, 2013603
LUIZ EDUARDO XAVIER, 2005625
MATHEUS VINÍCIUS DIAS BARBOSA, 1706186
MICHELLE GOMES GUIMARÃES, 2004795
MURILO DA SILVA AMARO, 2006172
RENATO NOGUEIRA DA SILVA, 2009044
WELLINGTON WASHINGTON ANDRADE DE MELO JUNIOR,
201476

**Zabbix Graphics Reports: relatórios gráficos de falhas de redes
monitoradas**

Vídeo de apresentação do Projeto Integrador

<https://www.youtube.com/watch?v=8TsVFJbvDs0>

Osasco - SP
2022

UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

Zabbix Graphics Reports: relatórios gráficos de falhas de redes monitoradas

Relatório Técnico-Científico apresentado na disciplina de Projeto Integrador para o curso de Bacharelado em Ciência de Dados da Universidade Virtual do Estado de São Paulo (UNIVESP).

SANTOS, Edicleide Alves dos; CORREA, Juliana Nelia do Nascimento; XAVIER, Luiz Eduardo; BARBOSA, Matheus Vinicius Dias; GUIMARÃES, Michelle Gomes; AMARO, Murilo da Silva; SILVA, Renato Nogueira da; JÚNIOR, Wellington Washington Andrade de Melo. **Zabbix Graphics Reports: relatórios gráficos de falhas de redes monitoradas**. 34f. Relatório Técnico-Científico. Bacharelado em Ciência de Dados – **Universidade Virtual do Estado de São Paulo**. Tutor: Salvador Falcón Canillas. Polo Osasco (UAB), 2022.

RESUMO

O Zabbix é uma ferramenta de monitoramento de infraestrutura de redes, possui código fonte aberto, sua arquitetura foi feita para servidores Linux e é capaz de monitorar qualquer equipamento que possua um endereço MAC (Media Access Control ou Controle de Acesso de Mídia). Possui uma API (Application Programming Interface ou Interface de Programação de Aplicação) bem completa que permite que as funcionalidades do Zabbix sejam acessadas, utilizadas e melhoradas através de programas auxiliares que podem ser desenvolvidos por programadores que tenham necessidades mais específicas. Em uma pesquisa feita na comunidade Zabbix Brasil (grupo do Telegram), foram levantadas algumas demandas de aplicações que poderiam ser desenvolvidas com a utilização da API do Zabbix para implementação de melhorias ou atendimento de necessidades dos administradores. Uma demanda levantada foi a falta de uma ferramenta, de fácil manuseio, para a criação gráficos de falhas a partir do banco de dados. Diante do que foi levantado, houve a idealização do Zabbix Graphics Reports. O Zabbix Graphics Reports (ZGR) é uma ferramenta web desenvolvida pelos alunos da UNIVESP, para a disciplina de Projeto Integrador II, em Javascript que utiliza a API do Zabbix para criar relatórios gráficos das falhas das redes monitoradas através dos registros do banco de dados, melhorando a usabilidade da plataforma Zabbix. Focado em User Experience o desenvolvimento do ZGR utilizou o GitHub para controle de versão, técnicas de acessibilidade e testes manuais e automatizados. O protótipo está hospedado na nuvem da AWS.

PALAVRAS-CHAVE: Zabbix; Gerenciamento de Redes; Aceitação de Tecnologias; API

LISTA DE ILUSTRAÇÕES

Figura 1 - Arquitetura do ZABBIX.....	18
Figura 2 - Modelo de Aceitação de Tecnologia.....	19
Figura 3 - Segundo Modelo de Aceitação de Tecnologia (TAM2).....	19
Figura 4 - Modelo Unificado de Aceitação e Uso de Tecnologia (UTAUT).....	20
Figura 5 – Plano de ação no Trello.....	24
Figura 6 – Login do ZABBIX.....	25
Figura 7 – Host em monitoramento pelo ZABBIX.....	26
Figura 8 – Laboratório no VirtualBOX com o servidor ZABBIX.....	26
Figura 9 – Relatório gerado pelo ZABBIX.....	27
Figura 10 – Tela de Login para validação de Cadastro no ZABBIX.....	27
Figura 11 – Lista contendo falhas de cada host no ZABBIX.....	28
Figura 12 – Tela para seleção de período a ser listado.....	28
Figura 13 – Tela para seleção de período para geração do gráfico.....	29
Figura 14 – Gráfico gerado.....	29
Figura 15 – Gráfico gerado.....	30
Figura 16 – Código fonte gerado utilizando o Visual Studio Code – Parte 1.....	30
Figura 17 – Código fonte gerado utilizando o Visual Studio Code – Parte 2.....	31
Figura 18 – Código fonte gerado utilizando o Visual Studio Code – Parte 3.....	31

SUMÁRIO

1 INTRODUÇÃO.....	11
2 DESENVOLVIMENTO.....	13
2.1 Objetivos.....	13
2.2 Justificativa e delimitação do problema.....	13
2.3 Fundamentação teórica.....	15
2.3.1 <i>Sistemas de Informações Gerenciais.....</i>	<i>16</i>
2.3.2 <i>Desenvolvimento de Software.....</i>	<i>16</i>
2.3.3 <i>APIs.....</i>	<i>16</i>
2.3.4 <i>Zabbix e o gerenciamento de Redes.....</i>	<i>17</i>
2.3.5 <i>Modelo de aceitação de tecnologias.....</i>	<i>18</i>
2.4 Metodologia.....	21
2.4.1 <i>Seleção e descrição do contexto.....</i>	<i>21</i>
2.4.2 <i>Interpretação do contexto.....</i>	<i>22</i>
2.4.3 <i>Criação e prototipação: criação do Laboratório de Testes.....</i>	<i>23</i>
2.4.4 <i>Execução do projeto: ferramentas de Gestão de Projetos.....</i>	<i>23</i>
3 RESULTADOS.....	25
3.1 Solução inicial.....	25
3.2 Solução Final.....	28
4 CONSIDERAÇÕES FINAIS.....	32
REFERÊNCIAS.....	33

1 INTRODUÇÃO

Gerenciamento de redes de computadores diz respeito à “[...] coordenação, controle de atividade, e monitoramento de recursos, assegurando [...] confiabilidade, segurança e alta disponibilidade” (BENÍCIO, 2015, p.20) e ocorre em três etapas: coleta de dados, diagnóstico e ação e controle. É uma atividade crítica em uma era digital como a que vivemos.

Um dos problemas enfrentados pela comunidade é a existência de softwares de custo acessível e escaláveis que permitam às empresas e profissionais monitorarem e solucionar problemas de suas redes de computadores.

O ZABBIX é um software livre para monitoramento de redes que tem sua API (*Application Programming Interface*) aberta, o que permite o uso de seu código para o desenvolvimento de interfaces de fácil comunicação com outras linguagens. É uma ferramenta interessante, pois é adequada ao suporte a uma atividade crítica na era digital – o monitoramento de redes – e tem acesso aberto ao seu código: é um software livre e aberto, o que permite o acesso, modificação e uso por qualquer pessoa, sem custo, e o benefício de um suporte amplo da comunidade de usuários.

Entretanto, sua interface não oferece a opção de criar gráficos do histórico de falhas, o que dificulta para as empresas de monitoramento emitirem relatórios detalhados sobre as intervenções que foram efetuadas nas redes e equipamentos de seus clientes.

Nesse sentido, tendo como base o tema do projeto integrador, a ideia básica deste projeto é desenvolver uma aplicação web que consumirá os dados da API do ZABBIX, e será capaz de gerar relatórios rápidos e de fácil entendimento.

Para atingir tal objetivo este trabalho organiza-se em torno do desenvolvimento de um artefato, ou seja, uma solução artificialmente construída por seres humanos para endereçar uma situação-problema de um contexto específico (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015).

A ferramenta de *Design Thinking* será empregada no processo de desenvolvimento do artefato: começando com a etapa de descoberta para selecionar a comunidade e identificar problema, compreender o desafio e planejar a pesquisa. Em seguida, para a interpretação do contexto, foi realizada uma entrevista inicial com membros da comunidade e serão coletadas mais informações com auxílio do modelo unificado de aceitação e uso de tecnologias (UTAUT - *Unified Technology Acceptance and Use Theory*) para avaliar a experiência do usuário com a funcionalidade nativa de relatórios do ZABBIX. Então, na fase de ideação – fase de pensamento divergente foram geradas ideias para solucionar o problema em questão

(uso de diferentes linguagens de programação, frameworks e funcionalidades); finalmente, na experimentação e evolução o protótipo será desenvolvido, testado, ajustado e avaliado com base em nova avaliação usando as perguntas da escala UTAT (EDUCADIGITAL, 2014).

2 DESENVOLVIMENTO

2.1 Objetivos

O objetivo geral deste trabalho é desenvolver uma ferramenta web que utilize a API do Zabbix para gerar relatórios gráficos do histórico de falhas das redes monitoradas, utilizando apenas requisições do tipo GET.

Para atingir o objetivo geral, apresentam-se os seguintes objetivos específicos:

- Levantar as necessidades de melhorias ou implementação de ferramentas para o software Zabbix junto à comunidade Zabbix Brasil.
- Criar uma ferramenta web para a construção de gráficos a partir de falhas registradas no banco de dados do Zabbix.
- Desenvolver um laboratório de testes para a ferramenta.
- Implementar técnicas de acessibilidade na ferramenta.
- Executar testes manuais e automatizados.
- Hospedar a ferramenta na nuvem da AWS.
- Disponibilizar a ferramenta para uso da comunidade Zabbix Brasil.

2.2 Justificativa e delimitação do problema.

O problema em torno do qual esta pesquisa se desenvolve pode ser enunciado sob a forma da seguinte pergunta:

Como melhorar a experiência do usuário com os relatórios gráficos do histórico de falhas das redes monitoradas, utilizando a API do Zabbix?

Os Sistemas de Informações Gerenciais (SIG) são artefatos (ferramentas artificiais, desenvolvidas por pessoas e não criadas pela natureza) que dão informações para dar suporte à tomada de decisão. As informações são úteis no processo de decisão na medida em que, se bem estruturadas, associam os diversos sistemas e capacitam as organizações a atingirem seus objetivos (LACERDA, 2013).

Assim, conhecer e desenvolver tecnologias que auxiliem no gerenciamento, como os sistemas de informação é vital para as organizações atingirem seus objetivos e, por essa razão, constituem um campo de estudo fundamental em administração e gerenciamento de todos os

tipos de organizações (WAKULICZ, 2016, p.15), o que demonstra a importância acadêmica e prática desta pesquisa.

Para a comunidade em específico, desenvolver uma ferramenta que melhore a experiência do usuário para gerar relatórios gerenciais baseado em softwares abertos, permite economizar tempo e recursos, o que tem impacto prático para a comunidade em específico.

2. 3 Fundamentação teórica

Conforme descrito no manual do Zabbix, o software é um código aberto que pode monitorar inúmeros parâmetros de rede, bem como a integridade da rede e de seus servidores. Também é utilizado para monitorar máquinas virtuais, que é a opção utilizada neste trabalho através de máquinas virtuais Oracle e demais máquinas virtuais em sistema Windows 7. Ele também monitora banco de dados, websites, nuvem e aplicações.

O Zabbix ao monitorar e encontrar falhas no processo, aciona um mecanismo flexível de notificação que permite aos usuários configurar as notificações por meio de email, que detalha os erros apresentados. O que é a proposta do Projeto Integrador, pois através da notificação e podendo a coleta de dados ser programada, possibilita aos usuários fazerem intervenções e solucionar os problemas, o que facilita a compreensão se os mesmos fossem em gráficos.

O Zabbix Server já tinha a capacidade de gerar relatórios, porém segundo o entrevistado da comunidade Zabbix Brasil do Telegram, o que fora perguntado “quais as características do Zabbix pode ser melhorada?”, tendo como resposta pelo mesmo que o gráfico ficava a desejar. O mesmo produzia relatórios e estatísticas, e utiliza dos parâmetros de configuração que são acessados via frontend na web, garantindo assim que o status de sua rede e a integridade de seus servidores possam ser avaliados e monitorados de qualquer lugar. E pensando ainda em tempos de pandemia onde teve uma grande migração de negócios que eram físicos e agora são online, dando assim a praticidade ao usuário de acesso e intervenção de qualquer lugar que tenha internet.

No nosso caso, criamos um sistema com poucos servidores para que fossem analisados (fluxo de dados) e monitorados usando o Zabbix Server e coletamos informações sobre o desempenho de cada host (servidor) que nos possibilitou observarmos um servidor que deixamos propositalmente com menos memória para que o mesmo gerasse relatórios e que pudessemos não só gerarmos os relatórios, mas também gerarmos os gráficos, o que era a principal queixa da comunidade. Para que pudesse ser monitorada foi utilizado o Zabbix agent, sendo o programa escrito em linguagem C, pois o Zabbix agent se comunica através dessa linguagem.

O Zabbix ainda tem um função que permite ao usuário definir limites de problemas muito flexíveis, tendo como referência do banco de dados de backend.

2.3.1 Sistemas de Informações Gerenciais

Sistema de Informações Gerenciais se propõe fornecer informações de equipamentos remotos ou locais que estejam conectados aos serviços de internet, informações estas geradas a partir do agente do sistema Open Source Zabbix com a finalidade de permitir o gerenciamento e monitoramento dos equipamentos de tal forma que se tenha informações atualizadas e de qualidade sobre os equipamentos, permitindo assim identificar se determinado equipamento apresenta algum tipo de falha ou mesmo se apresenta sobrecarga para se possa tomar as medidas corretivas necessárias para que os equipamentos estejam sempre em atividade e condições adequadas de utilização.

2.3.2 Desenvolvimento de Software

O termo “software” contempla os programas de computador e sua documentação, desenvolvidos para resolver problemas específicos ou facilitar a tarefas de pessoas e organizações com segurança (SOMMERVILLE, 2019; CARVALHO; LORENA, 2017).

Para a adequada seleção de métodos e técnicas de engenharia de software a serem adotadas – conjunto de técnicas e ferramentas para apoiar o desenvolvimento sistemático de software para uso de outras pessoas além do próprio desenvolvedor qualidade (PRESSMAN e MAXIM, 2016) – é essencial levar em consideração o tipo de aplicação a ser desenvolvida. Para um sistema web, por exemplo, o mais adequado é “o desenvolvimento e entrega iterativos, em que o sistema é composto de componentes reutilizáveis” (SOMMERVILLE, 2019).

Em se tratando de um projeto web os objetivos geralmente são ter uma ferramenta que seja simples de navegar, intuitiva, informativa; que tenha consistência na formatação; que tenha alinhamento estético e estilo consistentes com o domínio da aplicação; que seja robusta e entregue ao usuário aquilo que se compromete a entregar; seja facilmente navegável pelo usuário; e que seja compatível com aplicativos usados para acesso (PRESSMAN e MAXIM, 2016).

2.3.3 APIs

API (*Application Programming Interface*) significa Interface de Programação de Aplicação, um conjunto de definições e protocolos para criar e integrar softwares de aplicações (RED HAT, 2017). São “tradutores” com a função de conectar sistemas, softwares e aplicativos. Dessa forma, é possível entregar uma experiência de uso mais familiar para as pessoas. As APIs permitem que o usuário final utilize um aplicativo, software ou até uma simples planilha, consultando, alterando e armazenando dados de diversos sistemas, sem que o usuário precise acessá-los diretamente (COSTA, 2022).

As APIs são funcionalidades “centrais em muitas arquiteturas de softwares modernos uma vez que fornecem abstrações de alto nível que facilitam tarefas de programação, apoiam o desenvolvimento de aplicações modulares e distribuídas e o reuso de código” (MENG; STEINHARDT; SCHUBERT, 2017, p. 296, tradução nossa) e são particularmente úteis para aplicativos em dispositivos móveis e aplicações web (PRESSMAN; MAXIM, 2016).

2.3.4 Zabbix e o gerenciamento de Redes

Gerenciamento de redes de computadores diz respeito à “[...] coordenação, controle de atividade, e monitoramento de recursos, assegurando [...] confiabilidade, segurança e alta disponibilidade” (BENÍCIO, 2015, P.20) e ocorre em três etapas: coleta de dados, diagnóstico e ação e controle. É uma atividade crítica em uma era digital como a que vivemos.

Segundo Black (2008) Zabbix é um programa de gerenciamento de redes que usa o tipo de agente-servidor, o que permite funcionamento de multi-servidor executando ao mesmo tempo de forma redundante, coletando dados gerados pelos agentes, executando em diversos clientes. O Zabbix possui uma arquitetura distribuída, com efeito não permite ter um gestor centralizado para manejar todas as informações. Os dados geridos são guardados em bancos de dados relacionais tais como *MySQL*, *PostgreSQL*, *Oracle*, etc. Os agentes estão disponíveis para *Unix*, *Linux*, *MacOs*, *Solaris*, *Windows*, entre outros.

A solução de código aberto Zabbix é constituída pelas seguintes partes (detalhadas na Figura 1):

- Agente Zabbix: um processo implantado em *hosts* (ou seja, alvos de monitoramento) que monitora de forma ativa os recursos e aplicativos locais e relata os dados coletados ao servidor Zabbix.

- Servidor Zabbix: um processo central do software Zabbix que realiza monitoramento, interage com proxies e agentes Zabbix, calcula triggers e envia notificações.
- *Frontend*: a interface web do Zabbix usada para configurar a coleta de dados e regras de alerta, visualizar dados e fazer chamadas de API.
- Banco de dados: o armazenamento de *backend* para todas as informações de configuração e os dados de monitoramento coletados pelo servidor Zabbix.

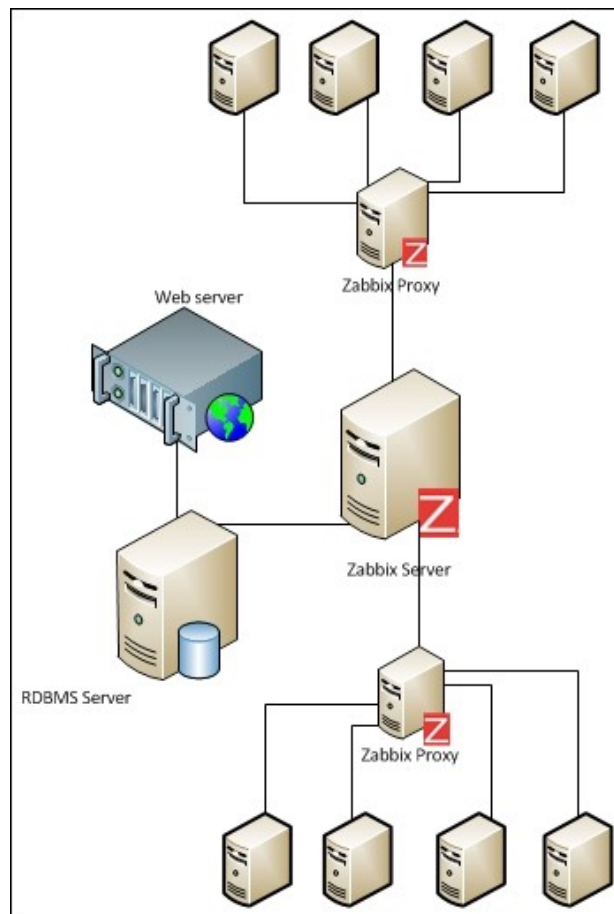


Figura 1 - Arquitetura do ZABBIX
Fonte: Black (2008)

2.3.5 Modelo de aceitação de tecnologias

O Modelo de Aceitação de Tecnologias foi proposto no final da década de 1980 para cobrir a falta de medidas confiáveis de fatores-chave para prever e explicar o uso de tecnologias por usuários, e tratar um problema antigo da área de TI/SI que é a impossibilidade de aproveitar melhorias de performance com o uso de tecnologias por falta de vontade de adoção por parte de usuários (DAVIS, 1989).

O modelo inicial propunha que a aceitação de um sistema de informação por um indivíduo é determinada fundamentalmente por duas variáveis, caracterizadas como crenças salientes (fatores motivacionais) (Moon & Kim, 2001). A primeira delas, a utilidade percebida (Perceived Usefulness) diz respeito à tendência de as pessoas usarem ou não uma tecnologia de acordo com o grau que acreditam que tais ferramentas podem ajudá-las a melhorar seu desempenho em determinada tarefa. Adicionalmente, o usuário pondera o potencial de melhoria de performance com o esforço despendido para usar a tecnologia, construto denominado facilidade de uso percebida (Perceived Ease of Use - PEOU) (DAVIS, 1989).

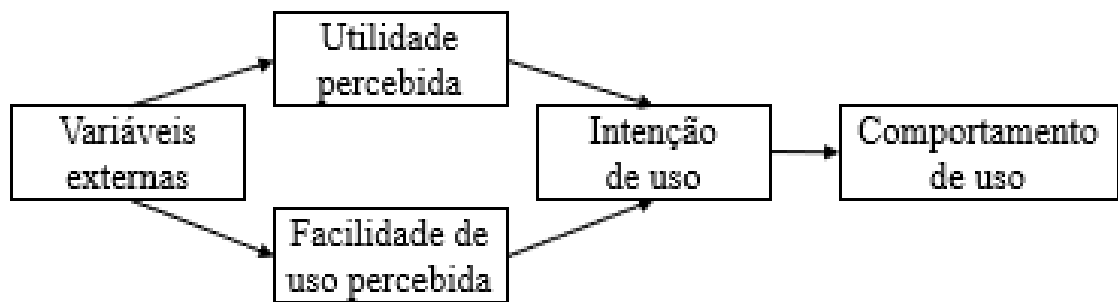


Figura 2 - Modelo de Aceitação de Tecnologia
Fonte: Adaptado de Lai (2017, p.28)

Em uma primeira extensão do modelo, no chamado TAM2, foram incluídas variáveis para explicar o que afeta a percepção do indivíduo sobre a utilidade da tecnologia na pré-implementação, um mês após a implementação e três meses após a implementação, avaliando os efeitos para os casos de adoção compulsória ou voluntária (LAI, 2017).

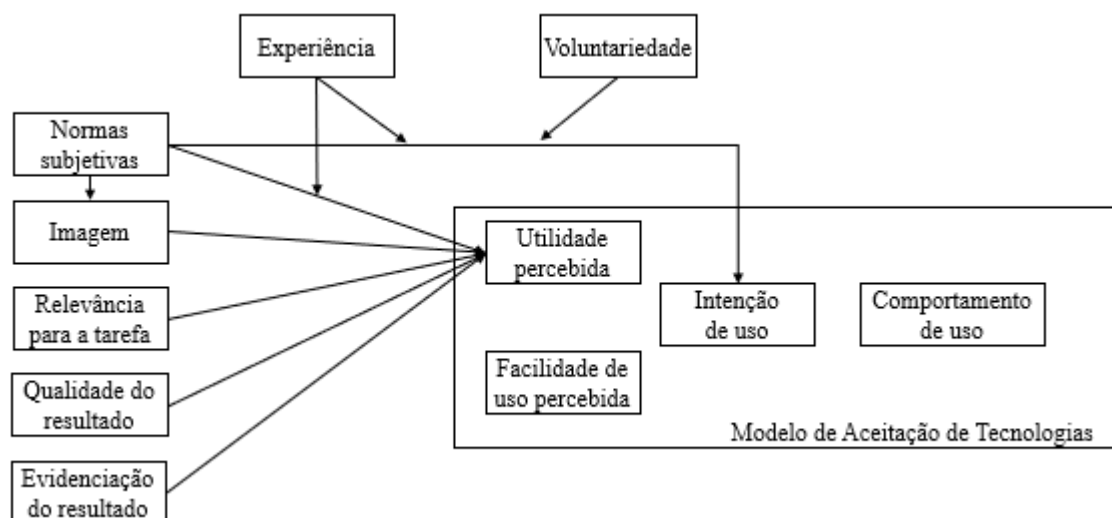


Figura 3 - Segundo Modelo de Aceitação de Tecnologia (TAM2)
Fonte: Adaptado de Lai (2017, p.29)

Finalmente, após outras modificações intermediárias, em 2003, foram incluídos novos preditores da intenção comportamental de uso de tecnologias (expectativa de desempenho, expectativa de esforço, influência social e condições facilitadoras), sendo a influência social não relevante em contextos de adoção voluntária de tecnologias. O modelo resultante, denominado Teoria Unificada de Aceitação e uso de tecnologia (UTAUT) é visto na Figura 4 a seguir e as perguntas dessa escala serão utilizadas para a coleta de dados sobre a funcionalidade da ferramenta proposta (LAI, 2017).

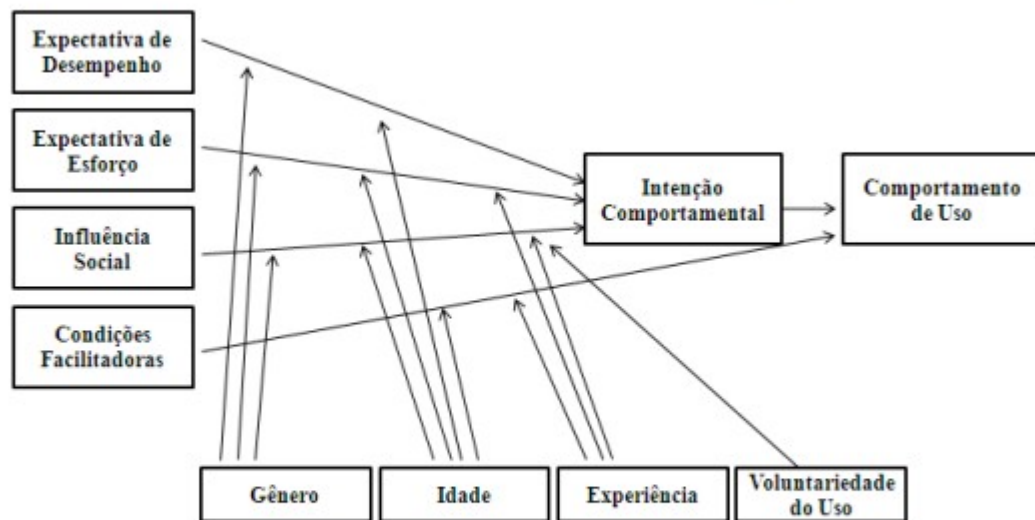


Figura 4 - Modelo Unificado de Aceitação e Uso de Tecnologia (UTAUT)
Fonte: Adaptado de Lai (2017, p.31)

2.4 Metodologia

2.4.1 Seleção e descrição do contexto

Para escolha da comunidade e do problema-alvo deste projeto, o grupo realizou um processo de *brainstorming* com a aplicação de reflexão divergente (geração de ideias) e convergente (seleção de ideias), característica do *design thinking* (LING, 2015). Segundo Tidd (2015, p. 108) “o *brainstorming* é uma técnica de grupo para gerar novas ideias e soluções em potencial.

Após uma reunião inicial, cada membro do grupo listou comunidades com problemas que gostariam de resolver. Nesse processo de divergência de ideias, com objetivo de gerar o máximo possível de alternativas, sem juízo de valor sobre a qualidade das ideias, surgiram as seguintes alternativas de problemas e comunidades a serem trabalhados:

- Criação de uma aplicação web para busca de editais de financiamentos de pesquisas em Administração, aplicando ferramentas de *webscrapping* para o problema de contingenciamento de recursos para pesquisa na comunidade acadêmica. Um membro do grupo é parte da comunidade, sendo membro de um grupo de pesquisa em Administração.
- Criação de uma aplicação web para identificar pequenos negócios de empreendedoras negras para o problema de inclusão e geração de renda para grupos sub-representados. Um membro do grupo é parte da comunidade, sendo membro de organização de funcionários de uma empresa de tecnologia que trabalham em iniciativas para comunidades sub-representadas.
- Construção de um app para auxiliar no controle de estoque de uma mercearia para o problema de controles gerenciais da comunidade de pequenos empreendedores. Um dos membros de grupo tem relacionamento com proprietários e funcionários de mercearias.
- Criação de uma aplicação web para monitoramento de redes através do ZABBIX para o problema de acessibilidade e experiência de usuário da comunidade de profissionais de gerenciamento de rede. Um dos membros do grupo conheceu a comunidade ZABBIX em um curso de redes de computadores e participa dos grupos de discussão do Telegram.

Na fase de seleção de ideias, cada alternativa foi avaliada com base em critérios de viabilidade técnica (expertise dos membros do grupo, tempo e recursos disponíveis), disponibilidade de acesso à comunidade em questão e aderência aos critérios de avaliação do PII. Então, a ideia escolhida foi a de criar uma aplicação web para monitoramento de redes utilizando o ZABBIX.

2.4.2 Interpretação do contexto

Com o ponto de partida definido, o segundo passo foi entrar em contato com a comunidade dos usuários do Zabbix para identificar potenciais problemas para atuação.

Para tanto, foi conduzido um levantamento de dados exploratório (COOPER; SCHINDLER, 2016) com auxílio de um instrumento de pesquisa desenvolvido na ferramenta Google Forms com as seguintes perguntas abertas:

- Quais as características do Zabbix podem ser melhoradas?
- Qual funcionalidade não-nativa do Zabbix melhoraria a funcionalidade da ferramenta?
- Qual o principal ponto forte do Zabbix?

O formulário foi compartilhado no grupo do Telegram Zabbix Brasil, que, atualmente, é o fórum com maior número de usuários do país.

Após uma semana, 15 usuários responderam ao questionário. As respostas foram consolidadas em uma planilha Google Sheets e analisadas utilizando codificação aberta, ou seja, um processo de divisão, conceituação e classificação dos dados coletados entre categorias pré-definidas ou que emergem dos dados (CASSIANI; CALIRI; PELÁ, 1996). A categoria que refletia melhor o tema norteado da disciplina Projeto Integrador em Computação II foi a carência de uma interface amigável para o desenvolvimento de relatórios no Zabbix.

Então, um sujeito cuja resposta fora enquadrada nessa categoria foi contactado para uma entrevista semi-estruturada para aprofundar o entendimento do problema e do contexto do usuário. O usuário trabalha em uma empresa que presta serviços de monitoramento de redes para outras empresas. Por questões de privacidade das informações com a quais trabalha, nos solicitou sigilo sobre seu nome e sobre o nome da empresa em que trabalha.

A partir da conversa com o usuário, a alternativa identificada pelo grupo para solução do problema foi a criação de uma ferramenta que pudesse ser disponibilizada aos clientes e fosse capaz de gerar relatórios gráficos a partir das informações de falhas registradas no banco de dados do Zabbix. Foi identificado que, embora ferramentas existam no mercado que fazem essa tarefa, como, por exemplo, o SolarWinds, essas ferramentas não são gratuitas e o alto preço da licença torna inviável a sua utilização.

Diante deste cenário, o tema já estava delimitado e os objetivos traçados, o projeto foi nomeado de Zabbix Graphics Reports.

2.4.3 Criação e prototipação: criação do Laboratório de Testes

Para o desenvolvimento do ZGR foi criado um laboratório de testes, com a finalidade verificar o funcionamento do software Zabbix e a forma como a sua API responde as requisições, o ambiente de testes é composto por 16 máquinas virtuais criadas e gerenciadas através do programa Oracle VM VirtualBox, sendo:

- 1 máquina com o sistema operacional Debian 11 instalado, nessa máquina foi instalada a versão 6.0 LTS do Zabbix, o banco de dados MySQL e o servidor web Apache. A máquina foi configurada para operar como Zabbix Server.
- 15 máquinas com o sistema operacional Windows 7, essas máquinas foram configuradas para serem monitoradas através do Zabbix Server, cada uma delas é denominada como um host, foram divididas em três grupos e cada grupo é denominado como um hostgroup.

Uma nova coleta de dados baseada nas perguntas da UTAUT está prevista para identificar a percepção da atual ferramenta de relatórios utilizada pelo usuário para então, compararmos com a percepção do usuário sobre a ZGR.

2.4.4 Execução do projeto: ferramentas de Gestão de Projetos

O grupo utilizou as seguintes ferramentas para organizar a elaboração do projeto:

- OneDrive – serviço de armazenamento de arquivos em nuvem, foi utilizado para centralizar o compartilhamento de mídias que foram utilizadas para o desenvolvimento do projeto e elaboração, em conjunto, da sua documentação;

- Collaborate - ferramenta de vídeo conferência disponível no ambiente virtual de aprendizagem, foi utilizado para reuniões com o orientador, com a comunidade externa e entre os membros do próprio grupo;
- GitHub – ferramenta de versionamento de códigos, foi utilizada como repositório de armazenamento dos códigos desenvolvidos e, também, para a sua própria finalidade que é o versionamento de códigos.
- Trello – ferramenta de quadros colaborativos, foi utilizada para a criação de um kanban para ajudar no gerenciamento e execução das atividades planejadas.

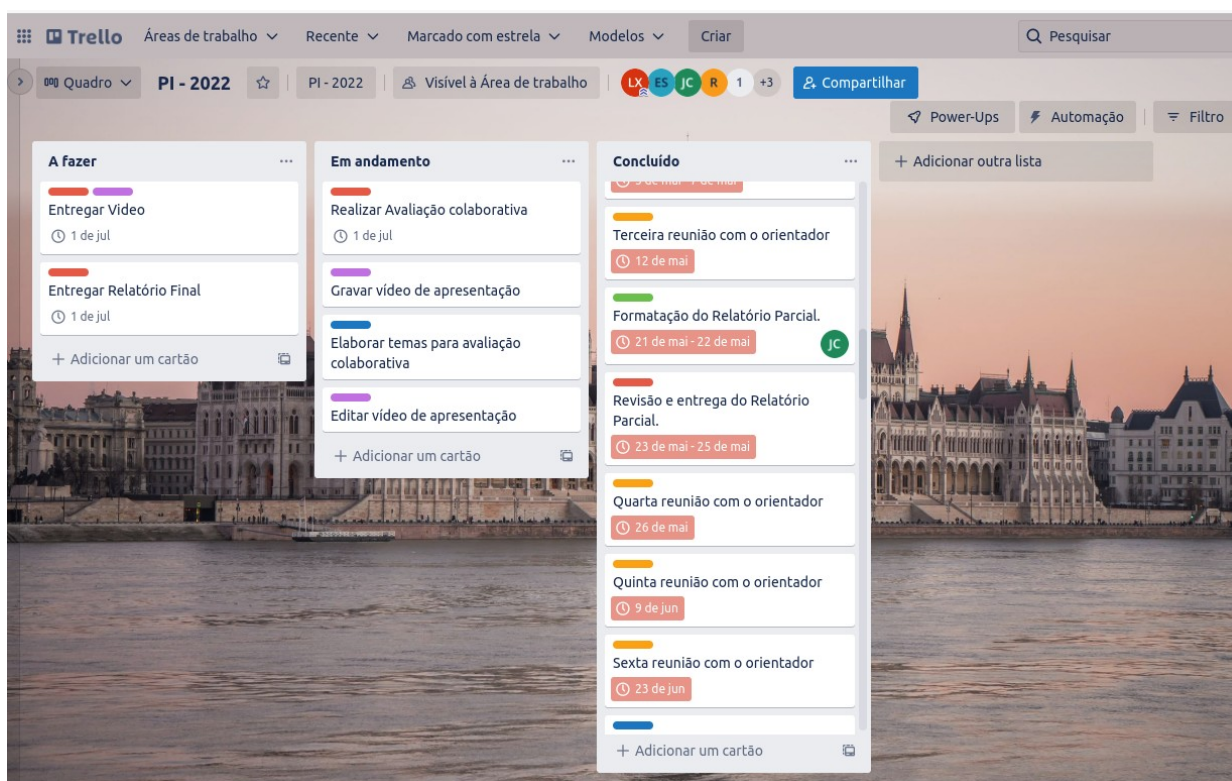


Figura 5 – Plano de ação no Trello
Fonte: Elaboração própria

3 RESULTADOS

Iniciamos o Projeto Integrador II ouvindo os problemas listados por cada integrante do grupo, que através de uma reunião cada um expos os problemas que gostariam de resolver, dessa maneira criando o maior número de ideias possível.

Após ouvir todos as ideias e avaliarmos os critérios de viabilidade para construção de cada ferramenta, o grupo em comum acordo optou por desenvolver uma ferramenta para monitorar redes que facilitasse a visualização e o entendimento dos relatórios extraídos da ferramenta ZABBIX.

3.1 Solução inicial

Inicialmente foi criado um login na ferramenta ZABBIX, onde através dela foi possível criar um grupo de hosts no qual gostaríamos de fazer o monitoramento.

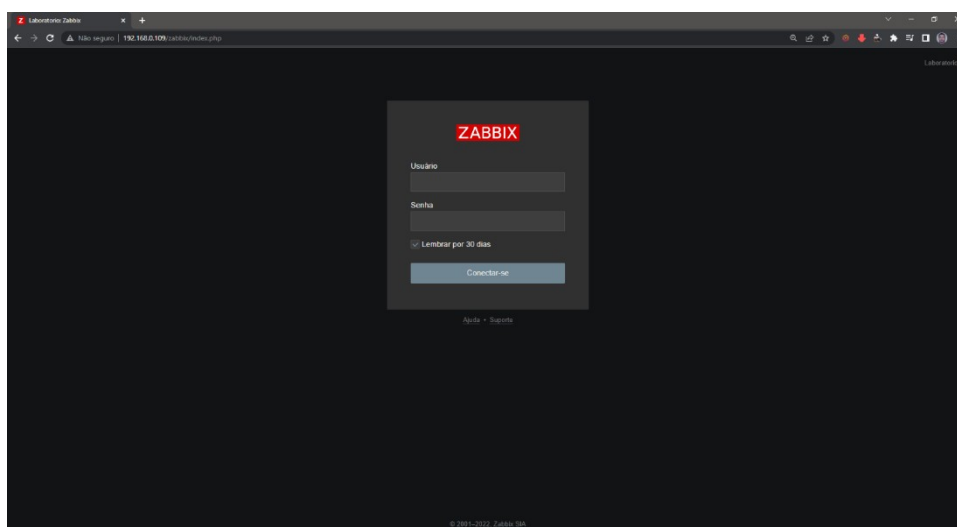


Figura 6 – Login do ZABBIX
Fonte: Elaboração própria

Para compor o grupo de hosts, o grupo optou por criar diversas máquinas virtuais onde pudessem ser simular diversos erros e falhas sem ter nenhum prejuízo. Abaixo a figura mostra um exemplo de máquina virtual criada.

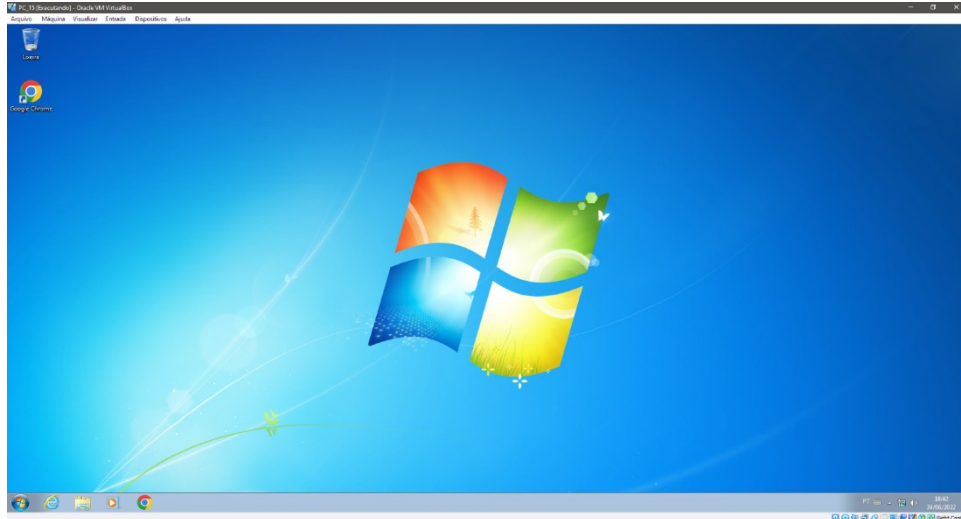


Figura 7 – Host em monitoramento pelo ZABBIX
Fonte: Elaboração própria

Desse modo, após criação das máquinas virtuais, foi criado um laboratório no VirtualBOX com o servidor ZABBIX e as máquinas a serem monitoradas.

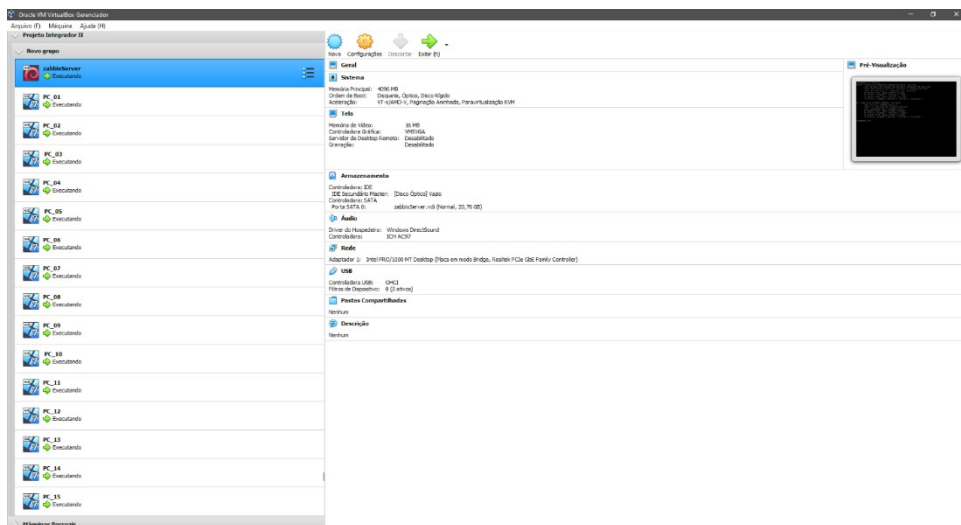


Figura 8 – Laboratório no VirtualBOX com o servidor ZABBIX
Fonte: Elaboração própria

Na ferramenta ZABBIX, foi realizado um monitoramento das máquinas criadas e extraído um relatório nativo da ferramenta para exemplificar a dificuldade de entendimento dos gráficos extraídos da própria ferramenta.



Figura 9 – Relatório gerado pelo ZABBIX
Fonte: Elaboração própria

Diante da dificuldade de entendimento do gráfico acima exposto, demos início a nossa ferramenta.

O primeiro passo, foi criarmos uma tela de login onde o usuário deve validar seu cadastro ZABBIX, consumindo a API do mesmo. Cabe ressaltar que se o usuário não utilizar dados corretos conforme cadastro no ZABBIX, aparecerá uma mensagem informando que os dados não conferem, conforme imagem abaixo.

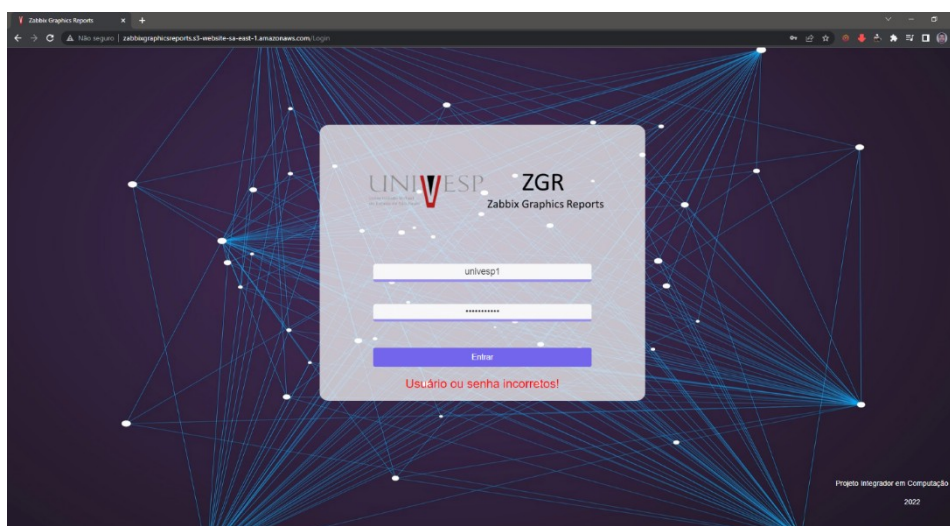
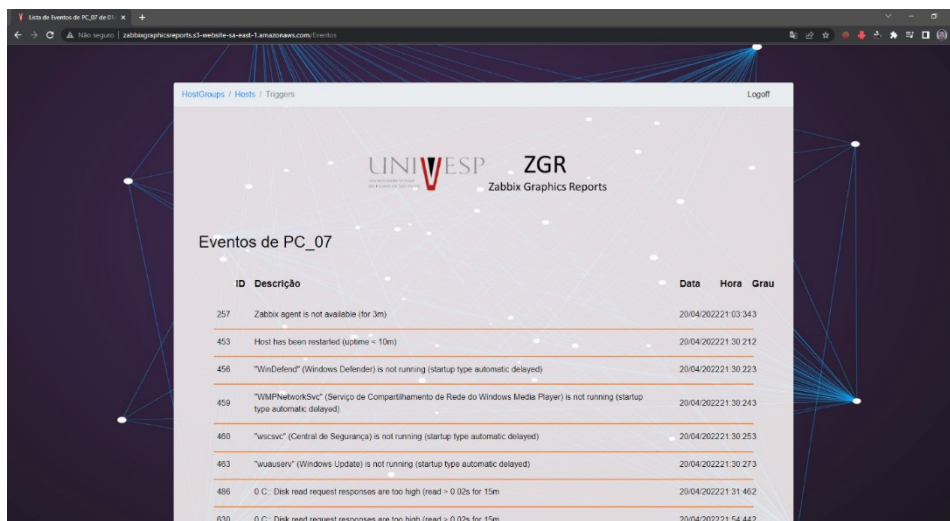


Figura 10 – Tela de Login para validação de Cadastro no ZABBIX
Fonte: Elaboração própria

Em sua primeira fase, a ferramenta era capaz de gerar uma lista contendo as falhas que cada host monitorado havia gerado.



ID	Descrição	Data	Hora	Grau
257	Zabbix agent is not available (for 3m)	20/04/2022	13:34:3	
453	Host has been restarted (uptime < 10m)	20/04/2022	13:21:2	
456	"WinDefender" (Windows Defender) is not running (startup type automatic delayed)	20/04/2022	13:22:3	
459	"WMPNetworkSvc" (Serviço de Compartilhamento de Rede do Windows Media Player) is not running (startup type automatic delayed)	20/04/2022	13:24:3	
460	"wscntsc" (Central de Segurança) is not running (startup type automatic delayed)	20/04/2022	13:25:3	
463	"wuauserv" (Windows Update) is not running (startup type automatic delayed)	20/04/2022	13:27:3	
486	0 C: Disk read request responses are too high (read > 0.02s for 15m)	20/04/2022	13:46:2	
630	0 C: Disk read request responses are too high (read > 0.02s for 15m)	20/04/2022	15:44:2	

Figura 11 – Lista contendo falhas de cada host no ZABBIX
Fonte: Elaboração própria

3.2 Solução Final

Após apresentação da ferramenta inicial criada à comunidade do ZABBIX, continuamos no desenvolvimento para que fosse possível extrair um relatório mais elaborado e um gráfico de melhor visualização.

Nesse passo foi desenvolvida uma tela onde é possível filtrar o período no qual devem ser listadas as falhas monitoradas.

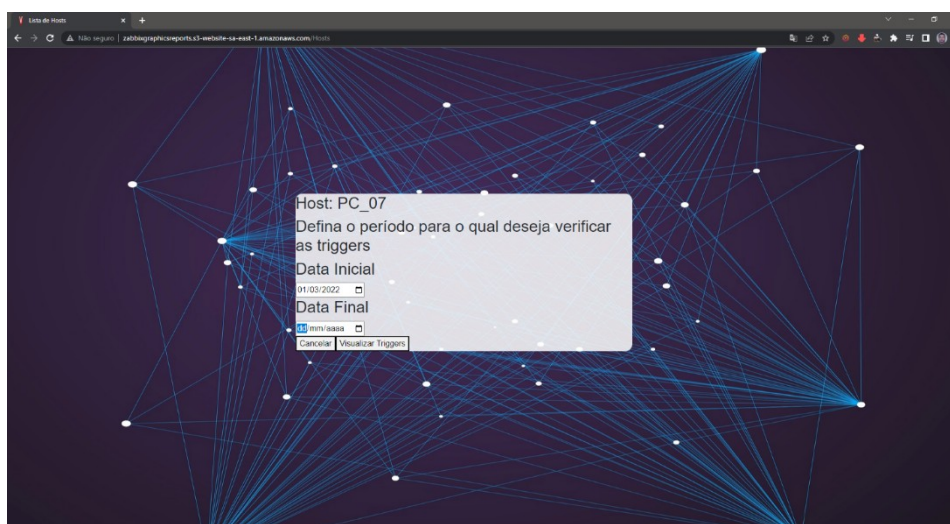


Figura 12 – Tela para seleção de período a ser listado
Fonte: Elaboração própria

Juntamente, foi criada uma tela que ao invés de mostrar a lista das falhas, possibilita a escolha do período para gerar os gráficos das falhas geradas.

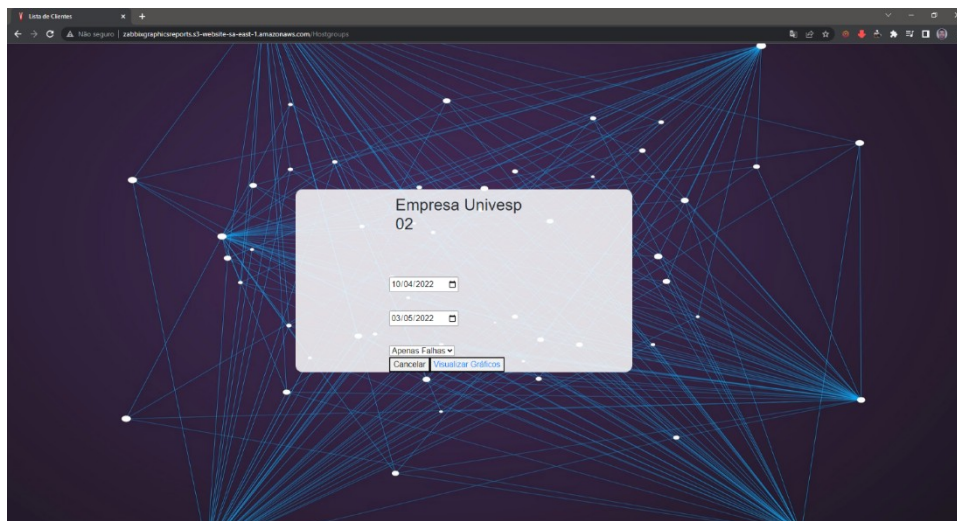


Figura 13 – Tela para seleção de período para geração do gráfico
Fonte: Elaboração própria

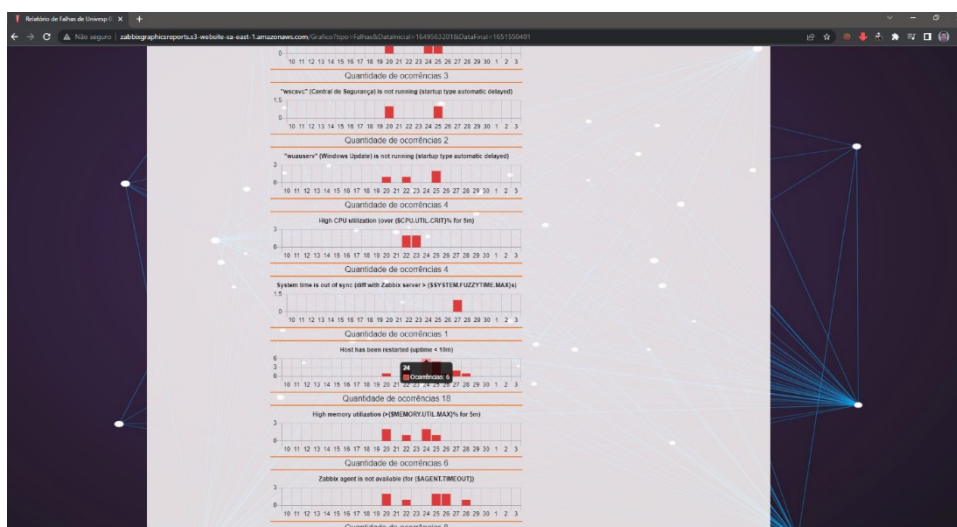


Figura 14 – Gráfico gerado
Fonte: Elaboração própria

Para finalizar foi desenvolvida a função que possibilita gerar um relatório composto por gráficos que mostra de maneira rápida e intuitiva as falhas geradas pela rede monitorada em um período determinado pelo usuário da ferramenta.

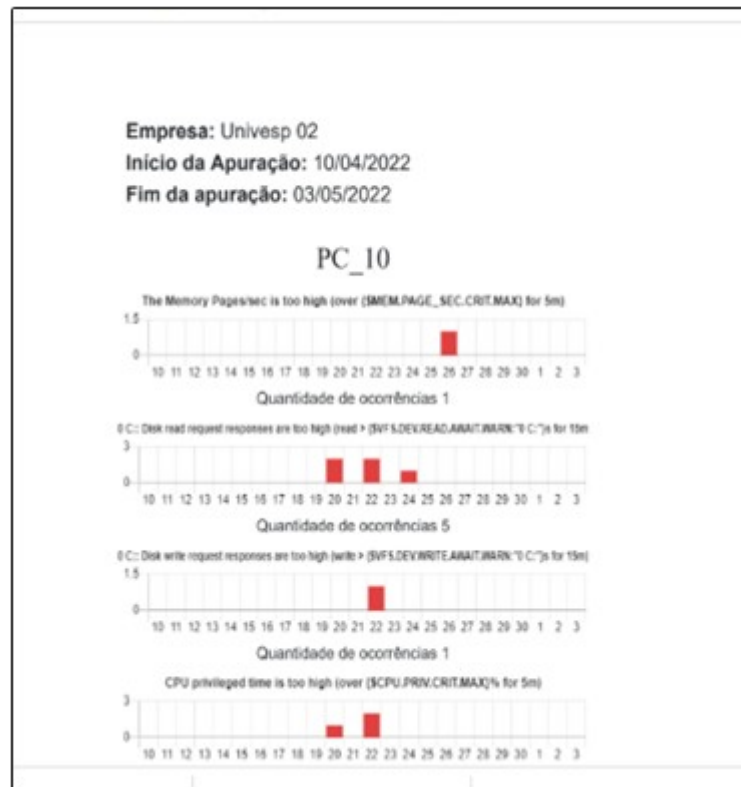


Figura 15 – Gráfico gerado
 Fonte: Elaboração própria

Todas as linhas de código foram feitas no Virtual Studio Code, conforme imagens abaixo:

```

import React, { Component } from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import logo from '../Assets/img/logo-technology-000108.png';
import login from '../Assets/img/tecnologia-login.png';
import api from '../Services/apiService';

import {
  Button,
} from 'react-bootstrap';

class Login extends Component {
  constructor() {
    super();
    this.state = {
      user: '',
      senha: '',
      userlogado: '',
      erro: ''
    };
  }

  componentDidMount() {
    document.title = 'Tela de Login';
    this.setState({ erro: '' });
  }

  handleSubmit(event) {
    event.preventDefault();
    let user = {
      username: '2.0',
      method: 'post',
      params: {
        user: this.state.user,
        password: this.state.senha
      },
      auth: null
    };
    fetch(api, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify(user),
    })
    .then(response => response.json())
    .then(data => {
      this.setState({ userlogado: data.token });
      // Se data.erro for false, redireciona o usuário e armazena no local storage o data.token
    });
  }
}

export default Login;
  
```

Figura 16 – Código fonte gerado utilizando o Visual Studio Code – Parte 1
 Fonte: Elaboração própria

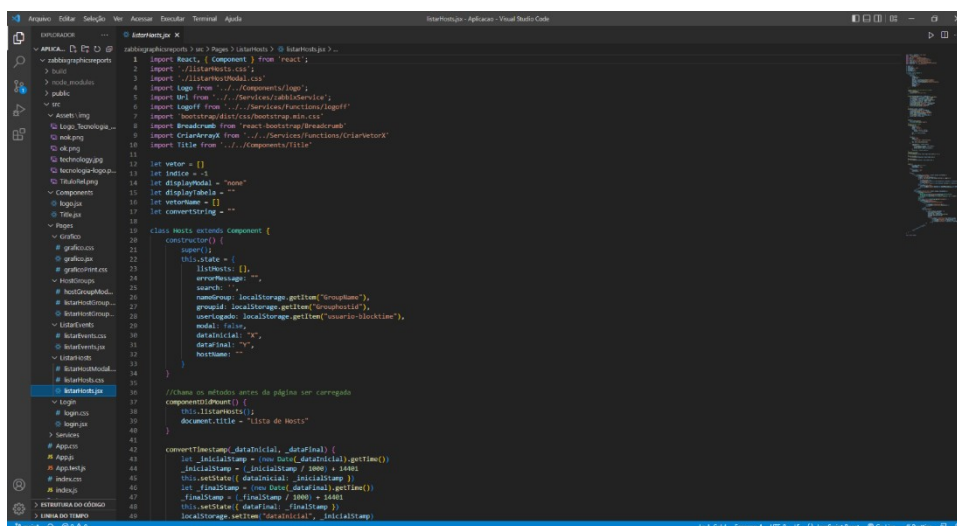


Figura 17 – Código fonte gerado utilizando o Visual Studio Code – Parte 2
Fonte: Elaboração própria

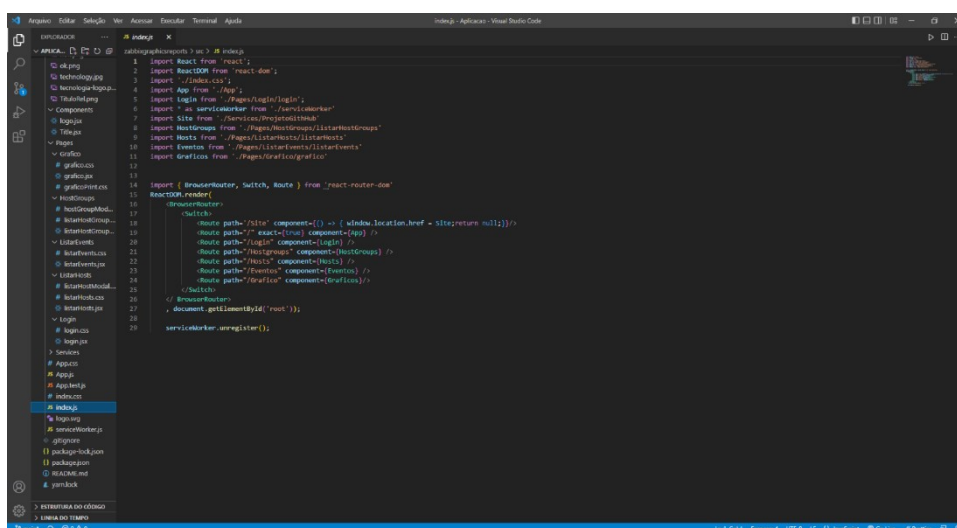


Figura 18 – Código fonte gerado utilizando o Visual Studio Code – Parte 3
Fonte: Elaboração própria

4 CONSIDERAÇÕES FINAIS

O presente estudo buscou analisar e propor uma solução compatível com um vasto número de protocolos/fabricantes diferentes, bem como deveria fornecer um ambiente de script extensível. Por fim, mas não menos importante que o software suportasse a visualização de dados, permitindo uma análise descritiva de dados a serem analisados.

A implantação do Zabbix foi executada em um período de 3 meses de projeto, tendo como base nas boas práticas de desenvolvimento ágil, utilizando o modelo de monitoramento com servidor em nuvem AWS. O ambiente de monitoramento flexível se ajusta de modo eficiente ao recurso de dimensionamento automático.

A comunicação entre servidores Zabbix, proxies e agentes é protegida por criptografia.

Após a implementação do Zabbix, surgiram vários problemas, que eram indetectáveis sem monitoramento, sendo possível identificar falhas de modo mais tempestivo e corrigi-las imediatamente acessando os dispositivos relevantes diretamente da interface do usuário.

Ademais, com alguns scripts externos, é possível simplificar o fluxo de trabalho e adicionar várias funções de conforto no Zabbix Map. A visibilidade integral de rede e dos dispositivos do cliente, está permitindo o conhecimento preditivo de qualquer interrupção antes do usuário final, afim de tomar as ações necessárias, habilitando também a supervisão e controle dos dispositivos com os dados visualizados na forma de gráficos. Gatilhos flexíveis foram implementados para serem notificados sobre as causas raiz do problema; Dessa forma obteve-se uma melhor desempenho e estabilidade do monitoramento.

REFERÊNCIAS

ABNT – Associação Brasileira de Normas Técnicas. NBR 14724: Informação e documentação. Trabalhos Acadêmicos - Apresentação. Rio de Janeiro: ABNT, 2002.

BLACK, T. L. **Comparação de Ferramentas de Gerenciamento de Redes**. Trabalho de Conclusão de Curso. Universidade Federal do Rio Grande do Sul. Porto Alegre: 2008. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/15986/000695315.pdf>. Acesso em 23 mai. 2022.

BENÍCIO, W.E.P. **Monitoramento e Gerenciamento De Redes Utilizando Zabbix**. Capivari: 2015. Disponível em: https://www.zabbixbrasil.org/files/Monitoramento_e_Gerenciamento_de_Redess_Utilizando_Zabbix.pdf. Acesso em 01. mai. 2022

CARVALHO; A.C.P.L.F; LORENA, A.C. **Introdução a Computação: hardware, software e dados**. Rio de Janeiro: LTC, 2017. 182 p.

CASSIANI, S. H. B.; CALIRI, M. H. L.; PELÁ, N. T. R. A teoria fundamentada nos dados como abordagem da pesquisa interpretativa. **Revista latino-americana de enfermagem**, v. 4, p. 75-88, 1996.

COOPER, Donald R.; SCHINDLER, Pamela S. **Métodos de Pesquisa em Administração- 12ª edição**. McGraw Hill Brasil, 2016.

COSTA, J. ZENVIA. **APIs: Tudo sobre o que é APIs, exemplos e importância!**. 27 de Janeiro de 2022. Página principal, Disponível em: < <https://www.zenvia.com/blog/apis-entenda-o-que-sao-e-como-funcionam/>>. Acessado em: 24 de mai. de 2022.

DRESCH, A.; LACERDA, D. P.; ANTUNES JÚNIOR, J.A.V.. **Design science research: método de pesquisa para avanço da ciência e tecnologia**. Bookman Editora, 2015.

EDUCADIGITAL. **Design Thinking para educadores**. Disponível em: https://www.dropbox.com/sh/hgnb8w3kbz1nhef/AACcd6QP66fBjlugTbdDti-Ia/DT_Livro_001a017.pdf?dl=0. Acesso em 19 nov. 2021.

ELMASRI, R; NAVATHE, S. B. **Sistemas de Banco de Dados**. Fundamentos e Aplicações. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

LACERDA, C. Sistema de Informação Gerencial. **Administradores.com**, Junho 2013. Disponível em:< <https://administradores.com.br/artigos/sistema-de-informacao-gerencial> >. Acesso em: 16 out. 2021.

LAI, Poey Chin. The literature review of technology adoption models and theories for the novelty technology. **JISTEM-Journal of Information Systems and Technology Management**, v. 14, p. 21-38, 2017.

LING, D. **Complete design thinking guide for successful professionals**. 2015.

MENG, M.; STEINHARDT, S.; SCHUBERT, A. Application programming interface documentation: what do software developers want?. **Journal of Technical Writing and Communication**, v. 48, n. 3, p. 295-330, 2018. <https://doi.org/10.1177/0047281617721853>.

MOON, Ji-Won; KIM, Young-Gul. Extending the TAM for a World-Wide-Web context. **Information & management**, v. 38, n. 4, p. 217-230, 2001. [http://doi.org/10.1016/S0378-7206\(00\)00061-6](http://doi.org/10.1016/S0378-7206(00)00061-6)

PRESSMAN, R.; MAXIM, B. **Engenharia de Software**. 8 ed. Porto Alegre: McGraw Hill Brasil, 2016.

RED HAT. O que é API?, 31 de outubro de 2017. Visão Geral. Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>>. Acesso em: 24 de mai. de 2022.

SOMMERVILLE, I. **Engenharia de software**. 10 ed. São Paulo: Pearson, 2019.

TEOREY, T. LIGHTSTONE, S.; NADEAU, T.. **Projeto e modelagem de banco de dados**. 2. ed. São Paulo: Elsevier, 2014.

WAKULICZ, Gilmar Jorge. **Sistemas De Informações Gerenciais**. Universidade Federal de Santa Maria, Colégio Politécnico, Rede e-Tec Brasil, 2016. Disponível em: <<https://central3.to.gov.br/a>>. Acesso em 16 out. 2021.

ZABBIX LLC (EUA). ZABBIX: Documentation. In: **Documentação do Zabbix: Manual do Zabbix**. [S. l.], 1 maio 2022. Disponível em: <https://www.zabbix.com/documentation/current/pt/manual>. Acesso em: 1 maio 2022.