



UFOP
Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

**Dizaê, plataforma para auxiliar os
processos de melhorias e manutenções
de espaços**

Pedro Henrique Sant'Ana

**TRABALHO DE
CONCLUSÃO DE CURSO**

**ORIENTAÇÃO:
Rafael Frederico Alexandre**

**Agosto, 2023
João Monlevade—MG**

Pedro Henrique Sant’Ana

Dizaê, plataforma para auxiliar os processos de melhorias e manutenções de espaços

Orientador: Rafael Frederico Alexandre

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Agosto de 2023



FOLHA DE APROVAÇÃO

Pedro Henrique Sant'Ana

Dizaê: Plataforma para Auxiliar os Processos de Melhorias e Manutenções de Espaços

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação

Aprovada em 04 de setembro de 2023

Membros da banca

Doutor - Rafael Frederico Alexandre - Orientador - Universidade Federal de Ouro Preto
Doutor - George Henrique Godim da Fonseca - Universidade Federal de Ouro Preto
Mestre - Tiago Franca Melo de Lima - Universidade Federal de Ouro Preto

Rafael Frederico Alexandre, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 08/11/2023



Documento assinado eletronicamente por **Rafael Frederico Alexandre, PROFESSOR DE MAGISTERIO SUPERIOR**, em 08/11/2023, às 09:35, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0620749** e o código CRC **D9ACD618**.

Dedico esse trabalho primeiramente a Deus, por me abençoar todos os dias. Aos meus pais e minha avó por sempre me apoiarem e me incentivarem ao longo de todo o curso. Por fim, dedico esse trabalho ao meu professor e amigo Rafael Alexandre, por todos os ensinamentos transmitidos e por sempre confiar no meu potencial.

Agradecimentos

Agradeço a Deus por sempre me abençoar, com muita força, saúde e sabedoria, para que conseguir desenvolver esse trabalho.

Agradeço aos meus pais Eliana e Edir, e minha avó Maria, pela educação que me deram e pelo amor que me dedicaram, sempre me incentivando, principalmente nos momentos difíceis, para que eu pudesse me empenhar e concluir mais essa etapa em minha vida. Sem o suporte e cuidado deles nada disso seria possível.

Agradeço também ao meu orientador Prof. Dr. Rafael Alexandre por todo o apoio e atenção durante a execução desse trabalho e também por confiar em mim para a execução deste e de vários outros projetos ao longo dos anos. As experiências e conhecimentos adquiridos durante essas oportunidades foram importantíssimos para o meu crescimento pessoal e profissional.

Por fim, agradeço a todos os meus familiares e amigos, que me acompanharam e me apoiaram ao longo desses anos.

“Information technology and business are becoming so tightly linked that you can’t talk about one without talking about the other.”

— Bill Gates,
in: Business @ the Speed of Thought.

Resumo

Os avanços das tecnologias trouxeram melhorias para os processos de manutenção e gerenciamento de ativos, atividades que estão diretamente relacionadas com os custos de uma organização, trazendo mais velocidade e praticidade, além de permitir uma melhor coleta e análise de dados úteis sobre os ativos. Este trabalho tem como propósito apresentar e documentar de forma clara, os desenvolvimentos e melhorias aplicadas à plataforma *Dizaê*, um *software* que tem como objetivos: auxiliar na interação entre os frequentadores e administradores de espaços físicos; e apresentar de forma prática as informações coletadas para os responsáveis, para que possam compreender melhor a situação dos ativos e as necessidades de manutenções. O trabalho se baseou na análise do sistema previamente desenvolvido, a fim de compreender o problema e verificar questões que poderiam ser aprimoradas; após as adaptações, o sistema foi implantado em um ambiente real, para verificar se realmente ocorreram as melhorias desejadas e, após um mês de utilização, os resultados coletados foram positivos, demonstrando a relevância do problema de pesquisa observado.

Palavras-chaves: Gerenciamento de Ativos; Manutenção; Informatização; Usabilidade

Abstract

Technological advances have brought improvements to asset management and maintenance processes, activities that are directly related to an organization's costs, bringing more speed and practicality, in addition to allowing better collection and analysis of useful data on assets. The purpose of this work is to clearly present and document the developments and improvements applied to the *Dizaê* platform, a software that aims to assist in the interaction between visitors and administrators of physical spaces; and present the collected data in a practical way to the admins, so that they can better understand the condition of the assets and the maintenance needs. The work was based on the analysis of the previously developed system, in order to understand the problem and verify issues that could be improved; after the adaptations, the system was deployed in a real environment, to verify if the changes had the desired effects and, after a month of use, the collected results were positive, demonstrating the relevance of the observed research problem.

Key-words: Asset Management; Maintenance; Informatization; Usability

Lista de ilustrações

Figura 1 – Diagrama de um sistema cliente-servidor	20
Figura 2 – Diagrama da <i>Clean Architecture</i>	22
Figura 3 – Diagrama do fluxo de funcionamento de um relato	24
Figura 4 – Diagrama demonstrando a interação entre as entidades do sistema <i>Dizaê</i>	25
Figura 5 – Diagrama demonstrando a interação cliente-servidor na plataforma <i>Dizaê</i>	27
Figura 6 – <i>QR Code</i> gerado pela plataforma	30
Figura 7 – Tela Login da aplicação <i>web</i>	31
Figura 8 – Tela Cadastro de Empresa e Administrador da aplicação <i>web</i>	31
Figura 9 – Tela de Local da aplicação <i>web</i>	32
Figura 10 – Tela de Local da aplicação <i>web</i>	32
Figura 11 – Tela <i>Home</i> da aplicação <i>web</i>	33
Figura 12 – Tela de Cadastro de Administrador da aplicação <i>web</i>	33
Figura 13 – Tela de Mensagens da aplicação <i>web</i>	34
Figura 14 – Tela de Mensagens da aplicação <i>web</i>	34
Figura 15 – Tela de Mensagens da aplicação <i>mobile</i>	35
Figura 16 – Tela de Mensagens do Relato da aplicação <i>mobile</i>	36
Figura 17 – Tela de Cadastro de Relato da aplicação <i>mobile</i>	37
Figura 18 – Tela de Detalhes da aplicação <i>mobile</i>	38
Figura 19 – Tela de Relato da aplicação <i>web</i>	38
Figura 20 – Tela de Relato da aplicação <i>web</i>	39
Figura 21 – Tela de Sugestões da aplicação <i>mobile</i>	40
Figura 22 – Opções de filtros de relatos	41
Figura 23 – Opções de filtros da Tela <i>Home</i> da aplicação <i>web</i>	41
Figura 24 – Gráfico anual de relatos	43
Figura 25 – Gráfico mensal de relatos	43
Figura 26 – Gráfico anual de relatos por categorias	44
Figura 27 – Gráfico mensal de relatos por categorias	44
Figura 28 – Gráfico mensal de relatos por prioridade	44
Figura 29 – Gráfico mensal de relatos por local	45
Figura 30 – Gráfico anual de relatos	45
Figura 31 – Gráfico mensal de relatos por categorias	45

Lista de abreviaturas e siglas

ADM Administrador

APIs Application Programming Interfaces

AWS Amazon Web Services

DBMS Data Base Management System

DDoS Distributed Denial of Service

EPIs Equipamentos de Proteção Individual

GPUs Graphics Processing Units

HTTP Hypertext Transfer Protocol

IA Inteligência Artificial

IoT Internet of Things

S3 Amazon Simple Storage Service

SGBD Sistema de Gerenciamento de Banco de Dados

TI Tecnologia da Informação

TPUs Tensor Processing Units

UI User Interface

Sumário

1	INTRODUÇÃO	13
1.1	O problema de pesquisa	14
1.2	Objetivos	15
1.3	Metodologia	16
1.4	Organização do trabalho	16
2	REVISÃO BIBLIOGRÁFICA	17
2.1	Gerenciamento de ativos	17
2.2	Arquitetura de <i>Software</i>	18
2.2.1	Cliente-Servidor	19
2.2.2	<i>Clean Architecture</i>	21
3	DESENVOLVIMENTO	24
3.1	Visão Geral	24
3.2	<i>Back-end</i>	27
3.2.1	<i>Node.js</i>	27
3.2.2	<i>PostgreSQL</i>	28
3.2.3	<i>Amazon Web Services</i>	28
3.3	Aplicação <i>WEB</i> e <i>Mobile</i>	28
3.3.1	Tecnologias	29
3.4	Versão Inicial	29
3.5	Versão atual da aplicação	30
4	RESULTADOS	42
4.1	Contexto	42
4.2	Utilização	42
4.3	Avaliação	46
5	CONCLUSÃO	47
5.1	Trabalhos Futuros	47
	REFERÊNCIAS	49

1 Introdução

Ao longo dos anos, a evolução das tecnologias e automação de processos causaram um aumento da complexidade nas relações de mercado entre as empresas dos mais diversos sistemas de produção. Isso levou as organizações a enfatizarem a importância da manutenção para os mais variados processos (CAVALCANTE; ALMEIDA, 2005), buscando manter os níveis de produção estáveis, assim como reduzir custos e prejuízos provenientes de problemas e paradas não previstas.

Marcorin e Lima (2003) apontam que a qualidade do processo e dos produtos gerados é diretamente relacionada à qualidade das condições de trabalho e equipamentos, já que a deterioração em alguma parte da cadeia de produção pode levar a falhas, falhas estas que podem gerar grandes consequências, como o atraso da produção, retrabalho, ineficiência e/ou desperdício de insumos. Dessa forma, a manutenção é um ponto fundamental para que a organização consiga atingir bons resultados. Estas consequências podem ser ainda mais graves, já que são capazes de impactar a relação com o cliente, ao trazer uma visão negativa a respeito do funcionamento da empresa (CAVALCANTE; ALMEIDA, 2005).

Nesse sentido, os conceitos de **Manutenção Preventiva** e **Manutenção Preditiva** se mostram como opções bem relevantes em comparação com a **Manutenção Reativa**, que é realizada somente quando uma falha acontece. Baseadas em intervenções periódicas, as manutenções preventivas são geralmente pautadas em tempos previamente estabelecidos por fabricantes, sem considerar a real situação do equipamento (MARCORIN; LIMA, 2003), o que pode levar a certos desperdícios, por exemplo, ao trocar uma peça que ainda possuía um grande tempo de vida útil.

Já a manutenção preditiva, é um processo que geralmente exige uma mão de obra mais especializada e/ou aparelhos; *softwares* específicos, pois ela é baseada em medições e análises de históricos, com o objetivo não somente de prever falhas (SELCUK, 2017), mas também melhorar a qualidade, eficiência, disponibilidade e reduzir custos. Este é um conceito que já existe há muitos anos, porém, os avanços mais recentes das tecnologias foram responsáveis por tornar essa prática mais acessível (RAN et al., 2019), aperfeiçoando as atividades de detecção, isolamento, prevenção, monitoramento e previsão, fornecendo apoio para o setor responsável tomar melhores decisões. Ran et al. (2019) destacam os seguintes avanços de tecnologia e suas relações com processos de manutenção:

- Internet of Things (IoT): trazendo facilidade de coleta de dados de várias fontes. Man, Na e Kit (2015) comentam a respeito da utilização de IoT para o gerenciamento de ativos em ambientes hospitalares, e Compare, Baraldi e Zio (2019) abordam desafios para a integração entre IoT e a manutenção preditiva;

- Técnicas de *Big Data*: facilitando a transformação dos dados em informações úteis, [McMahon, Zhang e Dwight \(2020\)](#) discorrem sobre a utilização dessas técnicas para gerenciamento de ativos de uma ferrovia, e [Wan et al. \(2017\)](#) comentam a respeito do uso de *Big Data* como solução para o funcionamento da manutenção preditiva;
- *Deep Learning*: para auxiliar análises e tomada de decisão, [Serradilla et al. \(2022\)](#) fazem a análise a respeito do uso de *Deep Learning* na manutenção preditiva, considerando trabalhos na área e os principais processos utilizados, como detecção de anomalias, diagnóstico, prognóstico e mitigação;
- *Hardware*: os avanços em semicondutores, Graphics Processing Units (**GPUs**) e Tensor Processing Units (**TPUs**), permitiram os computadores realizarem processamentos mais complexos e em tempos reduzidos.

1.1 O problema de pesquisa

Como explicitado na seção anterior, a prevenção e principalmente a predição de manutenções, são fatores essenciais para manter a qualidade e disponibilidade das operações, reduzindo custos com ações corretivas, e para que elas possam funcionar da melhor forma, o uso de tecnologias é essencial. [Schneider et al. \(2006\)](#) comentam que as análises estatísticas a respeito das manutenções são positivamente impactadas pelo grande número de dados adquiridos em redes distribuídas. [Mobley \(2002\)](#) aponta que o uso correto das tecnologias disponíveis também é essencial para o sucesso, exemplificando uma situação em um contexto industrial onde bombas e engrenagens geralmente são analisadas de forma isolada e não como parte de um sistema integrado, deixando de avaliar o impacto de variáveis como velocidade e carga na hora das análises. [Neto \(2007\)](#) comenta sobre a importância da coleta e análise correta de dados, para que eles possam ser contextualizados e sintetizados em informações valiosas, em conhecimento. É também comentado por [Rossetti e Morales \(2007\)](#) explicando que, considerar a tecnologia por si só, como fator de sucesso, sem levar em conta a qualificação e experiência do usuário é um equívoco.

Com esse contexto em mente, é possível perceber que a existência de um sistema customizável (para permitir aos responsáveis cadastrarem e monitorarem todos os recursos que consideram importantes), de fácil acesso e utilização (para que os usuários não se sintam desmotivados de utilizar, dessa forma conseguindo captar mais informações), que apresente os dados de forma mais adequada para a realização de análises estatísticas das informações coletadas (adquirindo assim informações mais úteis para o planejamento e gestão de ativos), é um fator estratégico para as organizações. Portanto, esse trabalho se propõe em apresentar uma ferramenta que possa atender esses requisitos.

1.2 Objetivos

A proposta desse trabalho é continuar o desenvolvimento, agregar melhorias e correções para a plataforma **Dizaê** (TOMAZ, 2021), que é um sistema que busca auxiliar os frequentadores e administradores de espaços físicos tornando mais simples e menos burocrático o processo de relatar e gerenciar problemas nos ambientes das organizações, além de analisar informações específicas a cerca dessas ocorrências. Além disso, esse trabalho também consiste na implantação da ferramenta em um ambiente real, para validar se a solução proposta cumpre com os seus objetivos.

Este trabalho possui aos seguintes objetivos específicos:

- Desenvolver e adaptar todas as funcionalidades do sistema para ser possível sua disponibilização no mercado, onde múltiplas organizações possam utilizá-lo, pois na versão inicial, todas as funcionalidades e entidades foram construídas baseadas na existência de uma única empresa no sistema, de forma que, qualquer administrador conseguisse gerenciar todos os locais, itens, relatos e demais entidades do sistema;
- Melhorar e adicionar novas opções, como a sinalização de prioridade e sistema de troca de mensagens, para a interação entre os usuários do sistema, a fim de melhorar a qualidade da comunicação e resolução de problemas, permitindo que os usuários possam sanar dúvidas e gerar/receber *feedbacks*;
- Melhorar e adicionar mais opções à funcionalidade de gerar gráficos, adicionando a possibilidade de filtrar, agrupar e ordenar as informações que serão usadas nos gráficos, para que os responsáveis pelos ambientes físicos possam realizar análises melhores e mais específicas para o contexto que pretendem verificar;
- Agregar opções ao sistema a fim de reduzir a quantidade de relatos semelhantes a um problema previamente informado por outro usuário, com a utilização de um sistema de sugestões, que possibilita ao usuário selecionar e reagir a outros problemas relatados e ainda não resolvidos no mesmo local em que ele se encontra;
- Continuar o desenvolvimento da plataforma, realizando correções de problemas e adições de funcionalidades menores, como atualização das versões das tecnologias e bibliotecas utilizadas, e melhorias de desempenho (por exemplo, os gráficos gerados estavam sendo renderizados diversas vezes, por um problema que existia na versão mais antiga da biblioteca utilizada);
- Implantar o sistema em um ambiente real, dessa forma permitindo que os objetivos previamente mencionados sejam testados, avaliando se atingiram os seus propósitos;

- Realizar o registro do domínio e demais configurações necessárias, como adquirir a licença de desenvolvedor da *Google Play*, para posteriormente disponibilizar o sistema para uso geral pela *web* e *mobile* em dispositivos *Android*.

1.3 Metodologia

Para a execução desse trabalho, os seguintes passos foram definidos:

- Revisão da literatura e do sistema previamente desenvolvido, buscando compreender o problema, as tecnologias utilizadas, as soluções desenvolvidas, assim como encontrar pontos que necessitam de correções e melhorias;
- Incorporar novas funcionalidades a plataforma *Dizaê*, realizando as adaptações necessárias para que o sistema possa atender a múltiplas organizações, além de outras melhorias e correções identificadas no passo anterior;
- Análise e discussão das questões desenvolvidas no passo anterior, para validar se as mudanças implementadas tiveram o impacto positivo previsto ou se ainda precisam de novas alterações, assim como se existem novos pontos que podem ser desenvolvidos e melhorados.

1.4 Organização do trabalho

Este trabalho está organizado da seguinte maneira: o Capítulo 2 apresenta os principais conceitos para o planejamento e o desenvolvimento da solução. O Capítulo 3 aborda a estrutura da aplicação, com as principais entidades, o fluxo de funcionamento da aplicação, tecnologias utilizadas e algumas das telas do sistemas. No Capítulo 4 estão presentes resultados observados com a utilização da aplicação. O Capítulo 5 apresenta as considerações finais sobre o desenvolvimento do trabalho. E o Capítulo 6, contém as referências bibliográficas utilizadas para a elaboração do trabalho.

2 Revisão bibliográfica

Neste capítulo são apresentados alguns conceitos importantes para o desenvolvimento do trabalho. Também são citadas aplicações com propósitos semelhantes disponíveis no mercado, e suas distinções com a solução apresentada.

2.1 Gerenciamento de ativos

A gestão de ativos pode ser considerada como a administração dos bens que a empresa possui, como móveis, equipamentos, eletrônicos, máquinas, durante todo seu ciclo de vida, garantindo funcionamento adequado, respeitando padrões de segurança e trazendo retornos adequados ([SCHNEIDER et al., 2006](#)).

[Marcorin e Lima \(2003\)](#) explicam que na busca de vantagens competitivas e aumento dos lucros, as empresas podem desviar de questões importantes, ao realizar análises rasas focadas apenas em redução de custos e aumento de produção, não frisando nos pontos de qualidade e produtividade. Nesse ponto, a manutenção e gestão dos ativos de uma organização são um pontos essenciais, para que eles possam gerar resultados em todas as fases do seu ciclo de vida, mas para isso é necessário equilíbrio entre o desempenho, os custos e os riscos associados ([PANEGOSS; SILVA, 2021](#)).

Os ativos intangíveis, bens que não possuem existência física, como o conhecimento dos colaboradores, vem ganhando cada vez mais importância dentro das organizações, passando a ser considerados recursos fundamentais ([STEFANO et al., 2014](#)). A gestão dos ativos intangíveis ou gestão do conhecimento, para melhorar o gerenciamento eficaz de recursos dentro de uma organização, vem se apresentado cada vez mais importante para ganhar e manter os níveis de excelência nas funções. Os conhecimentos podem ser representados de diversas formas, sendo normalmente difíceis de serem documentados, pois geralmente são subjetivos a criação e entendimento das pessoas. Dessa forma, identificar, captar, avaliar, sistematizar e aplicar informações e conhecimentos ([STEFANO et al., 2014](#)), são ações essenciais para melhorar a competitividade das empresas no mercado.

O termo Indústria 4.0, surgiu nas últimas décadas, motivado pelos desenvolvimentos e avanços em Tecnologia da Informação (TI), que trouxeram diversos benefícios as empresas. Segundo [Santos et al. \(2018\)](#), esse termo indica um cenário cada vez mais competitivo, com uma procura cada vez maior por produtos personalizados, custos reduzidos, maior complexidade e qualidade, contando com tecnologias de ponta ligadas à internet. [Panegossi e Silva \(2021\)](#) indicam que esse ambiente possui espaço para desenvolvimentos relacionados a áreas como gestão de ativos, manutenção preditiva e automação, com o objetivo de

reduzir o tempo de paradas por falhas, dessa forma, aumentando o lucro. Nesse sentido, diversas ferramentas surgiram no mercado para auxiliar no processo de manutenção e gestão de ativos, dentre eles podemos destacar alguns, como o IBM Maximo¹, o SAP Service and Asset Manager² e o Oracle Maintenance³.

Essas três soluções possuem pontos em comum como suporte ao ciclo de vida completo, gestão de estoque, integração com outros dispositivos, planejamentos e execução de manutenções, gerenciamento de custos, entre outros (ORACLE, 2023). Dessa forma, elas possuem o foco em grandes organizações, necessitando de análises específicas dos casos de uso de cada organização, além de especialistas para a implantação, manutenção e treinamento dos funcionários, o que leva a maiores custos e tempo para o estabelecimento do sistema dentro da empresa. O mercado também conta com soluções para empresas de médio e pequeno porte, como o Infraspak⁴, Fractal One⁵ e MaintainX⁶, sendo mais práticas e possuindo menores custos que as soluções anteriores, porém ainda necessitando do apoio das empresas para realizar os cadastros e treinamentos necessários.

Assim, a plataforma *Dizaê* se apresenta como uma solução atraente a organizações de pequeno e médio porte, pois possui um funcionamento simplificado, onde os administradores podem cadastrar diretamente os locais e itens de interesse, permitindo selecionar a abrangência dos ativos de acordo com as necessidades internas. Para os usuários dos espaços físicos a utilização também é intuitiva, necessitando apenas um telefone e um cadastro básico para começarem a relatar e acompanhar problemas. A plataforma também disponibiliza opções relacionadas à gestão de ativos intangíveis (conhecimento) trazendo filtros e gráficos para que os administradores possam avaliar os dados captados no sistema de forma mais simplificada, podendo assim realizar melhores avaliações a respeito dos ativos da empresa, levando a menores custos em manutenção, substituições e reparos. Além disso, para melhorar a interação entre os usuários, disponibiliza um canal de comunicação direto entre eles, através do envio e recebimento de mensagens.

2.2 Arquitetura de *Software*

Não existe uma definição exata a respeito do conceito de Arquitetura de *Software*, alguns consideram como a estrutura a ser seguida, um projeto, outros como a prescrição, recomendações do que se fazer (RICHARDS; FORD, 2020). Uma forma de abordar esse conceito é de que a arquitetura deve considerar a estrutura do sistema, como principais

¹ Disponível em: <<https://www.ibm.com/products/maximo>>. Acessado em: 20 jul. 2023

² Disponível em: <<https://www.sap.com/products/scm/asset-manager.html>>. Acessado em: 20 jul. 2023

³ Disponível em: <<https://www.oracle.com/br/scm/maintenance/>>. Acessado em: 20 jul. 2023

⁴ Disponível em: <<https://web.infraspeak.com/pt-pt/gartner-software-manutencao>>. Acessado em: 10 set. 2023

⁵ Disponível em: <<https://www.fractal.com/pt-br/fractal-one>>. Acessado em: 10 set. 2023

⁶ Disponível em: <<https://www.getmaintainx.com/pricing/>>. Acessado em: 10 set. 2023

componentes e interações entre eles; características para que a aplicação funcione corretamente; decisões a respeito das restrições para o sistema e time responsável; e princípios de *design* que atuam como recomendações para o desenvolvimento da aplicação. Porém essa mesma abordagem pode ser considerada desatualizada em alguns anos. [Richards e Ford \(2020\)](#) apontam que um motivo para isso são as rápidas evoluções com relação ao processo de desenvolvimento de *software*. Apesar das diversas possibilidades de se definir a arquitetura do *software*, de forma geral ela utiliza de abstrações, padronizações, estilos e codificações, com o objetivo de descrever a estrutura básica do sistema ([VALIPOUR et al., 2009](#)).

Com o objetivo de apresentar, compartilhar e reutilizar conhecimentos, diversas áreas de engenharia passaram a adotar padrões, que geralmente descrevem práticas de organizações que foram testadas e bem sucedidas em implantações anteriores ([SOMMERVILLE, 2019](#)), além disso deve possuir os pontos fortes e fracos do padrão, para auxiliar na decisão de quando é recomendada sua utilização. Um padrão muito utilizado é da arquitetura em camadas, onde o sistema é organizado em módulos hierárquicos classificados como camadas, dessa forma dividindo a complexidade do sistema em partes menores ([VALENTE et al., 2020](#)), trazendo facilidades para entender o fluxo, realizar testes, reutilizar componentes, realizar substituições e adições, ao realizar as ações na camada necessária.

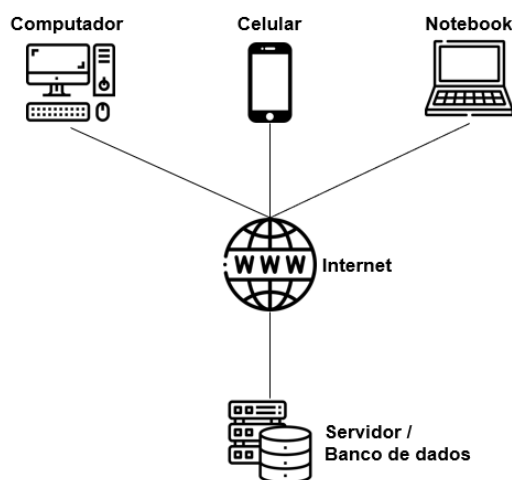
Atualmente, várias aplicações são desenvolvidas seguindo o conceito de Sistemas Distribuídos, sistemas que envolvem computadores que podem estar separados por pequenas ou grandes distâncias, comunicando e coordenando processos por meio de trocas de mensagens através da rede ([COULOURIS et al., 2013](#)). Nesse contexto, a ideia do compartilhamento de recursos é um ponto crucial para que essa distribuição ocorra de forma eficiente, recursos estes que podem ser de *hardware* como impressoras ou de *software* como os bancos de dados. [Tanenbaum e Steen \(2007\)](#) apontam a importância dos componentes interagirem com o sistema de forma consistente independente de onde estão, como se estivessem interagindo com um único sistema. Além disso, a escalabilidade e disponibilidade dos serviços devem ser tomadas como características fundamentais desses sistemas.

2.2.1 Cliente-Servidor

Na plataforma *Dizaê*, a comunicação entre os componentes é baseada na arquitetura Cliente-Servidor. [Sommerville \(2019\)](#) define aplicações desse tipo como sistemas distribuídos onde a interação é feita a partir da internet. Nesse tipo de arquitetura, o usuário interage com um programa, navegador ou dispositivo local, estes, que interagem com um servidor, que fornece serviços através de respostas aos requisitantes, que apresentam os resultados ao usuário. A [Figura 1](#) ilustra o funcionamento de um sistema cliente-servidor onde três clientes, representados por uma computador, um celular e um notebook se comunicam

com um servidor através da internet.

Figura 1 – Diagrama de um sistema cliente-servidor



Fonte: Produzido pelo autor

Oluwatosin (2014) explica que a comunicação entre processos nesses ambientes ocorre através da troca de informações entre os participantes, onde cada um realiza funções diferentes, com o cliente sempre enviando requisições e o servidor retornando as respostas necessárias. Com os avanços das tecnologias relacionadas a Computação em Nuvem (VELTE; ELSENPETER, 2010), mais empresas passaram a utilizar esses serviços e com isso a arquitetura cliente-servidor vem ganhando cada vez mais popularidade, ao trazer a possibilidade de escalabilidade horizontal para diversas máquinas virtuais. Os serviços do *Google Apps*, como o *Gmail*, são apontados por Oluwatosin (2014) como um exemplo de aplicações que usam essa tecnologia, que conta com vantagens como: persistência, eficiência e tolerância a falhas.

Uma parte importante para o funcionamento desse sistema são os *Middlewares*. Lewandowski (1998) caracteriza os *middlewares* como a rede existente entre o cliente e o servidor, que permite a comunicação entre eles de forma estruturada, atuando tanto no lado do cliente, com a inclusão de Application Programming Interfaces (APIs) (DE; DE, 2017), para fazer as requisições ao servidor por exemplo; quanto no lado do servidor, realizando o retorno da resposta ao cliente.

Nesse tipo de arquitetura, os sistemas geralmente são estruturados em camadas lógicas bem definidas, utilizando de interfaces para a realização da comunicação entre elas. Sommerville (2019) exemplifica esse modelo, com a utilização de quatro camadas contendo:

- Camada de apresentação: responsável pela interação do usuário e apresentação de informações para o mesmo;
- Camada de manipulação de dados (*middleware*): responsável pelo fluxo de dados

entre o cliente e o servidor;

- Camada de processamento da aplicação: responsável pela lógica da aplicação;
- Camada de banco de dados: responsável pelo armazenamento de dados.

Apesar dos benefícios, esse tipo de arquitetura também possui desafios para sua implementação eficaz, [Oluwatosin \(2014\)](#) destaca alguns pontos a serem observados, como: essas aplicações geralmente precisam de vários servidores para funcionar, podendo ser de difícil configuração, necessitando de mão de obra mais especializada e/ou demandar maiores gastos para garantir uma alta performance; questões de segurança, como danos físicos, vírus e ataques (por exemplo o Distributed Denial of Service (DDoS) ([HOQUE; BHATTACHARYYA; KALITA, 2015](#))) tanto com relação ao servidor, quanto na comunicação e acesso ao cliente.

A plataforma *Dizaê* foi construída com base nesse tipo de arquitetura, onde cada componente do sistema encontra-se de forma distribuída, com a comunicação entre eles acontecendo por meio da internet. Os usuários/clientes acessam a plataforma através de aplicativos móveis ou de uma página na internet, após o acesso, qualquer operação realizada comunica-se com o servidor *back-end* através da rede, este que processa a solicitação e envia uma requisição ao banco de dados e/ou banco de imagens, hospedados na nuvem, salvando e retornando os dados necessários para a operação.

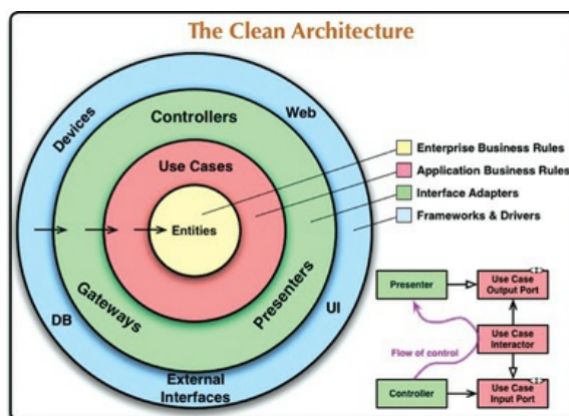
2.2.2 *Clean Architecture*

[Bui \(2017\)](#) comenta que diversos modelos de arquitetura foram propostos nos últimos anos, porém, apesar das diferentes conceituais e de implementação entre os modelos, eles apresentam um ponto em comum: a ideia da separação de responsabilidades nas aplicações. [Martin \(2017\)](#) explica que os sistemas desenvolvidos com base nesse princípio possuem algumas características em comum, como:

- Independência de *frameworks*: a aplicação não fica dependente de bibliotecas, com elas podendo ser substituídas para atender as necessidades do sistema, ao invés de forçar o *software* a se adaptar a biblioteca;
- Facilidade de testes: as camadas podem ser testadas de forma separada/independente;
- Independência de User Interface (UI): A interface do usuário pode ser substituída sem precisar alterar o restante da aplicação;
- Independência de Banco de Dados: As regras de negócio não são dependentes da base de dados, dessa forma é possível alterar o local de armazenamento.

Dessa forma, a *Clean Architecture* é apontada por (MARTIN, 2017) como uma tentativa de integrar essas diversas características e arquiteturas dentro de uma mesma ideia. Essa arquitetura é exemplificada em quatro camadas (o número de camadas não é limitado a somente quatro) representadas na Figura 2: *Enterprise Business Rules*; *Application Business Rules*; *Interface Adapters*; e *Frameworks and Drivers*. O funcionamento desse tipo de arquitetura é baseado na chamada Regra de Dependência, seguindo essa regra, uma camada mais interna não pode ter conhecimento de definições feitas em camadas mais externas, sendo independentes de mudanças que aconteceram nessas camadas (BUI, 2017). Devido a essas características, os conceitos da *Clean Architecture* foram utilizados para estruturar o servidor da plataforma Dizaê.

Figura 2 – Diagrama da *Clean Architecture*



Fonte: (MARTIN, 2017)

A camada mais interna é a *Enterprise Business Rules*, sendo responsável pelas principais regras de negócio da organização e/ou entidades do sistema (MARTIN, 2017). Essas regras/entidades, geralmente são independentes do ambiente externo, possuindo maior estabilidade e pouca probabilidade de mudança.

A camada *Application Business Rules*, contém todos os casos de uso da aplicação, responsáveis pelo fluxo de dados que vão e vem das entidades da camada anterior, mas sempre se mantendo isolada das outras camadas (MARTIN, 2017), ou seja, as mudanças nessa camada são diretamente relacionadas à mudanças no funcionamento do sistema, descritas pelos casos de uso.

Bui (2017) aponta a camada de *Interface Adapters* como a responsável por realizar conversões de dados enviados entre as camadas, para um formato mais adequado para a utilização nas camadas de destino.

No exemplo da *Clean Architecture* dividida em quatro camadas, a camada mais externa é chamada de *Frameworks and Drivers* (BUI, 2017), contendo bibliotecas e outras ferramentas, como banco de dados e interfaces gráficas, e geralmente possuem pouca dependência direta com o sistema, sendo mais fáceis de serem substituídas.

O servidor *back-end* da plataforma *Dizaê* foi desenvolvido com base nos princípios da *Clean Architecture*. Seguindo essa arquitetura, o código é dividido em camadas, como: *entities*, que definem as entidades da aplicação, como os administradores; *infra*, contendo funções para a conexão com o banco de dados, rotas e repositórios; *controllers*, responsáveis por processar a entrada e direcionar ao responsável por sua execução; *use-cases*, que são os casos de uso da aplicação.

A separação em camadas permite uma independência com relação aos módulos da aplicação, uma vez que, caso seja decidido mudar o local de armazenamento de imagens, bastaria implementar a conexão com o novo modo de armazenamento, e no serviço de *upload* de imagem, alterar do provedor atual para o novo, assim o resto da aplicação continuaria funcionando corretamente; facilitando também na realização de testes.

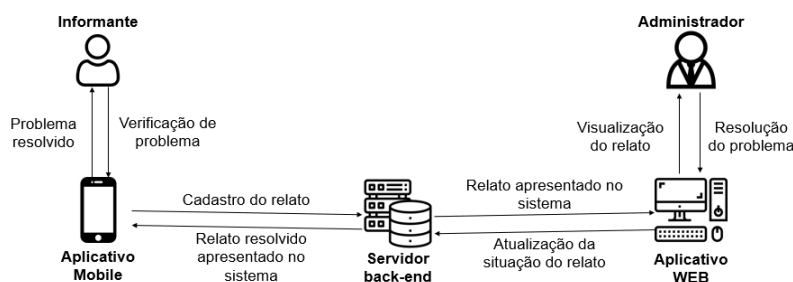
3 Desenvolvimento

Esse capítulo tem como objetivo apresentar ao leitor o funcionamento geral da plataforma *Dizaê*, explicitando as entidades necessárias para o seu funcionamento, tecnologias utilizadas para construção da aplicação dos clientes e do servidor, assim como algumas das telas desenvolvidas.

3.1 Visão Geral

A plataforma *Dizaê* foi inicialmente desenvolvida por Tomaz (2021), com o objetivo de facilitar a comunicação entre os administradores de espaços físicos e os frequentadores desses espaços, permitindo aos usuários relatarem problemas e acompanhar a sua situação, de uma forma mais rápida e prática. A Figura 3 representa um caso de funcionamento básico do sistema, não incorporando questões como a troca de mensagens entre os usuários, atualização de prioridade e geração de gráficos. Nesta figura, o utilizador de um espaço físico (Informante), verifica um problema e utiliza do seu celular para acessar o aplicativo e cadastrar o relato. Enquanto isso, o responsável pelo espaço físico, o Administrador (ADM), utiliza de um computador para acessar a plataforma e visualizar a falha informada. Após a resolução, o Administrador atualiza a situação do problema, que é transmitido ao Informante.

Figura 3 – Diagrama do fluxo de funcionamento de um relato

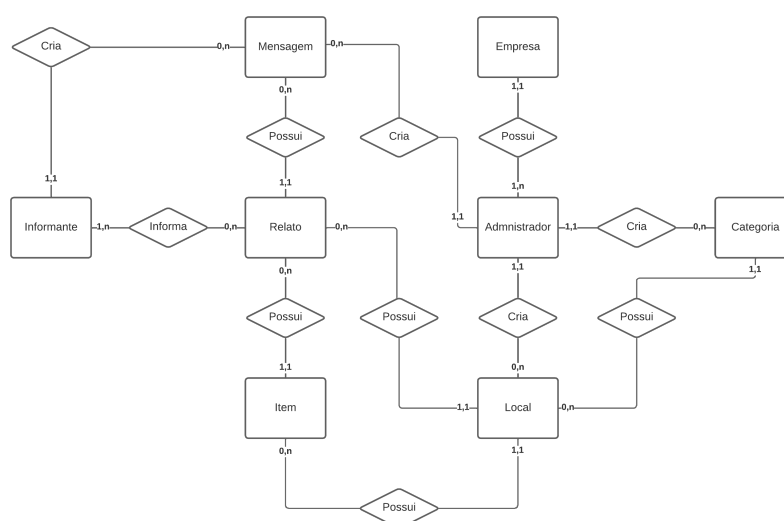


Fonte: Produzido pelo autor

A solução desenvolvida contava inicialmente com as seguintes entidades: **Administradores**, **Locais**, **Categorias**, **Itens**, **Informantes** e **Relatos**. Nessa versão, o único controle de acesso era com relação aos Informantes, que podiam visualizar somente as suas próprias informações e os Relatos cadastros por eles. Já com relação aos Administradores, eles não podiam ver as informações pessoais de outros administradores nem dos informantes, porém todas as outras entidades eram liberadas para qualquer ADM, não somente o que realizou o cadastro delas. Outro ponto era acerca da comunicação

entre os Informantes e Administradores, nesse momento a única interação entre eles era no momento da criação e na resolução ou rejeição do relato, não havendo opções para abordar dúvidas e/ou *feedbacks*. Além disso, não havia nenhum mecanismo para impedir ou diminuir a ocorrência de múltiplos relatos do mesmo problema. Com a intenção de resolver as limitações explicadas e outros pontos de melhorias identificados, na nova versão, além da adição de atributos nas antigas entidades, duas novas foram inseridas, sendo: **Empresas e Mensagens**. Na Figura 4 é possível perceber o relacionamento básico entre as entidades da aplicação.

Figura 4 – Diagrama demonstrando a interação entre as entidades do sistema *Dizaê*



Fonte: Produzido pelo autor

Os **Administradores** são os responsáveis pela manutenção dos espaços físicos nas organizações. Uma **Empresa** é cadastrada por um administrador, este que é considerado como o **responsável** pela organização, tendo maior controle sobre as entidades ligadas a ela, como a habilidade de cadastrar novos administradores para a empresa.

Os **Locais** são os espaços físicos da organização. Os administradores podem escolher o grau de granularidade em que dividem os espaços, que podem indicar desde espaços mais gerais, por exemplo, uma empresa pode ser dividida em andares, prédios, galpões; ou até mesmo espaços mais específicos, seguindo o mesmo exemplo, a divisão pode ser feita por cada sala no andar, cada cômodo de uma sala, entre outros. Cada local é de responsabilidade do administrador que realizou o seu cadastro, sendo somente visualizado por ele e/ou pelo **responsável** da empresa.

Os **Itens** são os elementos que serão monitorados pelos administradores, que estão associados a um local. Nem todos os itens de um local precisam ser cadastrados no sistema, eles podem ser escolhidos de várias maneiras, como valor econômico, importância para o funcionamento das funções, frequência de manutenções, entre outros. Computadores e

impressoras são exemplos de itens que possuem valor e importância para um funcionamento de um escritório. Outro exemplo seria o piso (chão), que apesar de não ter relação direta como o funcionamento de um escritório, a sua manutenção deve ser frequente, para evitar acidentes (caso ele esteja molhado ou escorregadio), mantendo sempre o local limpo e organizado.

As **Categorias** são uma forma de identificação atribuídas aos itens cadastrados, sendo utilizadas como opções para filtrar os dados na geração de relatórios.

Informantes são as pessoas que utilizam os espaços físicos. São eles que realizam o cadastro de relatos para que os administradores fiquem sabendo da necessidade de alguma manutenção em um ambiente.

Os **Relatos** são a base do sistema, eles são os registros da ocorrência gravada pelo informante. Cada relato possui: um título, que é o resumo do problema; o item em que foi constatado o problema; uma descrição, que detalha a situação; uma foto, para evidenciar a situação; a prioridade do informante para a resolução do problema, que pode ser Baixa, Média ou Alta. Cada relato pode estar em uma de três situações, ele pode estar: **Pendente**, que é o estado enquanto o problema ainda não foi resolvido; **Rejeitado**, quando o administrador considera que o problema não pode ser resolvido ou é dispensável; **Resolvido**, quando ele já foi solucionado. Além disso, cada relato possui um contador, identificado pelo campo **reações de usuários**, que demonstra quantos informantes constataram e reportaram o mesmo problema.

Por fim, existem as **Mensagens**, que são uma forma de comunicação ainda mais direta entre o informante e o administrador, podendo ser utilizada para tirar dúvidas, como em casos do administrador não compreender completamente alguma informação do relato, ou para dar algum aviso, por exemplo, um administrador pode avisar ao informante que o problema já foi constatado, mas que só poderá ser resolvido no dia seguinte.

O sistema é organizado da seguinte forma seguindo a divisão cliente-servidor: o servidor *back-end* da aplicação desenvolvido em *Node.js*¹; um cliente *web* desenvolvido em *React*², para que os administradores possam acessar o sistema, realizando os cadastros, acompanhamentos e análises dos espaços e ativos que são responsáveis; um cliente *mobile* desenvolvido em *React Native*³, para que os informantes possam se cadastrar, informar e acompanhar a resolução de problemas.

¹ Disponível em: <<https://nodejs.org/en>>. Acessado em: 20 jul. 2023

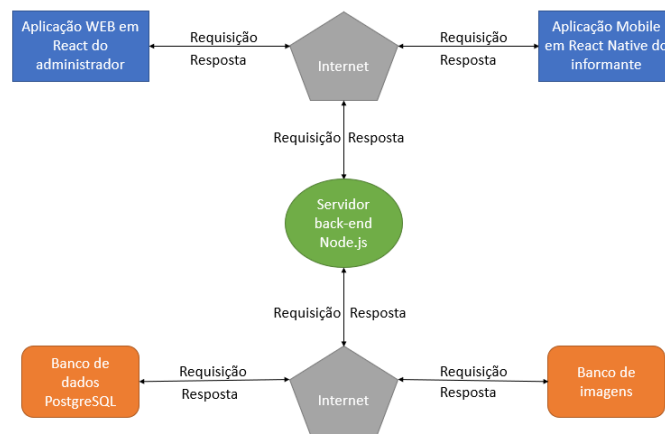
² Disponível em: <<https://react.dev/>>. Acessado em: 20 jul. 2023

³ Disponível em: <<https://reactnative.dev/>>. Acessado em: 20 jul. 2023

3.2 Back-end

O *back-end* da plataforma *Dizaê* é o responsável por executar as regras de negócio, garantindo integridade, segurança e controle correto do fluxo de informações da aplicação. A Figura 5 representa a interação entre as tecnologias aplicadas na construção da aplicação, o *Node.js* utilizado para o desenvolvimento do código *back-end*; o *PostgreSQL*⁴, escolhido como banco de dados do sistema; e o serviço de armazenagem de imagens hospedado através do Amazon Simple Storage Service (S3)⁵ ofertado pela Amazon Web Services (AWS). Nesta seção serão apresentados tópicos relacionados à escolha da linguagem de programação, organização do código, tecnologias utilizadas e mudanças com relação a versão anterior.

Figura 5 – Diagrama demonstrando a interação cliente-servidor na plataforma *Dizaê*



Fonte: Produzido pelo autor

3.2.1 Node.js

O *Node.js* é um ambiente de servidor gratuito e de código aberto, que realiza a execução de aplicações escritas de *JavaScript*, construído com base no motor V8 do *Chrome* (OLIVEIRA; ZANETTI, 2021). O *Node.js* possui uma grande variedade de bibliotecas construídas e mantidas pela comunidade, o que permite um desenvolvimento mais rápido de funcionalidades na aplicação.

O *Node.js* permite a construção de aplicações robustas e escaláveis (OLIVEIRA; ZANETTI, 2021), usando recursos da linguagem *JavaScript*, linguagem esta que é utilizada também nos componentes *front-end* e *mobile* do sistema. Estes motivos levaram à escolha do *Node.js* como a estrutura básica do *back-end*.

Além disso, não somente no *back-end*, mas também nos demais módulos da plataforma (*web* e *mobile*), foi utilizado o *TypeScript*, uma linguagem de programação fortemente

⁴ Disponível em: <<https://www.postgresql.org/>>. Acessado em: 20 jul. 2023

⁵ Disponível em: <<https://aws.amazon.com/pt/s3/>>. Acessado em: 20 jul. 2023

tipada, construída em *JavaScript* ([MICROSOFT, 2012-2023](#)), para facilitar o controle de tipagem das entidades e funções da aplicação.

3.2.2 PostgreSQL

Na sociedade, os bancos de dados, coleções de dados estruturados, possuem um papel crítico em quase todas as áreas em que os computadores são utilizados, desde medicina, comércio eletrônico, engenharia e ciências da informação, entre outras ([ELMASRI, 2002](#)).

Entre as diversas opções de bancos de dados relacionais, o *PostgreSQL* é um dos mais conhecidos, sendo um Sistema de Gerenciamento de Banco de Dados ([SGBD](#)) (conhecido como Data Base Management System ([DBMS](#)) em inglês ([CORONEL; MORRIS, 2022](#))) de código aberto, o que permite a sua evolução contínua através da colaboração da comunidade. Dessa forma, ele foi escolhido para armazenar todos os dados utilizados na plataforma *Dizaê*.

3.2.3 Amazon Web Services

A introdução da banda larga em grande escala em vários países, reforça a [TI](#) como alicerce importante do mundo baseado em informação ([VERAS, 2012](#)). Essa evolução da internet, permitiu o desenvolvimento da Computação em Nuvem, caracterizada por meio da utilização da internet ou outras formas de conexão remota para acessar aplicações, arquivos e/ou serviços ([SILVA, 2020](#)). Grandes organizações como *Google*, *Microsoft* e *Amazon*, oferecem serviços de computação em nuvem.

Para realizar o armazenamento de imagens na plataforma *Dizaê*, mais especificamente: as fotos de perfil dos usuários, as fotos colocadas como a evidência de cada relato e as fotos de cada item cadastrado no sistema; foi decidido pela utilização de serviços em nuvem, e dentre algumas opções disponíveis no mercado, o Amazon Simple Storage Service ([S3](#)), um serviço oferecido na Amazon Web Services ([AWS](#)), foi escolhido por ser uma solução segura e escalável, contando com soluções gratuitas durante um ano, facilitando a implantação do projeto. É importante destacar que, por seguir os princípios da *Clean Architecture*, o serviço de armazenamento de imagens pode ser substituído por outro, sem causar grandes impactos ao sistema, necessitando apenas da adaptação do módulo e serviço de *upload*, e da transferência das imagens entre os provedores.

3.3 Aplicação WEB e Mobile

A aplicação *web* é utilizada pelos administradores dos espaços físicos, para realizar o gerenciamento das entidades da aplicação. É através da Aplicação *mobile* que os informantes, usuários dos espaços físicos, tem acesso ao sistema, permitindo criar e acompanhar o

andamento dos relatos. Nesta seção serão apresentadas as tecnologias utilizadas nessas partes da aplicação.

3.3.1 Tecnologias

O *React* é uma biblioteca de código aberto construída em *JavaScript* pela equipe de desenvolvedores da *Meta*, sendo utilizada por grandes empresas, como a *Netflix*, *Uber* e *Airbnb*. Uma de suas grandes características é com relação a renderização de componentes, que definem a aparência e o comportamento das estruturas da tela, atualizando a página de acordo com as necessidades do usuário, mantendo elementos iguais e renderizando somente novos componentes (META, 2023a).

O *Next.js*⁶ é um *framework* para aplicações *web*, que utiliza o *React* como base para auxiliar no desenvolvimento, possuindo interfaces robustas para construção de *layouts* e controle de rotas, sendo utilizada por empresas como *Twitch* e *HBO Max* (VERCEL, 2023), além de ser compatível com o *TypeScript*.

O *Axios*⁷ é um cliente Hypertext Transfer Protocol (HTTP) baseado em promessas (um objeto *JavaScript*, contendo uma futura resposta de uma operação assíncrona), que pode ser utilizado tanto pelo *Node.js* quanto pelo navegador (PROJECT, 2014). No caso do *Dizaê*, ele é utilizado pelo cliente para realizar as requisições ao servidor *back-end*.

De forma semelhante ao *React*, o *React Native* é uma biblioteca baseada em *JavaScript*, desenvolvida pelo *Meta*, para desenvolvimento de aplicativos, tanto para sistemas *Android*, quanto para *iOS*, funcionando como aplicações nativas (aquelas construídas com linguagens de programação próprias de cada plataforma, como o *Kotlin*⁸ para *Android* e *Swift*⁹, para *iOS*), utilizando uma única base de código (META, 2023b).

3.4 Versão Inicial

A primeira versão da plataforma *Dizaê* desenvolvida por Tomaz (2021), tinha como foco a utilização para uma única organização, dessa forma todas as entidades, telas e dados, podiam ser observadas por qualquer administrador do sistema, nessa situação, o responsável por implantar a plataforma cadastrava todos os administradores.

O fluxo base da aplicação funciona da seguinte forma: Os administradores realizam o login através da aplicação *web* e cadastram locais, categorias e itens, estes que são acessíveis para todos os outros administradores do sistema e, após isso podem gerar e imprimir *QR Codes* para cada local cadastrado. O usuário de algum espaço relacionado

⁶ Disponível em: <<https://nextjs.org/>>. Acessado em: 20 jul. 2023

⁷ Disponível em: <<https://axios-http.com/ptbr/>>. Acessado em: 20 jul. 2023

⁸ Disponível em: <<https://developer.android.com/kotlin>>. Acessado em: 20 jul. 2023

⁹ Disponível em: <<https://www.apple.com/swift/>>. Acessado em: 20 jul. 2023

aos administradores do sistema, realiza o login através do aplicativo *mobile*. Ao realizar a leitura do *QR Code*, o informante visualiza todos os itens cadastrados naquele local e para finalizar o relato, é preenchido o título, descrição e foto. Em seguida, todos os administradores do sistema poderão visualizar esse relato na tela *Home*, onde podem acessá-lo e mudar seu *status* para Rejeitado ou Resolvido. Um exemplo de *QR Code* gerado pelo sistema pode se visualizado na [Figura 6](#).

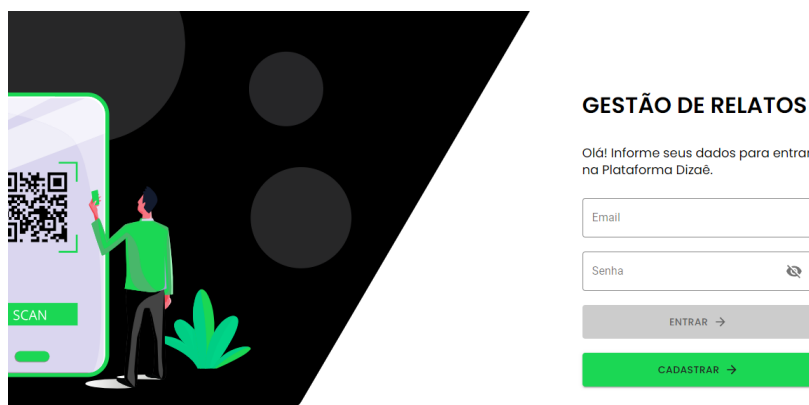
Figura 6 – *QR Code* gerado pela plataforma



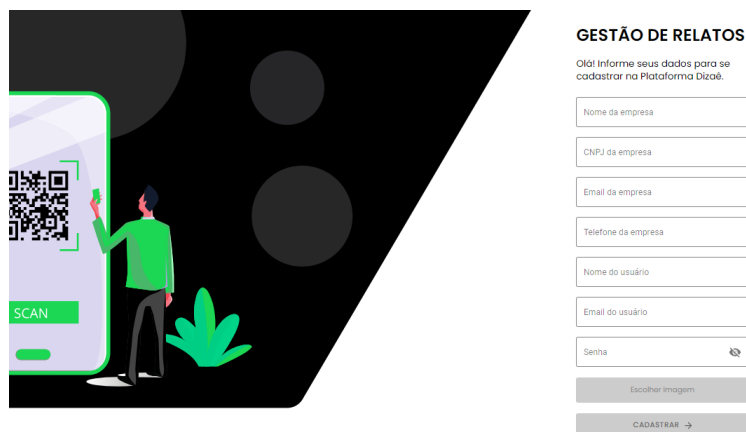
Fonte: Produzido pelo autor

3.5 Versão atual da aplicação

Na tela inicial da aplicação representada na [Figura 7](#), é possível acessar a nova opção de cadastro, identificada pelo botão **Cadastrar**, onde ao ser acessada, é possível preencher as informações básicas do **Administrador**, que são nome, e-mail, senha e imagem de perfil, assim como as informações para cadastro da **Empresa**, a nova entidade presente no sistema, que possui nome, e-mail, CNPJ e telefone. Essa tela pode ser visualizada na [Figura 8](#).

Figura 7 – Tela Login da aplicação *web*

Fonte: Produzido pelo autor

Figura 8 – Tela Cadastro de Empresa e Administrador da aplicação *web*

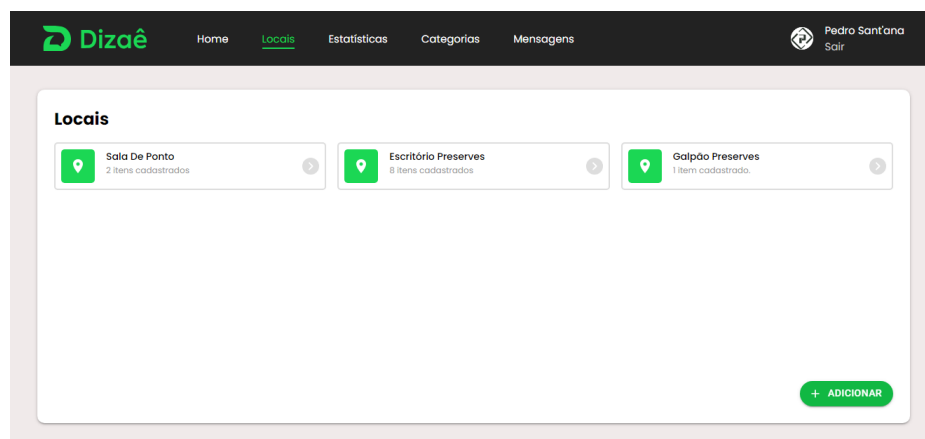
Fonte: Produzido pelo autor

A **Empresa** é a base para o funcionamento da nova versão. Todas as entidades do sistema, com exceção dos **Informantes**, são relacionadas a uma empresa, com o objetivo de realizar o controle de acesso das informações armazenadas no sistema, para que ele possa funcionar em um ambiente real, onde diversas empresas poderão utilizar a aplicação. Antes de explicar o funcionamento do controle de informações, outro conceito importante é a existência do **responsável**, este que é o administrador que foi cadastrado juntamente com a empresa, possuindo mais permissões que os outros administradores da mesma organização.

Na plataforma, os administradores só possuem acesso às informações que são relacionadas à mesma empresa que eles estão associados. Além disso um administrador normal tem acesso somente aos dados que ele é responsável, enquanto um administrador responsável tem uma visão geral de todos os itens, locais e relatos, vinculados à empresa. Por exemplo, um local só pode ser acessado e gerenciado pelo **Administrador** que o criou e/ou pelo administrador geral. Nas figuras [Figura 9](#) e [Figura 10](#) é possível visualizar

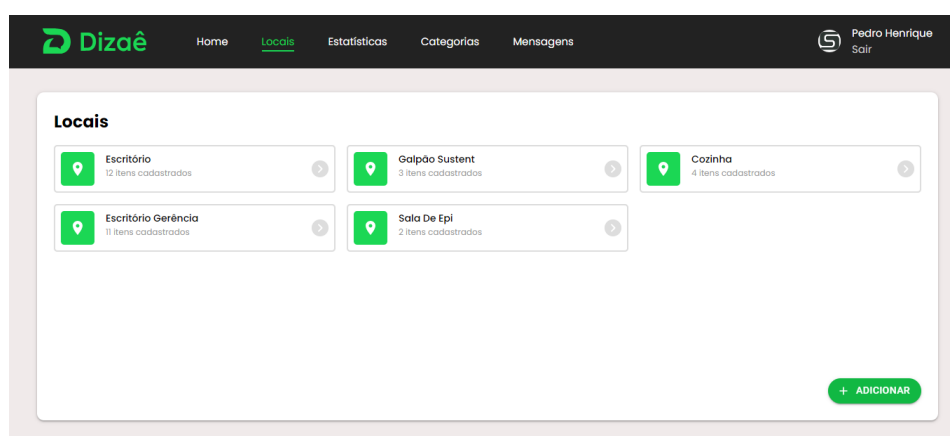
um exemplo desse funcionamento, onde na primeira foram cadastrados três locais e na segunda cinco, não sendo possível visualizar os locais cadastrados pela outra empresa.

Figura 9 – Tela de Local da aplicação *web*



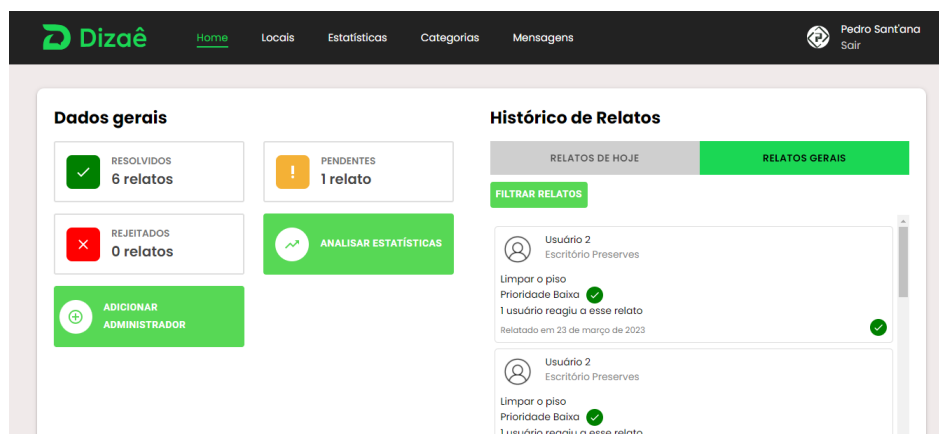
Fonte: Produzido pelo autor

Figura 10 – Tela de Local da aplicação *web*

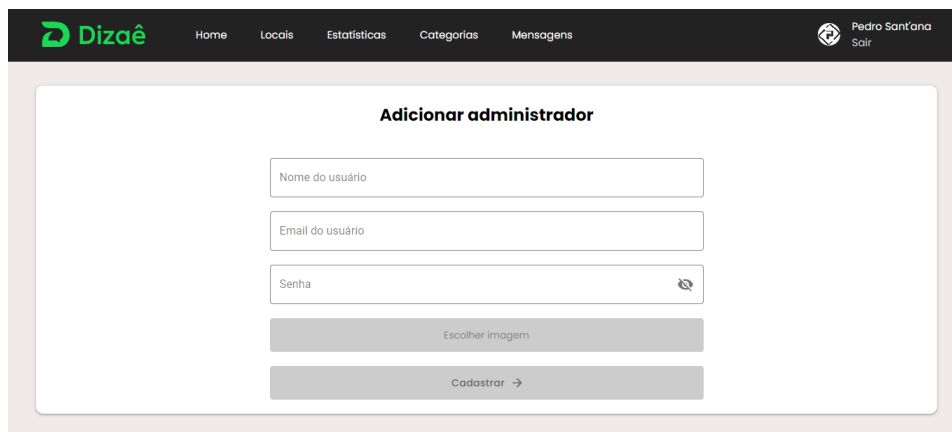


Fonte: Produzido pelo autor

Como foi informado, o cadastro do administrador responsável é feito no momento do cadastro da empresa, já o cadastro de novos administradores da empresa é feito por esse mesmo administrador responsável. Na tela *home*, é disponível para eles a opção de **Adicionar administrador**, representada na Figura 11. Ao clicar nesse botão o usuário é direcionado para a tela de cadastro de administrador, apresentada na Figura 12, onde é necessário preencher o nome, e-mail, senha e foto de perfil. Ao finalizar o cadastro, o novo administrador é automaticamente relacionado a empresa.

Figura 11 – Tela *Home* da aplicação *web*

Fonte: Produzido pelo autor

Figura 12 – Tela de Cadastro de Administrador da aplicação *web*

Fonte: Produzido pelo autor

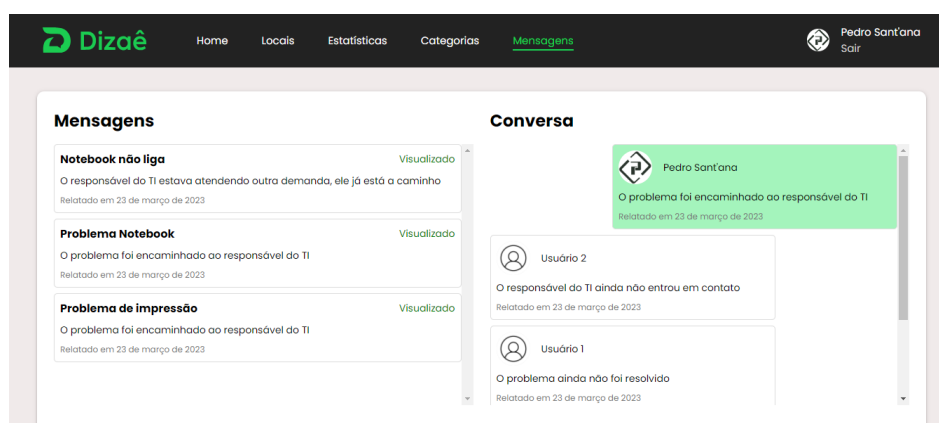
Outras mudanças importantes na plataforma são relacionadas à interação entre os administradores e informantes. Na versão inicial, essa relação era realizada somente através da criação e mudança de *status* dos relatos, que apesar de funcional, apresentava pontos para melhorias. A primeira mudança foi através da criação de um sistema de mensagens, para servir como um canal de comunicação direta entre informantes e administradores. Com essa adição, após a criação de um relato, é criada também uma “sala de bate-papo”, onde é possível enviar e receber mensagens, para tirar dúvidas, explicar melhor a situação, enviar atualizações sobre a situação da resolução do problema, entre outros motivos.

É possível visualizar as telas de mensagens na aplicação *web* na [Figura 13](#) e na [Figura 14](#), onde na esquerda estão as conversas de diferentes relatos, mostrando o título do relato, a última mensagem enviada, a data e se já foram visualizadas as mensagens. Ao clicar em cima de uma conversa, são abertas as mensagens na direita, contando com nome de quem enviou a mensagem, a mensagem enviada e a data; nessa parte as mensagens enviadas pelo usuário atualmente logado ficam em verde, e na parte de baixo é

disponibilizado o local para escrever novas mensagens e o botão de enviar.

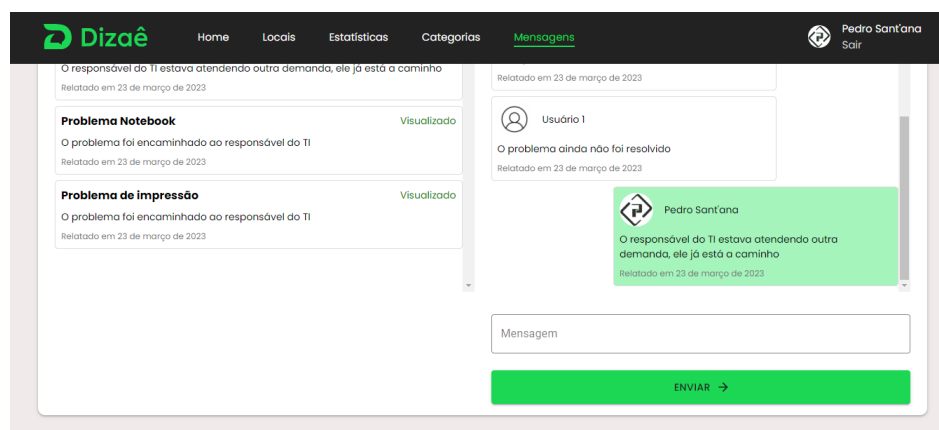
Já a parte *mobile* é apresentada na Figura 15, que mostra todas as conversas que o usuário está relacionado, com título do relato, data, última mensagem e o usuário que enviou a última mensagem. Ao selecionar uma conversa, é aberta a tela de mensagens do relato, apresentada na Figura 16, mostrando todas as mensagens enviadas com relação ao mesmo relato, contando com a mensagem, quem enviou a mensagem e a data de envio, além do campo e botão para envio de novas mensagens.

Figura 13 – Tela de Mensagens da aplicação *web*

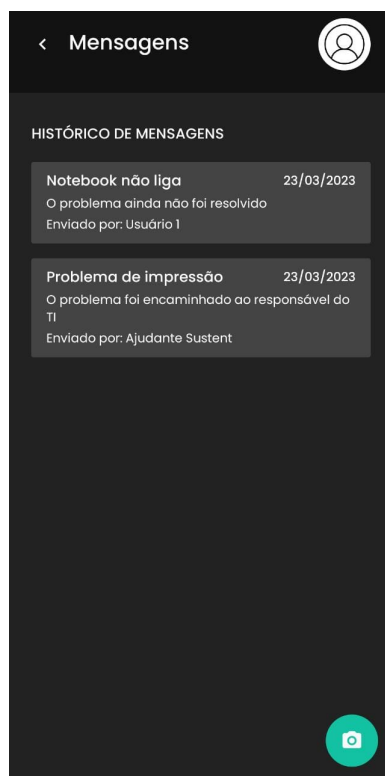


Fonte: Produzido pelo autor

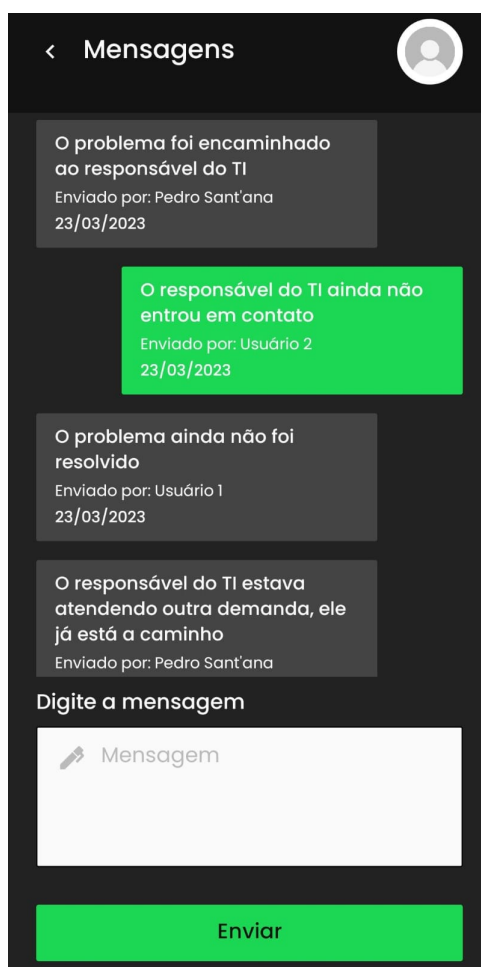
Figura 14 – Tela de Mensagens da aplicação *web*



Fonte: Produzido pelo autor

Figura 15 – Tela de Mensagens da aplicação *mobile*

Fonte: Produzido pelo autor

Figura 16 – Tela de Mensagens do Relato da aplicação *mobile*

Fonte: Produzido pelo autor

Outro ponto para a interação entre informantes e administradores, é através da indicação de prioridade do relato. Ao realizar o cadastro de um relato, representado na [Figura 17](#), existe um novo campo onde o informante pode selecionar a prioridade que ele atribuiu para a resolução do relato. Essa indicação de prioridade fica indicada tanto na tela de relato do informante, [Figura 18](#), quanto na tela de relato do administrador, [Figura 19](#) e [Figura 20](#). Além disso, o administrador pode alterar uma vez a prioridade do relato, caso considere que a indicação inicial precisa mudar, ao clicar no botão **Alterar prioridade**. Ao fazer isso é enviada automaticamente uma mensagem na conversa daquele relato, para que seja registrada a mudança realizada e as pessoas interessadas sejam informadas.

Figura 17 – Tela de Cadastro de Relato da aplicação *mobile*

A interface de usuário da aplicação mobile para o cadastro de relatos. O cabeçalho mostra um ícone de seta para trás e o texto "Escritório Preserves" ao lado de um ícone de perfil. Abaixo, há duas opções de categoria: "Mesas" com uma imagem de uma mesa de madeira e "Cadeiras" com uma imagem de uma cadeira de escritório. Seguem campos de texto para "Digite o título" e "Digite a descrição", ambos com ícones de lápis. Abaixo, há uma seção de "Prioridade" com três opções: "Baixa" (selecionada com um círculo verde), "Média" e "Alta". No rodapé, há um botão "Enviar foto".

< Escritório Preserves

Mesas Cadeiras

Digite o título

Título

Digite a descrição

Descrição

Prioridade

☒ Baixa

☐ Média

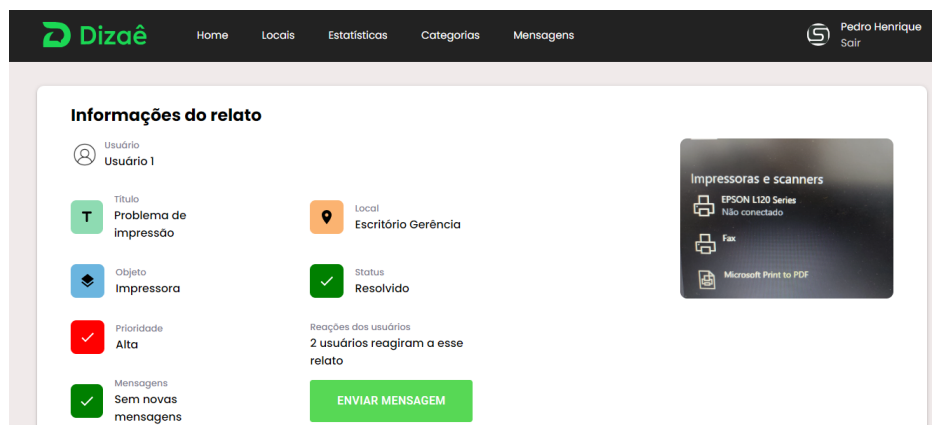
☐ Alta

Enviar foto

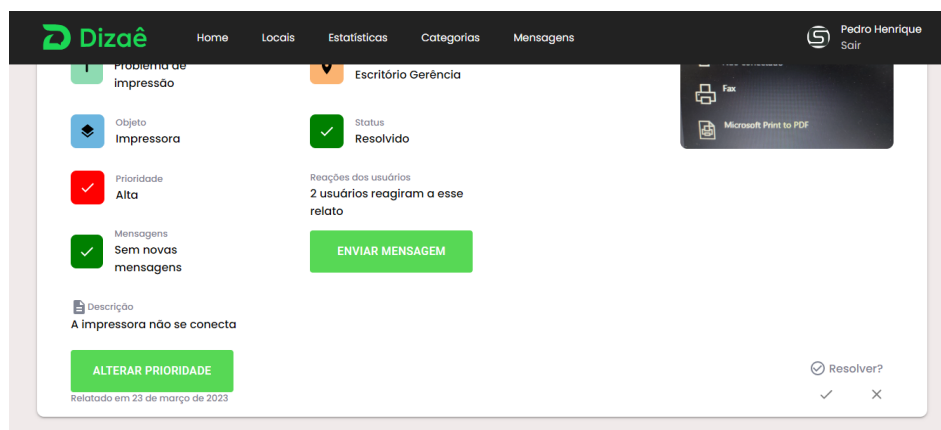
Fonte: Produzido pelo autor

Figura 18 – Tela de Detalhes da aplicação *mobile*

Fonte: Produzido pelo autor

Figura 19 – Tela de Relato da aplicação *web*

Fonte: Produzido pelo autor

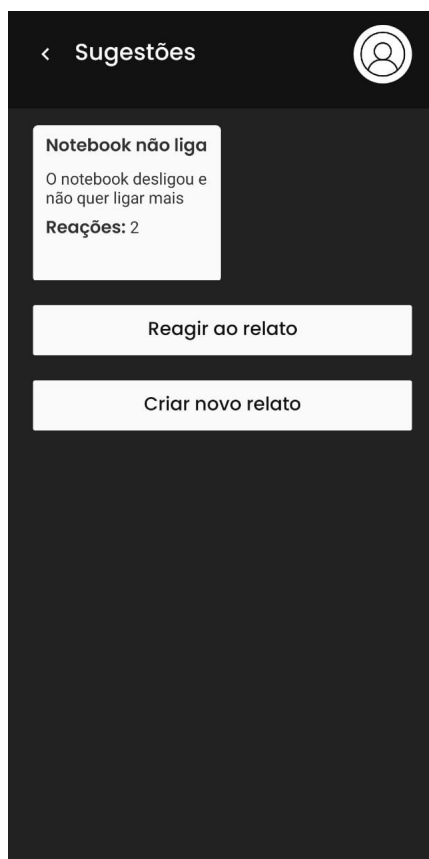
Figura 20 – Tela de Relato da aplicação *web*

Fonte: Produzido pelo autor

Nas telas de visualização de relato, tanto *web* quanto *mobile*, também é possível visualizar a presença do botão **Enviar mensagem**, que direciona o usuário para as respectivas telas de mensagens. Além disso, é possível perceber a existência de um campo indicando **reações de usuários**, campo este que é relacionado a outra adição ao sistema que será explicada a seguir.

Na versão original, foi identificado um possível problema: a existência de relatos semelhantes. Quando mais de um informante identificava o mesmo problema, a única solução era todos eles criarem um relato, isso poderia causar confusão ao administrador e também atrapalhar nas análises estatísticas dos dados, ao acreditar que diversos problemas estavam acontecendo em um local, quando na realidade era um só.

Com o objetivo de reduzir esse problema, foi adicionado o sistema de **Sugestões**. Nesse sentido, quando um informante realiza a leitura de um *QR Code*, o sistema verifica se existem relatos não finalizados naquele local e, caso existam, apresenta esses relatos como sugestões para que o usuário possa selecionar. Ao confirmar a seleção de uma sugestão, o sistema verifica se aquele usuário já reagiu ao problema (através dos identificadores do usuário e do relato), caso tenha, o sistema avisa que o usuário já reagiu àquela sugestão, não permitindo a operação. No caso de ainda não ter interagido, o sistema armazena os dados do usuário e do relato, e atualizando as informações do mesmo, com o campo **reações de usuários** sendo incrementado para demonstrar quantas pessoas identificaram aquele mesmo problema. E, se o problema verificado for diferente dos sugeridos, é possível criar um novo relato normalmente. Essas funcionalidade de sugestões e do botão de criar novo relato, podem ser verificadas na [Figura 21](#).

Figura 21 – Tela de Sugestões da aplicação *mobile*

Fonte: Produzido pelo autor

Outra questão que sofreu mudanças na nova versão foi com relação à geração de gráficos para análises estatísticas. Na versão inicial, as informações eram agrupadas apenas por seu *status* (Pendente, Resolvido, Rejeitado) por mês no ano atual. Com o objetivo de melhorar as análises pelos administradores, foram adicionadas novas opções para geração de gráficos, sendo elas filtrar somente por um tipo de *status*, agrupar as informações por situação (*status*), prioridade, local e categoria, selecionar o ano e se deseja agrupar os dados ao longo do ano todo ou um mês específico, esses filtros podem ser visualizados na [Figura 22](#). Com isso, é possível ter um melhor entendimento da situação da organização, ao analisar por exemplo, que os móveis possuem uma taxa maior de problemas do que os eletrônicos, ou que um andar de um hotel apresenta um aumento no número de relatos sempre no mês de março. Opções de filtros também foram adicionados a tela *Home* dos administradores, para que possam filtrar os relatos de forma mais eficiente, com opções como selecionar *status* específicos, filtrar por data (mais ou menos recentes) ou prioridade (mais ou menos urgentes), representados na [Figura 23](#).

Figura 22 – Opções de filtros de relatos

Tipo de relatos

☒ Todos

☐ Resolvidos

☐ Rejeitados

☐ Pendentes

Dividir por

☒ Situação

☐ Prioridade

☐ Local

☐ Categoria

Período

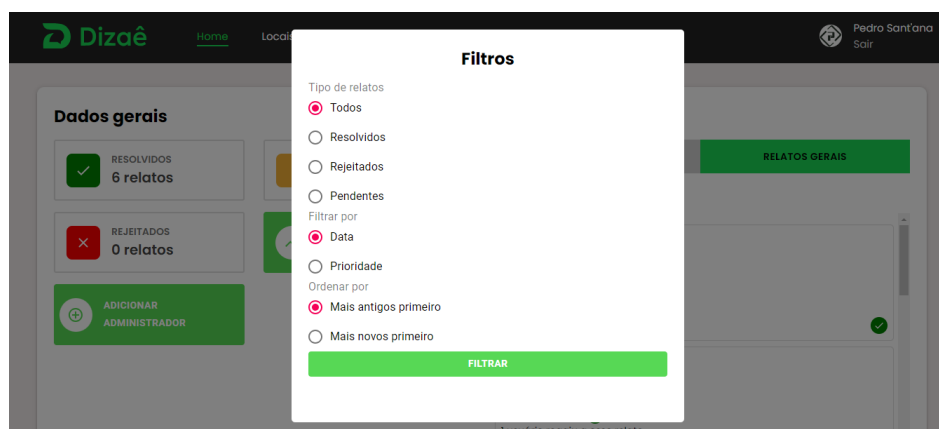
Ano

2023

Mês

Ano inteiro

Fonte: Produzido pelo autor

Figura 23 – Opções de filtros da Tela Home da aplicação *web*

Fonte: Produzido pelo autor

4 Resultados

Para testar o sistema em um ambiente real, ele foi implantado em uma empresa de pequeno porte na cidade de Barão de Cocais, no interior de Minas Gerais no mês de março de 2023. Este capítulo apresenta os resultados obtidos.

4.1 Contexto

A empresa onde a solução foi implantada realiza serviços de aluguel de máquinas pesadas, assim como serviços de engenharia, reparações, entre outros, como terceirizada para outras empresas. No município de Barão de Cocais, ela possui três galpões: um para manutenção das máquinas e outros dois utilizados pelos colaboradores da empresa, onde estão também situados os escritórios do administrativo e da gerência. Para esse trabalho, o galpão da manutenção não foi considerado.

A manutenção desses espaços é de responsabilidade de duas funcionárias, sendo os serviços mais específicos, como manutenção de computadores e elétrica, sendo de responsabilidade de outros profissionais, do TI e dos eletricitas, respectivamente. A comunicação com essa equipe, é feita informalmente, por meio de conversas diretas com as responsáveis, não apresentando um controle preciso dos problemas apresentados.

Para a implantação do sistema, cada galpão foi considerado como uma empresa diferente, já que cada um é utilizado por funcionários de contratos diferentes. As empresas foram cadastradas com usuários para acesso da gerência e do TI e, as duas funcionárias da manutenção, como administradoras, uma para cada empresa. Nesse contexto, os eletricitas não foram considerados, pois não atendem exclusivamente serviços internos, e no caso do TI, eles foram colocados como os responsáveis da empresa.

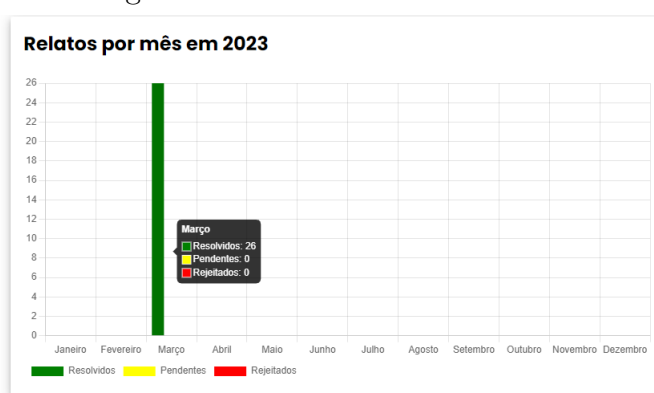
Na primeira “empresa”, identificada como *Sustent*, foram cadastrados 5 locais, possuindo um total de 32 itens. Já na segunda “empresa”, identificada como *Preserves*, foram 11 itens distribuídos em 3 locais. O aplicativo *Dizaê* foi instalado em 6 celulares *Android* de colaboradores da empresa. O sistema ficou em funcionamento durante o todo o mês de março de 2023.

4.2 Utilização

Ao longo do mês, na empresa *Sustent* foram relatados 26 problemas, sendo todos resolvidos, como representado na [Figura 24](#) e na [Figura 25](#). Na primeira figura, os itens são visualizados de forma anual, enquanto na segunda é aplicado o filtro para visualização

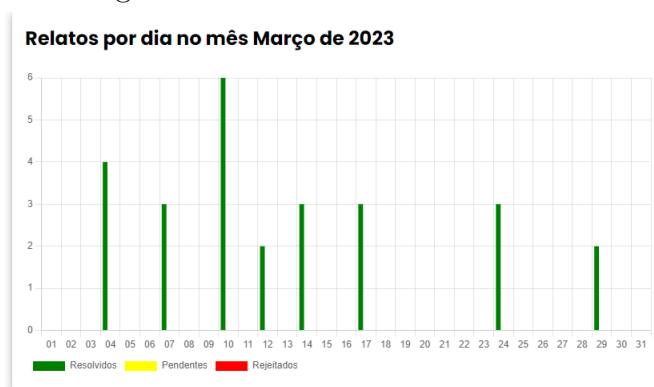
somente ao longo do mês. Ao separar as informações por categorias, conforme a [Figura 26](#) e a [Figura 27](#), é possível perceber que a maior ocorrência de problemas foi relacionada à limpeza e, ao analisar a divisão ao longo do mês, foi possível associar esta alta nos casos com um período de chuvas na região. Com a aplicação do filtro de prioridade ([Figura 28](#)) foi observado que a maioria dos casos encontrados eram problemas com baixa prioridade, mas que ainda sim existiram problemas mais urgentes. O local que apresentou maior quantidade de relatos foi o escritório, representado na [Figura 29](#), por se tratar de um lugar com maior presença e fluxo de pessoas, enquanto a sala de Equipamentos de Proteção Individual (EPIs), por ser um local mais restrito, não teve nenhum relato.

Figura 24 – Gráfico anual de relatos



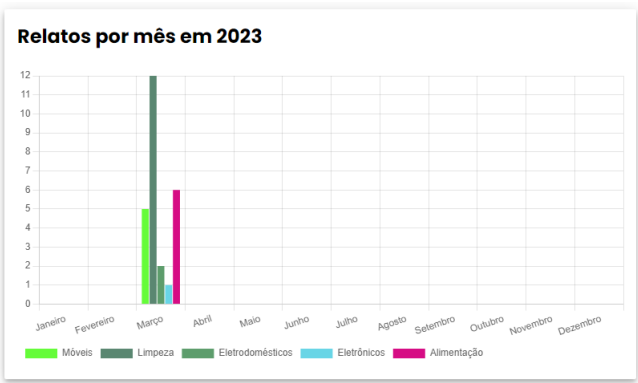
Fonte: Produzido pelo autor

Figura 25 – Gráfico mensal de relatos



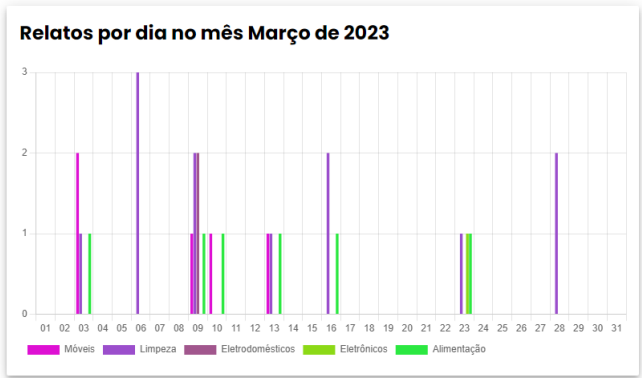
Fonte: Produzido pelo autor

Figura 26 – Gráfico anual de relatos por categorias



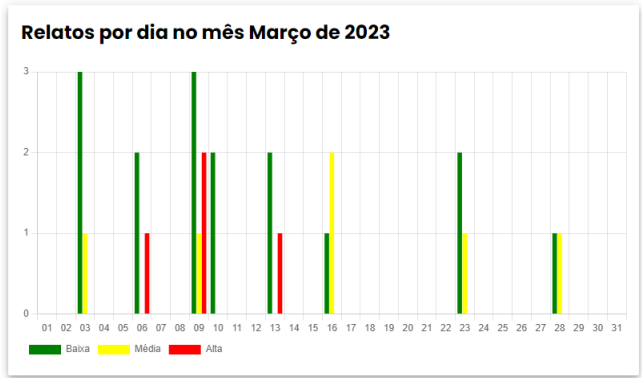
Fonte: Produzido pelo autor

Figura 27 – Gráfico mensal de relatos por categorias



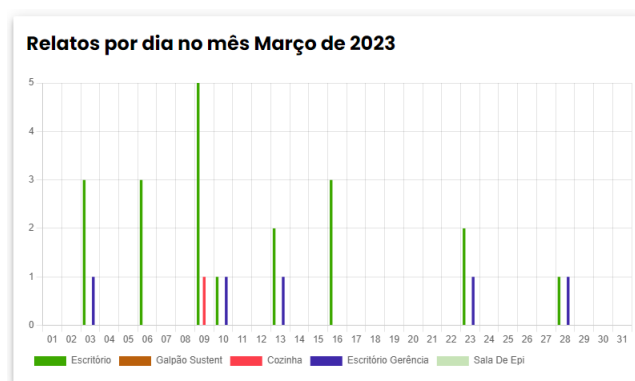
Fonte: Produzido pelo autor

Figura 28 – Gráfico mensal de relatos por prioridade



Fonte: Produzido pelo autor

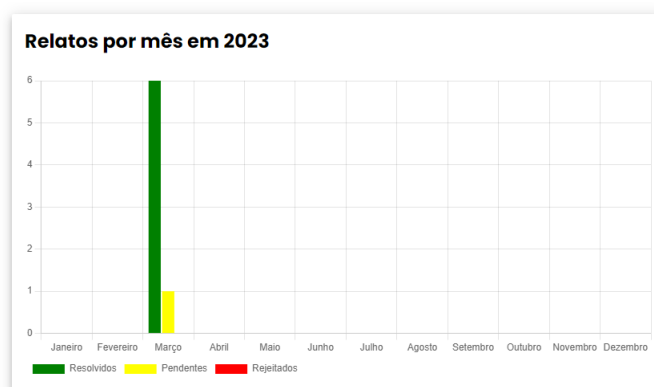
Figura 29 – Gráfico mensal de relatos por local



Fonte: Produzido pelo autor

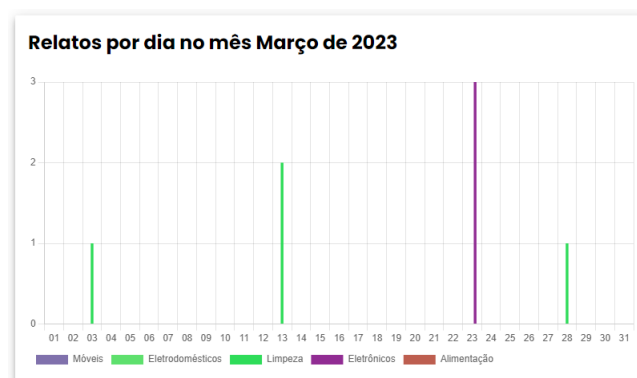
Já na empresa *Preserves*, foram identificados 7 problemas, representados na [Figura 30](#). Com a separação dos relatos por mês e por categorias ([Figura 31](#)) foi possível perceber que os problemas verificados foram relacionados com a limpeza e com eletrônicos, tendo os problemas dessa categoria acontecido no mesmo dia, um dos quais levou a falhas de conexão das impressoras, causadas por mau funcionamento da rede/internet da empresa.

Figura 30 – Gráfico anual de relatos



Fonte: Produzido pelo autor

Figura 31 – Gráfico mensal de relatos por categorias



Fonte: Produzido pelo autor

4.3 Avaliação

Após o período de testes, as pessoas que tiveram contato com a plataforma foram questionadas a respeito da sua experiência com a ferramenta. As responsáveis pela manutenção, assim como os colaboradores selecionados, avaliaram a aplicação de maneira positiva. Para as administradoras, o sistema permitiu um maior controle e organização das suas funções, conseguindo acompanhar de forma mais efetiva quais atividades foram completas e quais ainda estavam pendentes. Além disso, permitiu que pudessem direcionar mais facilmente problemas de responsabilidade de outras áreas, como manutenção de computadores, que puderam identificar, contatar o responsável do TI e avisar ao usuário que o relato já havia sido repassado ao setor correto.

Para os informantes, a facilidade de reportar o problema foi o principal fator de satisfação, pois sem o sistema, é necessário procurar pela empresa onde estão as responsáveis, que podem estar atendendo outro setor, perdendo tempo das suas funções no serviço. Além disso, a utilização das mensagens para acompanhamento do processo da manutenção foi muito interessante, pois permitiu que eles soubessem exatamente como estava a situação.

Outro grupo que avaliou positivamente a plataforma, foi o dos gerentes, pois o sistema permitiu que eles tivessem um melhor entendimento das demandas de manutenção na organização. Em especial, a tela de estatísticas foi a mais bem recebida, pois com os filtros foi possível ver qual categoria de itens mais dava problemas a manutenção, levando a mudanças no dia a dia relacionadas a esses itens. Além disso, foram sugeridas melhorias para futuras versões nessa funcionalidade, como a disponibilidade de mais filtros, novos tipos de gráficos e geração de *PDFs*, para melhorar ainda mais as possibilidades de avaliações.

5 Conclusão

Esse trabalho teve como objetivo apresentar a ferramenta *Dizaê*, uma plataforma para auxiliar no gerenciamento de ativos, e as melhorias adicionadas a ela. A solução desenvolvida se mostrou atrativa para pequenas e médias empresas, devido ao seu funcionamento e implantação mais simplificados em comparação com outras soluções disponíveis no mercado.

As mudanças implementadas na versão discutida ao longo desse trabalho trouxeram uma melhoria na interação entre os usuários e administradores ao criar um canal de comunicação direto entre os envolvidos, além de trazer a sinalização de prioridade, que permite aos utilizadores demonstrarem corretamente a necessidade de resolução para o problema, facilitando que os responsáveis pelos espaços físicos possam se organizar melhor, o que poderá gerar uma redução do tempo de resolução.

Outro ponto de destaque, foram as adições na parte de geração de gráficos, trazendo a possibilidade de análises mais específicas dos dados gerados pelo sistema, que podem ser utilizados para planejamento de manutenções preventivas, assim como a possível substituição de aparelhos/produtos que apresentam problemas frequentes, trazendo a possibilidade de gerar economias para as organizações.

5.1 Trabalhos Futuros

Apesar do sucesso de utilização durante o período de testes, existem aspectos que podem ser melhorados. Entre eles, a adição de novas opções de filtros, gráficos e geração de relatórios, para melhorar ainda mais a capacidade de análises estatísticas pela plataforma, foram pontos relatados pelos administradores que utilizaram o sistema. Além disso, a adição de sugestões aos usuários antes de relatarem os problemas, causou uma grande redução do número de relatos semelhantes, porém não impede totalmente a redundância de relatos, pois um usuário ainda pode relatar um mesmo problema. Então a utilização de técnicas de Inteligência Artificial (IA), como processamento de linguagem natural, poderiam automatizar as verificações de relatos semelhantes.

Outras questões que também podem apresentar melhorias com automação, seriam o envio de e-mails e notificações ao celular em momentos como a criação de novos relatos, mudança de prioridade e *status* do relato e recebimento de mensagens. Outro ponto seria adicionar a possibilidade de administradores receberem periodicamente relatórios automáticos, baseados em filtros previamente selecionados por eles.

Além disso, algumas adições e modificações foram sugeridas para melhorar a

qualidade geral do sistema, como: permitir o envio de vídeos e adicionar múltiplas fotos (tiradas na hora ou da galeria) ao cadastrar um relato; adição de login social, para que os usuários possam se cadastrar e entrar na aplicação de forma mais prática, através de informações existentes em outros serviços, como *Google*, *Facebook*, *LinkedIn*, entre outros; mudanças de *layouts* em algumas telas, padronizando componentes e melhorando a experiência do usuário com a plataforma.

Referências

- BUI, D. Reactive programming and clean architecture in android development. Metropolis Ammattikorkeakoulu, 2017. Citado 2 vezes nas páginas 21 e 22.
- CAVALCANTE, C. A. V.; ALMEIDA, A. T. d. Modelo multicritério de apoio a decisão para o planejamento de manutenção preventiva utilizando promethee ii em situações de incerteza. *Pesquisa Operacional*, SciELO Brasil, v. 25, p. 279–296, 2005. Citado na página 13.
- COMPARE, M.; BARALDI, P.; ZIO, E. Challenges to iot-enabled predictive maintenance for industry 4.0. *IEEE Internet of Things Journal*, IEEE, v. 7, n. 5, p. 4585–4597, 2019. Citado na página 13.
- CORONEL, C.; MORRIS, S. A. *Database systems: design, implementation and management*. [S.l.]: Cengage learning, 2022. Citado na página 28.
- COULOURIS, G. et al. *Sistemas Distribuídos-: Conceitos e Projeto*. [S.l.]: Bookman Editora, 2013. Citado na página 19.
- DE, B.; DE, B. *API management*. [S.l.]: Springer, 2017. Citado na página 20.
- ELMASRI, R. *Sistemas de banco de dados: fundamentos e aplicações*. 6. ed. [S.l.]: LTC, 2002. Citado na página 28.
- HOQUE, N.; BHATTACHARYYA, D. K.; KALITA, J. K. Botnet in ddos attacks: trends and challenges. *IEEE Communications Surveys & Tutorials*, IEEE, v. 17, n. 4, p. 2242–2270, 2015. Citado na página 21.
- LEWANDOWSKI, S. M. Frameworks for component-based client/server computing. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 30, n. 1, p. 3–27, 1998. Citado na página 20.
- MAN, L. C. K.; NA, C. M.; KIT, N. C. Iot-based asset management system for healthcare-related industries. *International Journal of Engineering Business Management*, SAGE Publications Sage UK: London, England, v. 7, p. 19, 2015. Citado na página 13.
- MARCORIN, W. R.; LIMA, C. R. C. Análise dos custos de manutenção e de não-manutenção de equipamentos produtivos. *Revista de ciência & tecnologia*, v. 11, n. 22, p. 35–42, 2003. Citado 2 vezes nas páginas 13 e 17.
- MARTIN, R. C. *Clean architecture*. [S.l.]: Prentice Hall, 2017. Citado 2 vezes nas páginas 21 e 22.
- MCMAHON, P.; ZHANG, T.; DWIGHT, R. Requirements for big data adoption for railway asset management. *IEEE Access*, IEEE, v. 8, p. 15543–15564, 2020. Citado na página 14.
- META. *React*. 2023. <<https://react.dev/>>. Acessado em: 20 jul. 2023. Citado na página 29.

- META. *React Native*. 2023. <<https://reactnative.dev/>>. Acessado em: 20 jul. 2023. Citado na página 29.
- MICROSOFT. *TypeScript*. 2012–2023. <<https://www.typescriptlang.org/>>. Acessado em: 20 jul. 2023. Citado na página 28.
- MOBLEY, R. K. *An introduction to predictive maintenance*. [S.l.]: Elsevier, 2002. Citado na página 14.
- NETO, R. C. D. d. A. *Gestão do Conhecimento em Organizações: Proposta de Mapeamento Coneitual Integrativo*. Disponível em: *Minha Biblioteca*, Editora Saraiva. [S.l.: s.n.], 2007. Citado na página 14.
- OLIVEIRA, C. L. V.; ZANETTI, H. A. P. *Node.js: programe de forma rápida e prática*. Disponível em: *Minha Biblioteca*, Editora Saraiva. [S.l.: s.n.], 2021. Citado na página 27.
- OLUWATOSIN, H. S. Client-server model. *IOSR Journal of Computer Engineering*, IOSR Journals, v. 16, n. 1, p. 67–71, 2014. Citado 2 vezes nas páginas 20 e 21.
- ORACLE. *Oracle Supply Chain Management (SCM)*. 2023. <<https://www.oracle.com/br/scm/maintenance/>>. Acessado em: 20 jul. 2023. Citado na página 18.
- PANEGOSSO, A. C. G.; SILVA, E. A evolução da gestão de ativos. In: _____. [S.l.: s.n.], 2021. p. 23–51. Citado na página 17.
- PROJECT, A. *Axios*. 2014. <<https://axios-http.com/ptbr/>>. Acessado em: 20 jul. 2023. Citado na página 29.
- RAN, Y. et al. A survey of predictive maintenance: Systems, purposes and approaches. *arXiv preprint arXiv:1912.07383*, 2019. Citado na página 13.
- RICHARDS, M.; FORD, N. *Fundamentals of software architecture: an engineering approach*. [S.l.]: O'Reilly Media, 2020. Citado 2 vezes nas páginas 18 e 19.
- ROSSETTI, A.; MORALES, A. B. O papel da tecnologia da informação na gestão do conhecimento. *Ciência da Informação*, SciELO Brasil, v. 36, p. 124–135, 2007. Citado na página 14.
- SANTOS, B. et al. Indústria 4.0: Desafios e oportunidades. *Revista Produção e Desenvolvimento*, v. 4, 01 2018. Citado na página 17.
- SCHNEIDER, J. et al. Asset management techniques. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 28, n. 9, p. 643–654, 2006. Citado 2 vezes nas páginas 14 e 17.
- SELCUK, S. Predictive maintenance, its implementation and latest trends. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, SAGE Publications Sage UK: London, England, v. 231, n. 9, p. 1670–1679, 2017. Citado na página 13.
- SERRADILLA, O. et al. Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects. *Applied Intelligence*, Springer, v. 52, n. 10, p. 10934–10964, 2022. Citado na página 14.

- SILVA, F. *Cloud Computing. Disponível em: Minha Biblioteca*. [S.l.: s.n.], 2020. Citado na página 28.
- SOMMERVILLE, I. *Engenharia de Software*. [S.l.: s.n.], 2019. Citado 2 vezes nas páginas 19 e 20.
- STEFANO, N. M. et al. Gestão de ativos intangíveis: implicações e relações da gestão do conhecimento e capital intelectual. *Perspectivas em Gestão & Conhecimento*, Centro de Ciências Sociais Aplicadas, v. 4, n. 1, p. 22–37, 2014. Citado na página 17.
- TANENBAUM, A. S.; STEEN, M. van. *Sistemas distribuídos: princípios e paradgmas*. [S.l.: s.n.], 2007. Citado na página 19.
- TOMAZ, F. A. M. *Dizaê - Gerenciamento inteligente de espaços físicos*. Dissertação (Graduação) — Universidade Federal de Ouro Preto, João Monlevade, 2021. Citado 3 vezes nas páginas 15, 24 e 29.
- VALENTE, M. T. D. O. et al. Engenharia de software moderna: princípios e práticas para desenvolvimento de software com produtividade. Universidade Federal de Minas Gerais, 2020. Citado na página 19.
- VALIPOUR, M. H. et al. A brief survey of software architecture concepts and service oriented architecture. In: IEEE. *2009 2nd IEEE International Conference on Computer Science and Information Technology*. [S.l.], 2009. p. 34–38. Citado na página 19.
- VELTE, A. T. V. T. J.; ELSENPETER, P. D. R. *Cloud computing*. [S.l.: s.n.], 2010. Citado na página 20.
- VERAS, M. *Cloud Computing: nova arquitetura da TI*. [S.l.]: Brasport, 2012. Citado na página 28.
- VERCEL, I. *Next.js*. 2023. <<https://nextjs.org/>>. Acessado em: 20 jul. 2023. Citado na página 29.
- WAN, J. et al. A manufacturing big data solution for active preventive maintenance. *IEEE Transactions on Industrial Informatics*, IEEE, v. 13, n. 4, p. 2039–2047, 2017. Citado na página 14.