



UNIVERSIDADE FEDERAL DO CEARÁ
INSTITUTO UNIVERSIDADE VIRTUAL
CURSO DE GRADUAÇÃO EM SISTEMAS E MÍDIAS DIGITAIS

ARTUR DE FREITAS ALMEIDA

**VIZAGE: O PROCESSO DE DESENVOLVIMENTO DE UMA PLATAFORMA PARA
CENTRALIZAR A HOSPEDAGEM DE PAINÉIS DE BI**

FORTALEZA

2023

ARTUR DE FREITAS ALMEIDA

VIZAGE: O PROCESSO DE DESENVOLVIMENTO DE UMA PLATAFORMA PARA
CENTRALIZAR A HOSPEDAGEM DE PAINÉIS DE BI

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas e Mídias Digitais do Instituto UFC Virtual da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas e Mídias Digitais.

Orientador: Prof. Dr. Alysso Diniz dos Santos.

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

A444v Almeida, Artur de Freitas.

Vizage : o processo de desenvolvimento de uma plataforma para centralizar a hospedagem de painéis de BI / Artur de Freitas Almeida. – 2023.

75 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Instituto UFC Virtual, Curso de Sistemas e Mídias Digitais, Fortaleza, 2023.

Orientação: Prof. Dr. Alysson Diniz dos Santos.

1. Análise de dados. 2. Centralização da análise. 3. Plataforma web. I. Título.

CDD 302.23

ARTUR DE FREITAS ALMEIDA

VIZAGE: O PROCESSO DE DESENVOLVIMENTO DE UMA PLATAFORMA PARA
CENTRALIZAR A HOSPEDAGEM DE PAINÉIS DE BI

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas e Mídias Digitais do Instituto UFC Virtual da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas e Mídias Digitais.

Aprovada em: xx/xx/xxxx.

BANCA EXAMINADORA

Prof. Dr. Alysson Diniz dos Santos (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Ricardo Brauner dos Santos
Universidade Federal do Ceará (UFC)

Prof. Dr. Natal Anacleto Chicca Junior
Universidade Federal de Pernambuco (UFPE)

Dedico esse trabalho aos meus pais, por acreditarem em mim e estarem sempre do meu lado. Obrigado por serem os melhores pais do mundo.

AGRADECIMENTOS

Aos meus amados pais, que são minha inspiração constante e apoio incondicional, o meu mais profundo agradecimento por todo amor, incentivo e dedicação ao longo dessa jornada. Sem vocês, este TCC não seria possível. Sou eternamente grato por ter pais tão maravilhosos como vocês.

Ao Estevam, por ter me ensinado tanto e acreditado em mim até mesmo quando eu não acreditava, obrigado por ser meu porto seguro.

Agradeço ao meu amigo Roberto, por sempre me dar bons conselhos e por aguentar passar horas conversando comigo sobre a vida.

À minha amiga Sara, por ter me dado todo o apoio que eu precisava e me ter me alegrado tantas vezes com sua risada contagiante.

À toda a equipe da PATH, em especial à Veronica Simões e ao Victor Mendes, por serem amigos e mentores incríveis que me introduziram ao mundo dos dados.

Ao Prof. Alysson, pela excelente orientação e pelo auxílio no desenvolvimento deste trabalho, obrigado por não ter desistido de mim.

Aos amigos e colegas no curso Sistemas e Mídias Digitais que sempre me apoiaram ao longo do curso.

“Uma alma saudável habita em uma mente
saudável, e em um corpo saudável”

(Maka Albarn)

RESUMO

Com a crescente geração de dados nas organizações, a análise de informações tem se tornado cada vez mais importante para a tomada de decisões estratégicas. Nesse sentido, as ferramentas de *Business Intelligence* (BI) têm sido desenvolvidas para fornecer visualizações e insights valiosos, permitindo que as organizações tomem decisões mais informadas e eficazes. Porém, por diversos motivos, diferentes áreas de uma organização acabam utilizando diferentes ferramentas de BI para analisar seus dados. Esta abordagem causa problemas, como a dispersão de informações em diferentes locais, o que dificulta a visualização e entendimento dos dados e a tomada de decisões com base neles. Além disso, a gestão e a manutenção de múltiplas plataformas podem se tornar tarefas complexas e onerosas. Pensando nesse problema, este trabalho documenta o desenvolvimento de uma solução web para centralizar a hospedagem de dashboards de análises de dados, mesmo que tenham sido desenvolvidos em plataformas de BI diferentes. Os resultados alcançados com o teste de usabilidade sobre o protótipo, e o questionário SUS (*System Usability Scale*) revelaram uma aderência positiva pelos usuários com a solução proposta, também foram atendidos todos os requisitos estabelecidos durante o processo de desenvolvimento.

Palavras-chave: Análise de dados; Centralização da análise; Plataforma web; Vue; Firebase.

ABSTRACT

With the growing generation of data in organizations, the analysis of information has become increasingly important for strategic decision making. In this sense, Business Intelligence (BI) tools have been developed to provide valuable visualizations and insights, allowing organizations to make more informed and effective decisions. However, for various reasons, different areas of an organization end up using different BI tools to analyze their data. This approach causes problems, such as the dispersion of information in different locations, which makes it difficult to visualize and understand the data and make decisions based on them. Furthermore, managing and maintaining multiple platforms can become complex and costly tasks. With this problem in mind, this work documents the development of a web solution to centralize the hosting of data analysis dashboards, even if they have been developed on different BI platforms. The results achieved with the usability test on the prototype, and the SUS questionnaire (System Usability Scale) revealed a positive adherence by the users with the proposed solution, and all the requirements established during the development process were also met.

Keywords: Data analysis; Centralization of analysis; Web platform; Vue; Firebase.

LISTA DE FIGURAS

Figura 1	– As etapas do design thinking	19
Figura 2	– As etapas do ciclo de vida dos dados	24
Figura 3	– As etapas do ciclo de vida dos dados, incorporando a plataforma VIZAGE	26
Figura 4	– Etapas da metodologia	30
Figura 5	– Modelo de dados de um dashboard	35
Figura 6	– Objeto dashboard sendo passado em formato JSON para o banco de dados	36
Figura 7	– Dashboard gravado no banco de dados Firebase	36
Figura 8	– Código de conexão com o Firebase	37
Figura 9	– Código das operações C.R.U.D no arquivo “firebase.js	38
Figura 10	– Protótipo da página Home	40
Figura 11	– Protótipo da página Painéis	40
Figura 12	– Protótipo do pop-up da página Painéis	41
Figura 13	– Protótipo da página Explorar	41
Figura 14	– Protótipo da página “Página não encontrada”	42
Figura 15	– Diretório raiz do projeto	42
Figura 16	– Diretório pasta “dist”	43
Figura 17	– Diretório pasta “node_modules”	43
Figura 18	– Diretório pasta “public”	43
Figura 19	– Diretório pasta “src”	44
Figura 20	– Diretório expandido da pasta “src”	44
Figura 21	– Código das rotas no arquivo “router.js”	45
Figura 22	– Código no arquivo “App.vue”	46
Figura 23	– Código no arquivo “main.js”	47
Figura 24	– Diretório dos arquivos de configurações	48
Figura 25	– Tag template do componente Home.vue	48
Figura 26	– Tag template do componente Navbar.vue	49

Figura 27	– Tag template do componente Header.vue	49
Figura 28	– Diagrama da estrutura do componente Home.vue	50
Figura 29	– Diagrama da estrutura do componente Dashboards.vue	51
Figura 30	– Código para abrir ou fechar pop-ups no componente Dashboards.vue	52
Figura 31	– Funções para abrir ou fechar pop-ups no componente Dashboards.vue	53
Figura 32	– Código para criar botões “Explorar” no componente Dashboards.vue	53
Figura 33	– Código para retornar os dashboards do banco de dados	54
Figura 34	– Diagrama da estrutura dos componentes DashboardTableau.vue e DashboardPowerBI.vue.....	54
Figura 35	– Código das funções onSubmitTableau e onSubmitPowerBI	55
Figura 36	– Painel: Human Development Index in Brazil	56
Figura 37	– Painel: YTD IT Spend Trend Analysis	57
Figura 38	– HTML tag tableau-viz	57
Figura 39	– HTML tag iframe	58
Figura 40	– Resultados do teste de usabilidade	62
Figura 41	– Menu de navegação	63
Figura 42	– Página “Em breve”	63
Figura 43	– Página Início	64
Figura 44	– Página Painéis	64
Figura 45	– Galeria de dashboards na página Painéis	65
Figura 46	– Formulário para importação de dashboard	66
Figura 47	– Novo dashboard importado para a galeria	66
Figura 48	– Visualizando dashboard Power BI recém importado	66
Figura 49	– Destaque no botão de deletar dashboard	67
Figura 50	– Página painéis atualizada sem o dashboard que foi deletado	67

LISTA DE TABELAS

Tabela 1	– Comparação da plataforma Vizage com outras plataformas do mercado	17
Tabela 2	– Exemplo de resposta e pontuação do questionário SUS	28
Tabela 3	– Classificação de satisfação com a pontuação SUS total de uma aplicação	29
Tabela 4	– Requisitos Funcionais do Back-end	33
Tabela 5	– Requisitos Funcionais do Front-end	33
Tabela 6	– Resultado do Questionário SUS da plataforma VIZAGE	62

LISTA DE ABREVIATURAS E SIGLAS

BI	Business Intelligence
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
SUS	System Usability Scale
NPM	Node Package Managment
UFC	Universidade Federal do Ceará
SMD	Sistemas e Mídias Digitais

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Contextualização e Motivação	14
1.2	Objetivos	15
1.3	Ferramentas concorrentes	16
<i>1.3.1</i>	<i>Curator</i>	16
<i>1.3.2</i>	<i>Loom Software</i>	16
<i>1.3.3</i>	<i>The Reporting Hub</i>	17
1.4	Diferencial da plataforma	17
1.5	Estrutura do documento	17
2	REFERENCIAL TEÓRICO	18
2.1	Design Thinking	18
2.2	Requisitos de software	19
2.3	Aplicações Web	20
2.4	Framework Vue.js	20
2.5	Conceitos de Business Intelligence e Análise de dados	21
2.6	Ferramentas de Análise de dados	22
2.7	Dashboards de Análise de dados	22
2.8	Plataformas web de Hospedagem de Dashboards nativas das ferramentas de BI	23
2.9	O problema da integração e centralização de informações nas empresas ...	24
2.10	Questionário SUS	27
3	METODOLOGIA	30
4	DESENVOLVIMENTO	32
4.1	Escolha de tecnologias e ferramentas	32
4.2	Processo de desenvolvimento	32
<i>4.2.1</i>	<i>Levantamento de requisitos</i>	32
<i>4.2.2</i>	<i>Desenvolvimento: Back-end</i>	34
<i>4.2.2.1</i>	<i>Visão geral</i>	34
<i>4.2.2.2</i>	<i>Modelo de dados</i>	34
<i>4.2.2.3</i>	<i>Conexão com o banco de dados Firebase</i>	36
<i>4.2.2.4</i>	<i>Definição das operações C.R.U.D</i>	37

4.2.3	Desenvolvimento: Front-end	39
4.2.3.1	<i>Visão geral</i>	39
4.2.3.2	<i>Protótipo da Interface</i>	39
4.2.3.3	<i>Estrutura de pastas do projeto</i>	42
4.2.3.4	<i>Arquivos de configurações</i>	47
4.2.3.5	<i>Anatomia dos componentes principais</i>	48
4.2.3.6	<i>Importação de dashboards</i>	55
4.2.3.7	<i>Incorporando dashboards na página</i>	55
4.3	Reflexão sobre o desenvolvimento	56
5	RESULTADOS	60
5.1	Métodos	60
5.1.1	<i>Teste assistido de usabilidade</i>	61
5.1.2	<i>Questionário SUS</i>	61
5.2	Apresentação dos Resultados	61
5.3	Atendimento dos requisitos funcionais: Front-end	63
5.4	Atendimento dos requisitos funcionais: Back-end	68
5.5	Discussão	68
6	CONCLUSÕES E O FUTURO DO PROJETO	71
6.1	Considerações finais	71
6.2	O Futuro da plataforma	71
	REFERÊNCIAS	73
	APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)	75

1 INTRODUÇÃO

1.1 Contextualização e Motivação

Com a crescente geração de dados nas organizações, a análise dessas informações tem se tornado cada vez mais importante para a tomada de decisões estratégicas. Nesse sentido, as ferramentas de Business Intelligence (BI) têm sido desenvolvidas para fornecer visualizações e insights valiosos, permitindo que as organizações tomem decisões mais informadas e eficazes (DILIGENT, 2023).

No entanto, muitas vezes, diferentes áreas de uma organização utilizam diferentes ferramentas de BI para analisar seus dados. Quando um dashboard é feito em uma dessas ferramentas, ele só pode ser publicado na plataforma online referente à ferramenta em que foi construído. Então, por exemplo, se criarmos um dashboard na ferramenta Tableau, teríamos que publicá-lo no Tableau Online ou Tableau Server, se criássemos no Power BI, teríamos que publicá-lo no portal online do Power BI.

Ao publicar os dashboards de análise de dados em uma plataforma online, é possível permitir o acesso de várias pessoas ao mesmo tempo. Isso significa que os usuários podem colaborar em projetos, trabalhar em equipe e compartilhar informações de maneira rápida e fácil. Além disso, é possível gerenciar e controlar o acesso aos dashboards, definindo quem pode visualizá-los, editá-los ou compartilhá-los (DA SILVA; COELHO, 2020).

A utilização de diferentes ferramentas de BI para criar dashboards é comum em empresas, especialmente quando diferentes equipes e departamentos têm necessidades distintas de análise de dados. Cada ferramenta de BI possui recursos específicos que atendem a necessidades específicas, como a análise de vendas, finanças ou logística.

Além disso, em muitos cenários, a empresa pode estar passando por uma transição de ferramentas de BI quando precisa atualizar suas soluções tecnológicas ou quando decide adotar uma nova abordagem de análise de dados. No entanto, essa transição pode não ser rápida e pode exigir que as ferramentas de BI coexistam por um longo período.

Essa abordagem de utilizar duas ferramentas de BI pode levar a uma desvantagem: publicar dashboards em diferentes plataformas. O problema é que as informações ficam dispersas em diferentes locais, o que pode dificultar a visualização dos dados e a tomada de decisões com base neles. Além disso, a gestão e a manutenção de múltiplas plataformas podem se tornar tarefas complexas e onerosas (PRIMAK, 2008).

Em resumo, a desvantagem de publicar dashboards em diferentes plataformas é

que pode ser difícil para os usuários gerenciar e acessar as informações em um só lugar. Isso pode levar a problemas de comunicação e colaboração, pois as informações são distribuídas em várias plataformas diferentes, dificultando a análise desses dados. Além disso, a troca de informações entre as diferentes ferramentas pode gerar perda de dados e informações importantes.

Por isso, é importante unificar as informações em um só lugar, criando uma plataforma que possa hospedar dashboards de análise de dados criados em diferentes ferramentas de BI, como Tableau e Power BI. Essa plataforma deve ser capaz de integrar dashboards de diferentes fontes em um só lugar, permitindo que os usuários gerenciem e acessem as informações de forma fácil e eficiente. Dessa forma, é possível garantir que as informações estejam atualizadas e precisas e que os usuários possam colaborar e compartilhar informações de maneira mais eficaz, melhorando a tomada de decisões baseada em dados (DA SILVA; COELHO, 2020).

1.2 Objetivos

Diante desse cenário, este trabalho tem como objetivo geral desenvolver uma plataforma web para unificar / centralizar a hospedagem de dashboards de análises de dados. Esta plataforma se integrará a ferramentas de BI populares, como Tableau e Power BI, e irá incorporar os painéis na página da web. O diferencial da nossa plataforma será a integração de painéis, mesmo que tenham sido criados em ferramentas diferentes, permitindo que os usuários acessem todas as informações relevantes em um único local. Para alcançar esse resultado, temos como objetivos específicos:

- Apresentar o problema da centralização de dashboards de análise de dados.
- Desenvolver a plataforma web como solução proposta.
- Avaliar o sistema por meio dos aspectos de usabilidade.

Essa plataforma será comparada com outras já existentes no mercado, a fim de avaliar seus pontos fortes e fracos, bem como sua capacidade de atender às necessidades dos usuários.

Por meio de uma pesquisa exploratória do mercado, foram decididos alguns critérios para realizar essa comparação. Serão considerados os critérios: preço, escalabilidade, facilidade de uso e o número de ferramentas de BI suportadas. Além disso, serão realizados testes de desempenho e usabilidade, a fim de avaliar a experiência do usuário e a capacidade da plataforma de lidar com grandes volumes de dashboards.

Espera-se que os resultados desta pesquisa possam contribuir para o desenvolvimento de plataformas de análise de dados mais eficientes e integradas, permitindo que as organizações tomem decisões mais informadas e colaborativas. Além disso, a plataforma será gratuita.

1.3 Ferramentas concorrentes

O desenvolvimento de novas plataformas requer uma análise criteriosa das soluções concorrentes já disponíveis no mercado. A análise das plataformas concorrentes é fundamental para entender as necessidades dos usuários, identificar oportunidades de melhoria e garantir a relevância e competitividade da nova solução (CORTEX INTELLIGENCE, 2023).

1.3.1 Curator

Curator é uma plataforma criada pela empresa Interworks para a integração de dashboards de diferentes ferramentas de BI em um só lugar. Ela permite aos usuários criar um portal de dashboards que integra painéis de plataformas diferentes, como Tableau e Power BI. Uma das vantagens da plataforma é a facilidade de uso e configuração, permitindo que os usuários configurem a plataforma e criem portais de dashboards personalizados. No entanto, o Curator pode ser uma opção cara dependendo do tamanho e complexidade da implementação (CURATOR, 2023).

1.3.2 Loome Software

Loome Software é uma plataforma de análise de dados baseada na nuvem que permite às empresas visualizar, analisar e compartilhar informações. Ela oferece recursos para a criação de dashboards interativos e personalizados, permitindo que os usuários explorem dados de várias fontes de forma intuitiva. Com o Loome Software, é possível conectar-se a diferentes sistemas de banco de dados, serviços de nuvem e planilhas, unificando os dados em um único local para análise. Porém, esta plataforma também apresenta um preço caro para implementação. (LOOMESOFTWARE, 2023).

1.3.3 The Reporting Hub

O The Reporting Hub é um portal de Business Intelligence baseado na web que se integra ao Power BI, incorporando os painéis na página web. Apesar da plataforma oferecer várias funcionalidades para a criação do seu hub de dashboards, atualmente ela possui suporte apenas para painéis feitos no Power BI e também é paga. (THE REPORTING HUB, 2023).

1.4 Diferencial da plataforma

Este trabalho vai apresentar a plataforma Vizage, detalhada na seção 4 **DESENVOLVIMENTO**, mas na *Tabela 1* já foram colocadas algumas das características para comparação com as outras plataformas do mercado.

Tabela 1 – Comparação da plataforma Vizage com outras plataformas do mercado

	Plataformas de BI suportadas	Valor para adquirir
Curator	Tableau, Power BI e ThoughtSpot	A partir de \$20k por ano
Loome	Tableau e Power BI	A partir de \$30k por ano
Reporting Hub	Power BI	A partir de \$2k por ano
Vizage	Inicialmente: Tableau e Power BI	Grátis

Fonte: Elaborado pelo autor

1.5 Estrutura do Documento

Os artefatos desenvolvidos neste trabalho são disponibilizados na forma de software livre e assim este TCC se caracteriza como Relatório Técnico + Produto Multimídia.

Este relatório técnico é composto por seis capítulos. No Capítulo 1, é feita uma introdução, que inclui a contextualização, a motivação, as questões de pesquisa e os objetivos. O Capítulo 2 aborda todos os conceitos teóricos necessários para uma melhor compreensão do projeto. O Capítulo 3 descreve os métodos científicos utilizados. O Capítulo 4 descreve as etapas adotadas para realizar o trabalho. No Capítulo 5, são analisados e discutidos os resultados experimentais da pesquisa de usabilidade para validar a plataforma desenvolvida. Por fim, o Capítulo 6 conclui o trabalho e apresenta alternativas do que pode ser feito no futuro para dar continuidade ao desenvolvimento da plataforma.

2 REFERENCIAL TEÓRICO

O propósito desta seção é fornecer uma introdução aos conceitos fundamentais que constituem a base do desenvolvimento da plataforma. Serão abordados conceitos de Design Thinking, requisitos de software, aplicações web, Vue.js e da área de BI. Esses conceitos variam desde as ferramentas utilizadas no desenvolvimento da plataforma até os princípios teóricos essenciais para um entendimento aprofundado do trabalho, fornecendo uma base sólida para a análise do projeto em questão.

2.1 Design Thinking

A metodologia Design Thinking é uma abordagem inovadora e centrada no ser humano para solucionar problemas complexos e promover a inovação. Segundo Brown (2008), o Design Thinking é "uma abordagem que combina empatia pelo contexto do problema, criatividade na geração de insights e soluções, e racionalidade na análise e planejamento". O principal motivo para a escolha desta metodologia foi a abordagem simplificada do autor, que contribuiu para uma compreensão mais profunda do processo de Design Thinking e sua aplicação prática.

Segundo Tim Brown (2008), sem uma fórmula pré-determinada, as etapas do Design Thinking são cinco, podendo variar de acordo com o autor, são elas: empatia, definição, ideação, prototipação e validação.

De modo resumido suas respectivas definições serão descritas abaixo:

- **Empatia:** Diz respeito à pesquisa do público alvo para quem você está projetando, por meio do conhecimento de seu ambiente e costumes.
- **Definição:** Diz respeito a análise de todos os problemas recorrentes no ambiente deste público ou pessoa para quem você está projetando, e escolher um problema específico para começar e resolver.
- **Ideação:** Após a escolha e definição do problema a montagem de um painel de ideias para solucionar este problema deve ser montado, todas as ideias que surgem e que podem ser úteis devem ser guardadas.
- **Prototipação:** Após a escolha de ideias viáveis algo tangível que possa ser testado com o público alvo deve ser construído. Depois da fase teste o feedback e resultados relevantes devem ser analisados.

- **Validação:** Com a finalização do projeto e readequações depois da fase de teste, o produto pode ser aplicado em uma situação real e sua experiência pode ser validada pelos usuários.

Figura 1 – As etapas do design thinking



Fonte: (Brown, 2008, com adaptações).

O Design Thinking destaca-se por sua abordagem iterativa e colaborativa, colocando o usuário no centro do processo de criação (Brown, 2008). Por meio da prototipagem rápida e do constante ciclo de feedback, as soluções são continuamente refinadas e melhoradas, garantindo uma abordagem flexível e adaptativa.

Em suma, a metodologia Design Thinking tem sido amplamente adotada por empresas e organizações como uma maneira eficaz de abordar problemas complexos e estimular a inovação centrada no ser humano. Ao combinar empatia, criatividade e racionalidade, o Design Thinking oferece uma abordagem sistemática para a geração de soluções inovadoras, capazes de atender às necessidades e expectativas dos usuários.

2.2 Requisitos de software

Os requisitos de software são as especificações que definem o que um sistema de software deve ser capaz de fazer e quais características ele deve possuir. Esses requisitos são essenciais para o desenvolvimento e implementação bem-sucedidos de um software.

De acordo com Sommerville (2011), os requisitos de software podem ser classificados em dois tipos principais: requisitos funcionais e requisitos não funcionais. Os requisitos funcionais descrevem as funcionalidades específicas que o sistema deve fornecer, ou seja, o que o sistema deve ser capaz de realizar. Esses requisitos geralmente são expressos em termos de entradas, processamentos e saídas do sistema. Por outro lado, os requisitos não funcionais se referem a atributos do sistema, como desempenho, segurança, usabilidade e confiabilidade. Eles representam características do sistema que não estão diretamente relacionadas às suas funcionalidades, mas são igualmente importantes para o seu sucesso.

A definição precisa dos requisitos de software é um passo crítico no processo de desenvolvimento de software, pois influencia diretamente o design, a implementação e o teste do sistema. Uma análise completa e detalhada dos requisitos é fundamental para evitar erros e lacunas na funcionalidade do software, além de garantir que o sistema desenvolvido atenda às expectativas dos usuários.

2.3 Aplicações Web

As aplicações web, também conhecidas como web apps, são sistemas de software projetados para serem acessados através de um navegador web, eliminando a necessidade de instalação prévia nos dispositivos dos usuários (SOUZA,2018).

Estas aplicações desempenham um papel fundamental no contexto tecnológico atual, oferecendo dois principais recursos aos usuários: Primeiramente, a acessibilidade. Os usuários podem acessar a aplicação de qualquer lugar, a qualquer momento, desde que tenham acesso à internet. Segundo, a facilidade de manutenção das plataformas. As atualizações e correções do projeto podem ser implementadas de forma centralizada no servidor, facilitando a manutenção e garantindo que todos os usuários estejam utilizando a versão mais atualizada (SOUZA,2018).

2.4 Framework Vue.js

Vue.js é um framework JavaScript amplamente utilizado para criar interfaces de usuário interativas em aplicações web. Desenvolvido por Evan You, o Vue.js é considerado uma das principais ferramentas para construir Aplicações Web. (FELIZARDO, 2018).

Uma das características mais marcantes do Vue.js é sua abordagem incremental. Isso significa que ele pode ser facilmente incorporado em projetos existentes, permitindo que os desenvolvedores adotem o Vue.js gradualmente e aproveitem seus recursos específicos conforme necessário. Essa flexibilidade torna o Vue.js uma escolha popular para desenvolvedores que desejam migrar ou adicionar funcionalidades a projetos existentes (VUEJS, 2023).

Outra característica notável do Vue.js é a sua simplicidade e curva de aprendizado suave. Com sua sintaxe intuitiva e documentação abrangente, o Vue.js permite que os desenvolvedores construam rapidamente componentes reutilizáveis e criem uma arquitetura escalável para suas aplicações. Além disso, o Vue.js oferece recursos poderosos, como diretivas, filtros, mixins e roteamento, que facilitam a criação de funcionalidades complexas (VUEJS, 2023).

O Vue.js também possui uma comunidade ativa e crescente, o que significa que há um vasto ecossistema de bibliotecas, ferramentas e plugins disponíveis para expandir suas capacidades. Essa comunidade engajada oferece suporte, compartilha recursos e contribui para o aprimoramento contínuo do framework (VUEJS, 2023).

2.5 Conceitos de Business Intelligence e Análise de dados

Business Intelligence (BI) refere-se ao conjunto de estratégias, tecnologias e práticas utilizadas para coletar, integrar, analisar e apresentar informações relevantes para o processo de tomada de decisão nas organizações. O objetivo do BI é transformar os dados brutos em conhecimento acionável, fornecendo insights valiosos que auxiliam no direcionamento estratégico, na identificação de oportunidades de negócio e na resolução de problemas (SILVA, 2017).

O BI envolve a coleta de dados de diversas fontes, como bancos de dados internos, sistemas transacionais, redes sociais, dispositivos IoT, entre outros. Esses dados são

processados, transformados e consolidados em um formato adequado para análise. Em seguida, são aplicadas técnicas de análise de dados para identificar padrões, tendências, correlações e insights significativos (SILVA, 2017).

A análise de dados, por sua vez, é o processo de examinar, interpretar e extrair informações úteis a partir dos dados coletados. Existem diferentes abordagens de análise de dados, incluindo análise estatística, análise descritiva, análise preditiva e análise prescritiva. Cada uma delas fornece uma perspectiva única sobre os dados e auxilia na compreensão do desempenho passado, na previsão de eventos futuros e na recomendação de ações a serem tomadas (SILVA, 2017).

As soluções de BI e análise de dados fornecem dashboards interativos, relatórios, gráficos e visualizações que facilitam a compreensão dos dados de forma clara e intuitiva. Isso permite que os usuários, como gestores, analistas e tomadores de decisão, explorem os dados, façam perguntas, identifiquem padrões e obtenham insights valiosos para orientar as estratégias de negócio (SILVA, 2017).

2.6 Ferramentas de Análise de dados

As ferramentas de análise de dados são softwares e plataformas que oferecem uma variedade de recursos e funcionalidades para auxiliar na análise de dados, tais como:

- Coleta e integração de dados
- Limpeza e preparação de dados
- Visualização de dados
- Análise estatística
- Análise exploratória de dados
- Modelagem preditiva

As ferramentas de análise de dados são utilizadas em uma ampla variedade de setores e organizações, desde pequenas empresas até grandes corporações. Elas desempenham um papel fundamental no auxílio à tomada de decisões informadas, identificação de oportunidades de negócios, otimização de processos e melhoria do desempenho empresarial (LAPA, 2015).

2.7 Dashboards de Análise de dados

Dashboards de análise de dados são painéis visuais interativos que consolidam e apresentam informações relevantes e insights derivados de conjuntos de dados. Eles são os produtos criados pelas ferramentas de BI e são projetados para fornecer uma visão rápida e abrangente do desempenho de uma organização, processo ou sistema, permitindo que os usuários monitorem e analisem os dados de forma intuitiva (LAPA, 2015).

Os dashboards são compostos por diferentes elementos, como gráficos, tabelas, medidores, mapas e indicadores-chave de desempenho (KPIs). Esses elementos são organizados de maneira clara e concisa, permitindo que os usuários identifiquem tendências, padrões, anomalias e insights relevantes de forma visual e compreensível (LAPA, 2015).

A principal finalidade dos dashboards é fornecer informações acionáveis em tempo real, ajudando as organizações a tomar decisões informadas e orientadas por dados. Eles permitem que os usuários monitorem o desempenho de métricas-chave, identifiquem áreas de oportunidade, acompanhem o progresso em relação às metas estabelecidas e detectem problemas potenciais (LAPA, 2015).

2.8 Plataformas web de Hospedagem de Dashboards nativas das ferramentas de BI

As plataformas web de hospedagem de dashboards nativas das ferramentas de Business Intelligence (BI) são sistemas online fornecidos pelas próprias empresas desenvolvedoras de software de BI, como Tableau, Power BI, QlikView, entre outras. Essas plataformas são projetadas para permitir que os usuários hospedem, compartilhem e acessem seus dashboards de análise de dados de forma centralizada e segura.

Essas plataformas web oferecem recursos específicos para facilitar a publicação e distribuição de dashboards desenvolvidos nas respectivas ferramentas de BI. Os usuários podem enviar seus dashboards para a plataforma, configurar permissões de acesso, criar painéis de controle personalizados e compartilhá-los com colegas de trabalho, clientes ou outras partes interessadas.

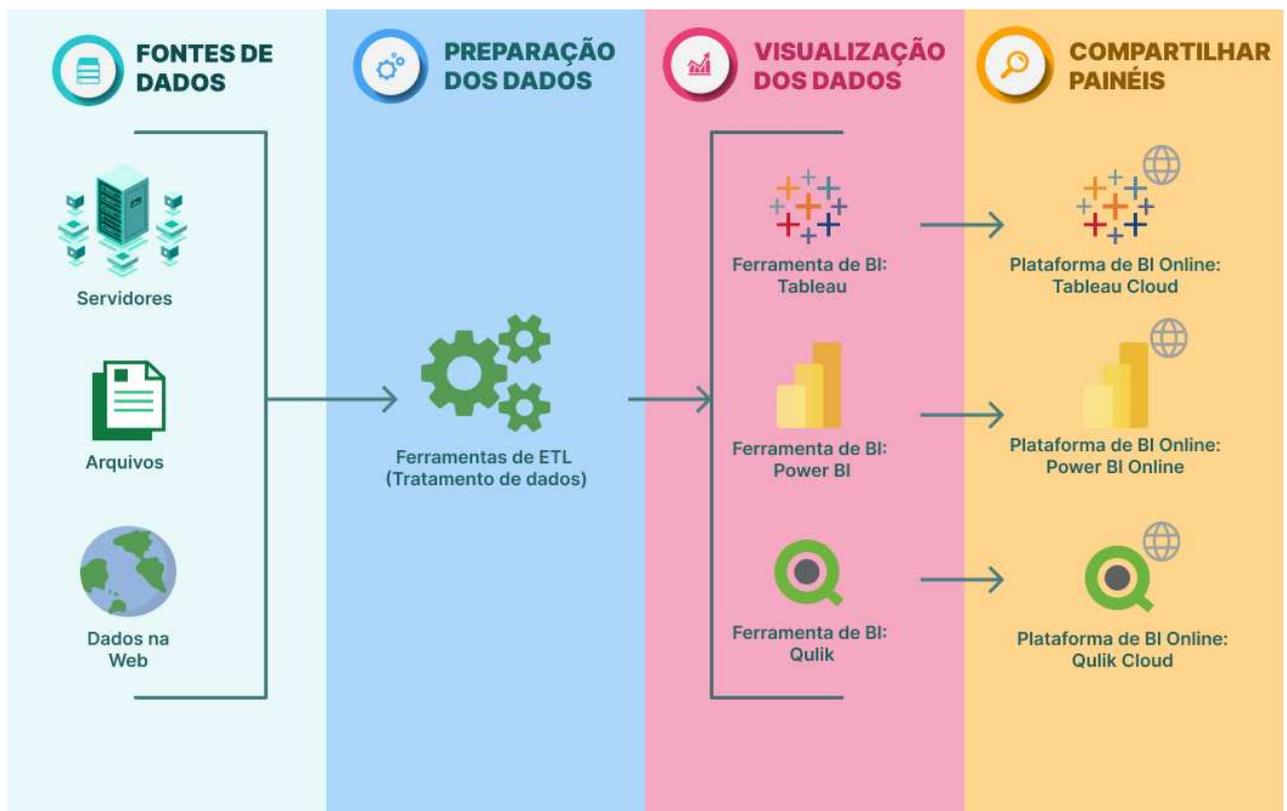
A escolha de utilizar a plataforma web nativa de hospedagem de dashboards da

ferramenta de BI está diretamente relacionada à familiaridade com a ferramenta, ao ecossistema existente na organização e à necessidade de recursos específicos oferecidos pela plataforma. Essas soluções são projetadas para fornecer uma experiência completa para os usuários que desejam utilizar os dashboards desenvolvidos na própria ferramenta de BI, porém essa também representa sua limitação, pois ela fornece suporte apenas para dashboards que foram criados na sua ferramenta de BI específica (LAPA, 2015).

2.9 O problema da integração e centralização de informações nas empresas

Partindo do princípio que precisamos centralizar informações em um único ambiente, primeiro temos que entender o processo que leva à criação dos dashboards de análise de dados, que são a chave da plataforma.

Figura 2 – As etapas do ciclo de vida dos dados



Fonte: Elaborado pelo autor.

Segundo a Tableau Software (2021), o processo de vida dos dados, desde uma base de dados até um dashboard publicado na nuvem e compartilhado para pessoas em uma empresa, envolve várias etapas importantes. Vamos percorrer essas etapas:

- Armazenamento dos dados: Os dados coletados são armazenados em uma base de dados centralizada, como um servidor, um banco de dados em nuvem, ou um arquivo Excel.
- Modelagem e preparação dos dados: Nesta etapa, os dados são modelados e preparados para a criação do dashboard. Isso envolve a seleção dos dados relevantes, a definição de métricas e indicadores-chave de desempenho (KPIs), a aplicação de cálculos e agregações, a criação de hierarquias e dimensões, entre outros processos de preparação. Nesse estágio, os dados podem ser processados, limpos, transformados e enriquecidos para garantir a qualidade e a integridade dos dados.
- Design e criação do dashboard: Com os dados preparados, inicia-se o processo de design e criação do dashboard. Isso envolve a escolha da ferramenta de visualização de dados, como Tableau, Power BI ou outra plataforma de BI, e a definição da estrutura, layout, gráficos, tabelas e outros elementos visuais do dashboard. O objetivo é criar uma interface intuitiva e informativa que permita aos usuários explorar e compreender os dados de maneira eficaz.
- Publicação na nuvem: Após a criação do dashboard, ele é publicado em uma plataforma de hospedagem na nuvem. Isso permite que o dashboard seja acessado e visualizado por usuários autorizados em qualquer lugar, a qualquer momento, utilizando apenas um navegador web. A publicação na nuvem também garante a escalabilidade e disponibilidade dos dashboards, além de facilitar as atualizações e o compartilhamento dos mesmos.

Uma vez publicado, o problema se encontra em como esse dashboard vai chegar no usuário final, pois como ilustrado na imagem acima, com a utilização de diferentes ferramentas de Business Intelligence (BI), as informações acabam dispersas em diversas plataformas diferentes, que podem variar em termos de credenciais para login, funcionalidades, interface de usuário e recursos oferecidos. Essa diversidade pode ser um desafio para os usuários finais, pois eles precisam aprender a utilizar cada plataforma separadamente para acessar todas as informações necessárias.

Esse cenário de múltiplas plataformas pode ser problemático por dois motivos principais:

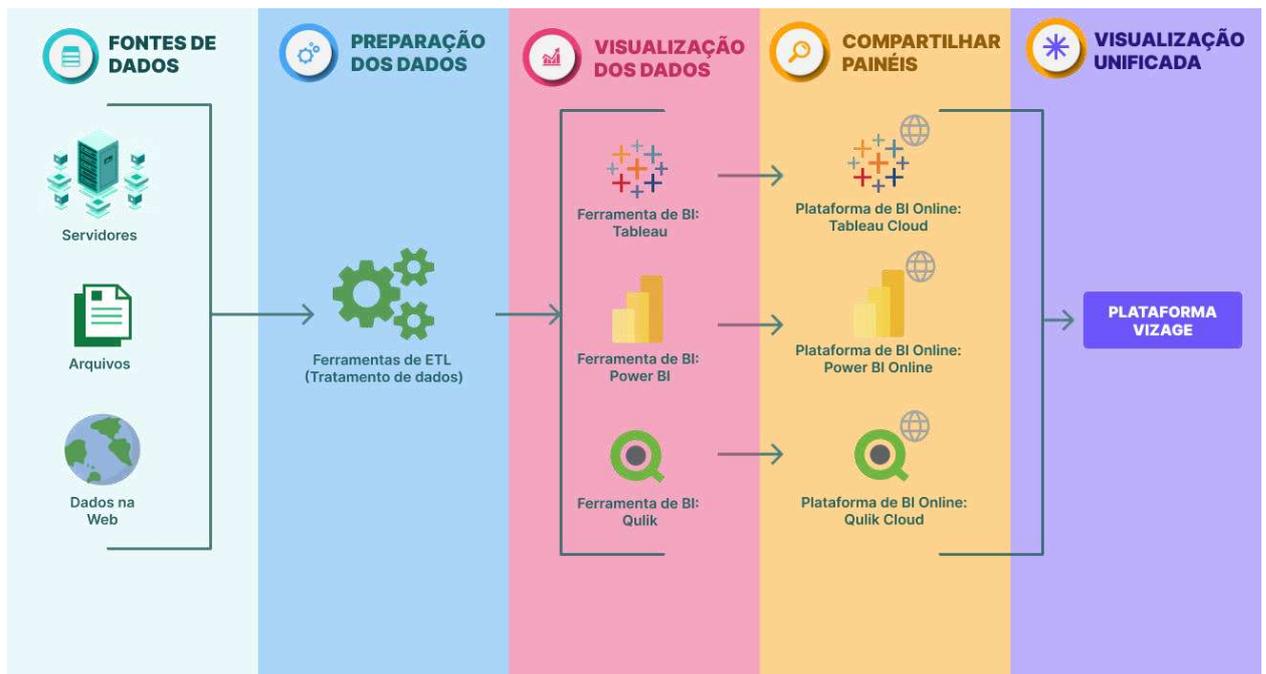
Primeiro, a curva de aprendizado pode ser acentuada para os usuários, já que eles precisam se familiarizar com a navegação e as ferramentas específicas de cada plataforma.

Isso demanda tempo e esforço, e pode resultar em uma experiência fragmentada e confusa.

Além disso, a dispersão das informações em diferentes plataformas pode dificultar a obtenção de uma visão completa e integrada dos dados. Os usuários podem ter que alternar entre várias interfaces e consultas diferentes para acessar diferentes partes do painel de controle, o que pode ser inconveniente e propenso a erros.

Nesse contexto, uma plataforma para hospedar os dashboards, que permita a integração e centralização dessas informações, desempenha um papel fundamental. A plataforma atuará como um repositório central, onde os dashboards criados em diferentes ferramentas de BI, como Tableau, Power BI, entre outras, podem ser agregados e visualizados em um só lugar. Ela adicionará uma camada de agregação, após os dashboards serem publicados em suas respectivas plataformas.

Figura 3 – As etapas do ciclo de vida dos dados, incorporando a plataforma VIZAGE



Fonte: Elaborado pelo autor.

A vantagem de ter uma plataforma de hospedagem de dashboards é que ela permite que os usuários acessem e interajam com as informações de forma intuitiva e integrada. Ao centralizar os dashboards em um único local, os usuários podem realizar análises comparativas, identificar correlações entre diferentes conjuntos de dados e obter uma

visão abrangente das informações empresariais. Tudo dentro da experiência de usuário única centralizada nesta plataforma.

2.10 Questionário SUS

O questionário SUS (System Usability Scale) é uma ferramenta utilizada para avaliar a usabilidade de sistemas, interfaces e produtos. Ele consiste em um conjunto de perguntas que visam medir a percepção dos usuários sobre a facilidade de uso de um determinado sistema ou interface. (BROOKE, 1986)

O questionário é criado a partir de 10 perguntas que serão avaliadas posteriormente (BROOKE, 1986):

1. Eu acho que gostaria de usar esse sistema com frequência.
2. Eu acho o sistema desnecessariamente complexo.
3. Eu achei o sistema fácil de usar.
4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
5. Eu acho que as várias funções do sistema estão muito bem integradas.
6. Eu acho que o sistema apresenta muita inconsistência.
7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
8. Eu achei o sistema atrapalhado de usar.
9. Eu me senti confiante ao usar o sistema.
10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

As perguntas são respondidas pelos participantes em uma escala de 5 pontos que variam entre “Discordo Totalmente” até “Concordo Totalmente”, onde o valor 1 representa a opção "Discordo Totalmente" e o valor 5 representa a opção "Concordo Totalmente", enquanto os valores intermediários 2, 3 e 4 são relativos aos extremos 1 e 5.

O SUS produz um único número que representa uma medida composta da usabilidade geral do sistema que está sendo estudado. Para calcular a pontuação do SUS, primeiro é preciso somar a pontuação de cada participante para cada pergunta. Para cada pergunta, a contribuição da pontuação varia de 0 a 4. Para as perguntas 1, 3, 5, 7 e 9, a contribuição da pontuação é a posição na escala de resposta menos 1. Para os itens 2, 4, 6, 8 e 10, a contribuição é 5 menos a escala posição. Por exemplo, se um participante respondeu 5

(Concordo totalmente) para a pergunta 1, o score desta pergunta para este participante seria 5 menos 1, ou seja, 4. Multiplique a soma das pontuações de cada pergunta por 2,5 para obter a pontuação do SUS para aquele participante. (BROOKE, 1986).

Após o cálculo das pontuações de todos os participantes, a pontuação SUS total para a aplicação é calculada com média simples dos scores dos participantes. As pontuações do SUS variam de 0 a 100 (BROOKE, 1986).

Por exemplo, a tabela abaixo mostra um exemplo de como seria calculada a pontuação de um participante:

Tabela 2 – Exemplo de resposta e pontuação do questionário SUS

	Resposta do participante	Score da Pergunta
1 - Eu acho que gostaria de usar esse sistema com frequência.	5 (Concordo totalmente)	4
2 - Eu acho o sistema desnecessariamente complexo.	4 (Concordo parcialmente)	1
3 - Eu achei o sistema fácil de usar.	2 (Discordo parcialmente)	1
4 - Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.	1 (Discordo totalmente)	4
5 - Eu acho que as várias funções do sistema estão muito bem integradas.	2 (Discordo parcialmente)	1
6 - Eu acho que o sistema apresenta muita inconsistência.	3 (Indiferente)	2
7 - Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.	2 (Discordo parcialmente)	1
8 - Eu achei o sistema atrapalhado de usar.	4 (Concordo parcialmente)	1
9 - Eu me senti confiante ao usar o sistema.	5 (Concordo totalmente)	4
10 - Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.	2 (Discordo parcialmente)	3

Fonte: Elaborado pelo autor.

A pontuação do participante vai ser calculada somando os scores das perguntas e multiplicando o resultado por 2,5. Portanto para esse participante:

- Soma dos scores das perguntas = 22
- Score SUS do participante = $22 * 2,5 = 55$

Segundo Bangor, Kortum e Miller (2009), a pontuação total do SUS para a aplicação, aquela que vai ser calculada pela média da pontuação dos participantes da pesquisa, pode ser classificada na seguinte escala de satisfação:

Tabela 3 – Classificação de satisfação com a pontuação SUS total de uma aplicação

Classificação	Pontuação SUS Total
Pior Possível	12.5
Péssimo	20.3
Ruim	35.7
OK	50.9
Bom	71.4
Excelente	85.5
Melhor Possível	90.0

Fonte: Elaborado pelo autor.

Esta tabela nos dará uma base para avaliar o que a pontuação adquirida pela plataforma desenvolvida representa em questão de qualidade da usabilidade.

3 METODOLOGIA

O presente trabalho mostra-se como sendo de caráter exploratório possui uma natureza quali-quantitativa, utilizando uma combinação de métodos, a fim de obter uma compreensão abrangente do problema de pesquisa. (CRESWELL; CRESWELL, 2017).

Figura 4 – Etapas da metodologia



Fonte: Elaborado pelo autor.

A primeira etapa consistiu na investigação de problemas relacionados à centralização de painéis de análise de BI publicados em diferentes plataformas, e, partir disso, o levantamento de requisitos, o qual teve como objetivo compreender as necessidades e expectativas em relação à plataforma a ser desenvolvida.

Com base nos requisitos levantados, deu-se início à segunda etapa, que consiste na prototipação e desenvolvimento da plataforma. Nessa fase, foi utilizada a ferramenta Figma para prototipação da interface e da navegação. A abordagem iterativa do Design Thinking permitiu testar e refinar os protótipos, buscando uma solução que melhor atendesse aos requisitos e expectativas do projeto. O desenvolvimento da plataforma em si foi feito na ferramenta Visual Studio Code, utilizando a framework de Javascript Vue.js. Durante o desenvolvimento da plataforma, foi dada atenção especial à usabilidade, à acessibilidade e à experiência do usuário. Foram aplicados princípios de design centrado no usuário para garantir uma interface intuitiva e funcional. A análise dos protótipos de interface da ferramenta será aprofundada posteriormente na seção **4.2.3.2** deste documento.

A última etapa concentrou-se na avaliação da usabilidade da plataforma, correspondendo à fase de “Testes e Validação” do Design Thinking. Nessa etapa, a plataforma foi submetida a testes práticos com usuários reais, observando suas interações e coletando

feedback valioso. Essa abordagem iterativa permitiu identificar pontos de melhoria e promover ajustes na plataforma antes de sua implementação final.

Além dos testes práticos, também foi aplicado um questionário do tipo SUS para obter percepções mais aprofundadas sobre a usabilidade e a experiência geral com a plataforma. A análise dos resultados destes testes será aprofundada posteriormente na seção **5.2** deste documento.

4 DESENVOLVIMENTO

Neste capítulo será mostrado o processo de desenvolvimento da aplicação, passando pelo estágio de pesquisa, a metodologia aplicada, a explicação das ferramentas utilizadas e o desenvolvimento propriamente dito.

É importante ressaltar que, dentro do escopo desta primeira versão da plataforma, não foi implementada nenhum tipo de integração com APIs das plataformas de BI para realmente importar os painéis publicados nestas plataformas. Dado ao tempo disponível para o processo de desenvolvimento, optou-se por criar um protótipo que emulasse a interface e usabilidade da plataforma para possibilitar os testes com base no critério de usabilidade. Embora a usabilidade e a navegação tenham se mostrado muito próximas do esperado para uma versão final, a integração com APIs e importação de dashboards reais é uma funcionalidade que foi planejada, mas ficará para ser implementada em versões futuras da plataforma.

O processo de importação de dashboards que foi usado nessa primeira versão do projeto será aprofundado na seção 4.2.3.7.

4.1 Escolha de tecnologias e ferramentas

1. Front-end:

- Figma para prototipação da interface.
- Framework Vue.js para desenvolvimento da interface.
- Visual Studio Code para escrita do código fonte.

2. Back-end:

- FireBase como banco de dados.
- Visual Studio Code para escrita do código fonte.

4.2 Processo de desenvolvimento

O processo de desenvolvimento seguiu a metodologia de design thinking descrita na seção 2.1 do Referencial Teórico, portanto foi um processo iterativo seguindo as etapas

estabelecidas para este trabalho no capítulo 3 **METODOLOGIA**. As etapas do desenvolvimento foram: Levantamento de requisitos, prototipação e desenvolvimento, avaliação do desenvolvimento.

4.2.1 Levantamento de requisitos

Como citado na seção 2.2 do Referencial Teórico, o levantamento de requisitos é uma etapa importante no desenvolvimento do projeto. Os requisitos funcionais foram definidos em processos de brainstorm, refinados por dois pesquisadores, e estão divididos em duas seções apresentadas nas tabelas a seguir: Requisitos de Back-end (Tabela 4) e Requisitos de Front-end (Tabela 5).

Tabela 4 – Requisitos Funcionais do Back-end

RF01	Realizar a persistência e coleta de dados de um banco externo não relacional (FireBase)
RF02	Definição de um modelo de dados para os dashboards
RF03	Realização das operações C.R.U.D para dashboards importados

Fonte: Elaborado pelo autor.

Tabela 5 – Requisitos Funcionais do Front-end

RF01	Menu de navegação para telas de dispositivos mobile e desktop, contendo as opções: “Início”, “Painéis”, “Usuários” e “Entrar”
RF02	As páginas “Usuários” e “Entrar” deverão retornar um conteúdo do tipo “Em breve”. Dado o tempo de desenvolvimento, o escopo foi reduzido para entregar apenas as páginas “Início” e “Painéis” apresentando conteúdo (que é onde os dashboards de análise de dados serão apresentados)
RF03	Página “Início” exibindo um texto de boas-vindas à plataforma, um texto explicando um pouco da funcionalidade, uma imagem de introdução e um botão para nos levar à página de dashboards
RF04	Página “Painéis” exibindo um botão para importar um dashboard publicado na plataforma da Tableau e um dashboard publicado na plataforma do PowerBI e uma galeria de dashboards já importados
RF05	Cada dashboard já presente na galeria deve conter uma seção com os seguintes elementos: Ícone de qual plataforma pertence (por enquanto, Tableau ou PowerBI),

	título, autor, data da publicação, um botão para ver o dashboard e um botão para deletá-lo
RF06	Ao clicar em adicionar um novo dashboard (seja ele de origem da Tableau ou do PowerBI), deve abrir um pop-up com um formulário para ser populado com as informações do dashboard que será incluído na galeria. Essas informações serão: Qual dashboard da plataforma de origem será importado (que será selecionado através de um <i>dropdown</i> , onde terá o nome dos dashboards hospedados na plataforma de origem), nome do dashboard e descrição
RF07	Ao final do formulário, deve existir um botão de confirmação, que, ao clicado, deverá validar o formulário e criar o dashboard com as informações preenchidas
RF08	Ao ser criado um novo dashboard, deve aparecer na galeria a seção com suas informações
RF09	Ao clicar no botão para visualizar o dashboard, o usuário deve ser levado para a página de visualização, onde terá o dashboard completo para ser analisado. Ao clicar no botão de deletar, o dashboard deve ser deletado

Fonte: Elaborado pelo autor.

4.2.2 Desenvolvimento: Back-end

4.2.2.1 Visão geral

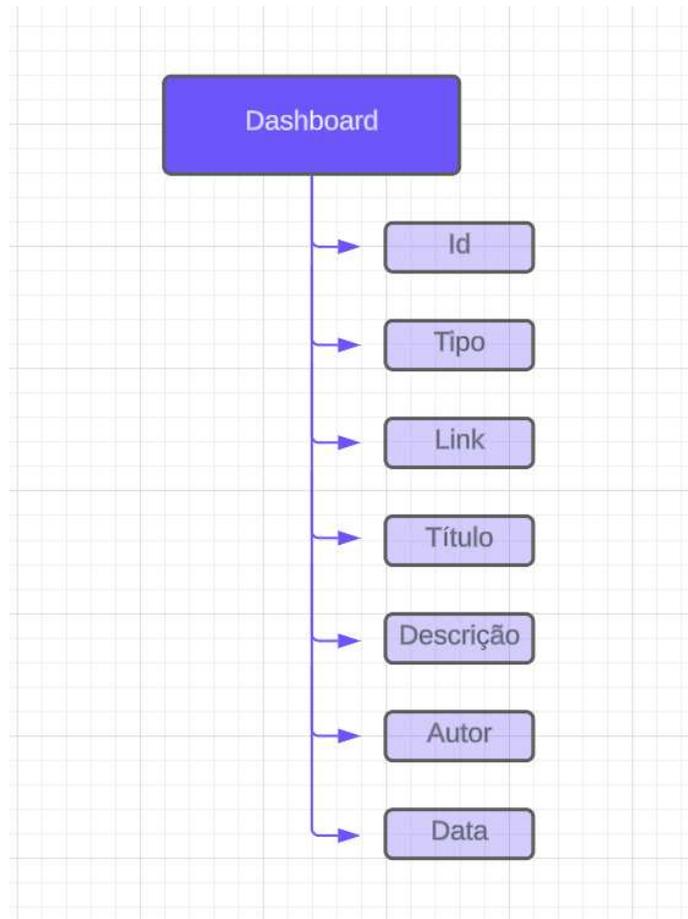
Além de utilizar o a linguagem JavaScript, seu repositório também conta com outras bibliotecas auxiliares como:

- Cors para permitir requisições originadas do front-end.
- Babel para compilar o código JavaScript.

4.2.2.2 Modelo de dados

Foi preciso criar um modelo de dados para padronização dos dashboards e suas informações dentro da aplicação. Para isso, usamos o seguinte modelo (Figura 5):

Figura 5 – Modelo de dados de um dashboard



Fonte: Elaborado pelo autor.

1. Atributo “Id” para armazenar o Id único vinculado ao dashboard (tipo: texto).
2. Atributo “Tipo” para armazenar o tipo do dashboard (tipo: texto).
3. Atributo “Link” para armazenar o link de acesso referente ao dashboard (tipo: texto).
4. Atributo “Título” para armazenar o título (tipo: texto).
5. Atributo “Descrição” para armazenar a descrição (tipo: texto).
6. Atributo “Autor” para armazenar o nome do autor (tipo: texto).
7. Atributo “Data” para armazenar a data de importação (tipo: texto).

Durante a troca de informações entre o servidor e o cliente, ocorre a conversão do modelo de dados da sua linguagem original para o formato JSON, e vice-versa. Esse formato é utilizado para transmitir os dados entre o front-end e o back-end, conforme as requisições são feitas entre eles (Figura 6).

Figura 6 – Objeto dashboard sendo passado em formato JSON para o banco de dados

```

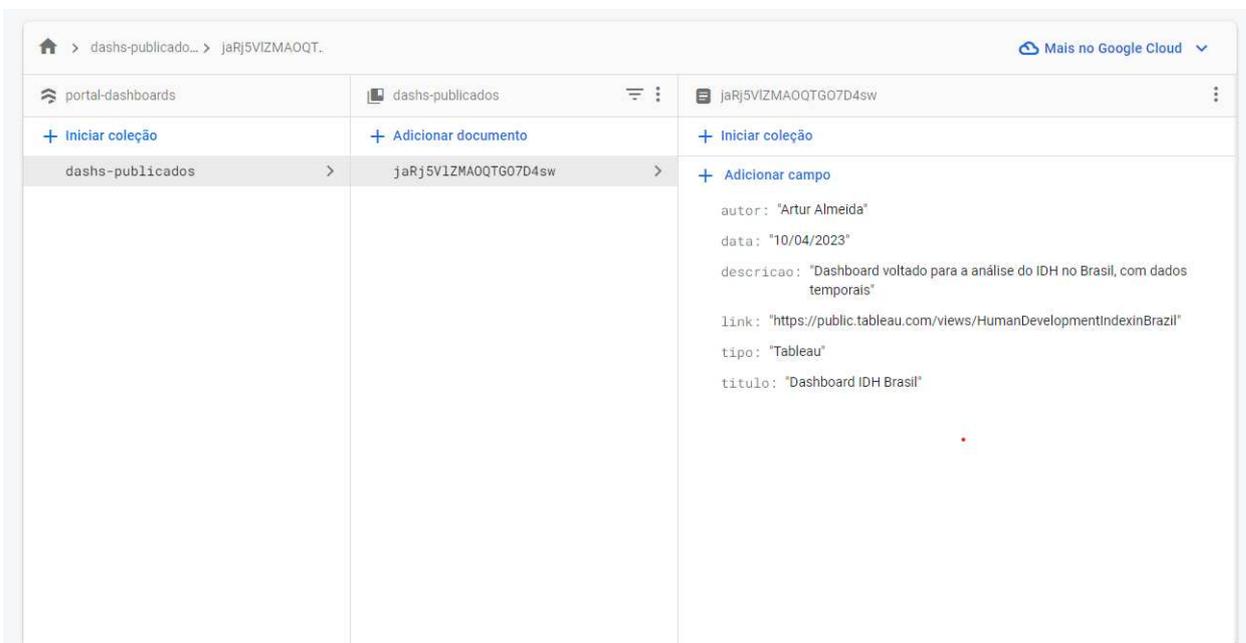
autor: "Artur Almeida"
data: "10/04/2023"
descricao: "Dashboard voltado para a análise do IDH no Brasil, com dados temporais"
id: "jaRj5VIZMAOQTG07D4sw"
link: "https://public.tableau.com/views/HumanDevelopmentIndexinBrazil"
tipo: "Tableau"
titulo: "Dashboard IDH Brasil"

```

Fonte: Elaborado pelo autor.

E então, os dados são gravados no banco de dados Firebase com o modelo definido (Figura 7):

Figura 7 – Dashboard gravado no banco de dados Firebase



Fonte: Elaborado pelo autor.

4.2.2.3 Conexão com o banco de dados Firebase

Esta plataforma utiliza da funcionalidade Cloud Firestore Database, que nos possibilita criar um banco de dados flexível e escalonável para desenvolvimento focado em dispositivos móveis, Web e servidores pelo Firebase. Ele mantém seus dados em sincronia em tempo real.

Como mostrado na figura 8, a conexão do servidor com o banco é feita no arquivo firebase.js, onde são passados todos os parâmetros necessários para que haja a conexão por meio da função initializeApp().

Figura 8 – Código de conexão com o Firebase

```
JS firebase.js X
src > JS firebase.js > getDash
1  import firebase from 'firebase/compat/app'
2  import 'firebase/compat/auth';
3  import 'firebase/compat/firestore';
4
5
6  const config = {
7    apiKey: "AIzaSyCJf666PedDse3SucekeqS4Est1534tNVM",
8    authDomain: "portal-dashboards.firebaseio.com",
9    projectId: "portal-dashboards",
10   storageBucket: "portal-dashboards.appspot.com",
11   messagingSenderId: "52256734347",
12   appId: "1:52256734347:web:249d7461dd7f82aa6d288e"
13 }
14
15 const firebaseApp = firebase.initializeApp(config)
16
17 const db = firebaseApp.firestore()
18 const publishedDashesCollection = db.collection('dashes-publicados')
19
20
```

Fonte: Elaborado pelo autor.

A variável *publishedDashesCollection* é responsável por receber a conexão com a coleção de dados chamada “dashes-publicados”, que é onde os dados dos dashboards serão armazenados. É por meio dela que serão realizadas as operações C.R.U.D.

4.2.2.4 Definição das operações C.R.U.D

Após a conexão com o banco de dados, é preciso definir as operações CRUD. A sigla CRUD significa as iniciais das operações create (criação), read (leitura), update (atualização) e delete (exclusão). Essas quatro siglas tratam a respeito das operações básicas executadas em bancos de dados (Figura 9).

Figura 9 – Código das operações C.R.U.D no arquivo “firebase.js”

```

JS firebase.js X
src > JS firebase.js > ...
20
21 export const createDash = dash => {
22   return publishedDashesCollection.add(dash)
23 }
24
25 export const getDash = async id => {
26   const dash = await publishedDashesCollection.doc(id).get()
27   return dash.exists ? dash.data() : null
28 }
29
30 export const updateDash = (id, dash) => {
31   return publishedDashesCollection.doc(id).update(dash)
32 }
33
34 export const deleteDash = id => {
35   return publishedDashesCollection.doc(id).delete()
36 }
37
38 export const useLoadDashes = () => {
39   const dashes = ref([])
40   const close = publishedDashesCollection.onSnapshot(snapshot => {
41     dashes.value = snapshot.docs.map(doc => ({ id: doc.id, ...doc.data() })))
42   })
43   onUnmounted(close)
44   return dashes
45 }

```

Fonte: Elaborado pelo autor.

A função `createDash()` é responsável por receber o objeto dashboard do front-end e criar uma instância desse dashboard no banco de dados.

A função `getDash()` é responsável por receber um Id e retornar o dashboard do banco de dados que corresponde à este Id.

A função `updateDash()` é responsável por atualizar as informações de um dashboard, especificado pelo Id.

A função `deleteDash()` é responsável por deletar as informações de um dashboard e apagá-lo por completo da base.

A função `useLoadDashes()` é responsável por retornar uma lista de todos os dashboards que existem na base.

4.2.3 Desenvolvimento: *Front-end*

4.2.3.1 *Visão geral*

O código-fonte do front-end é desenvolvido utilizando o framework Vue.js, além da própria biblioteca do Vue, o repositório front-end utiliza outras bibliotecas complementares, tais como:

- Vue router para gerenciamento de rotas.
- Bootstrap.css para gerenciamento do layout.
- Tailwind.css para auxiliar no gerenciamento do layout.
- Sass como pre-processor CSS.

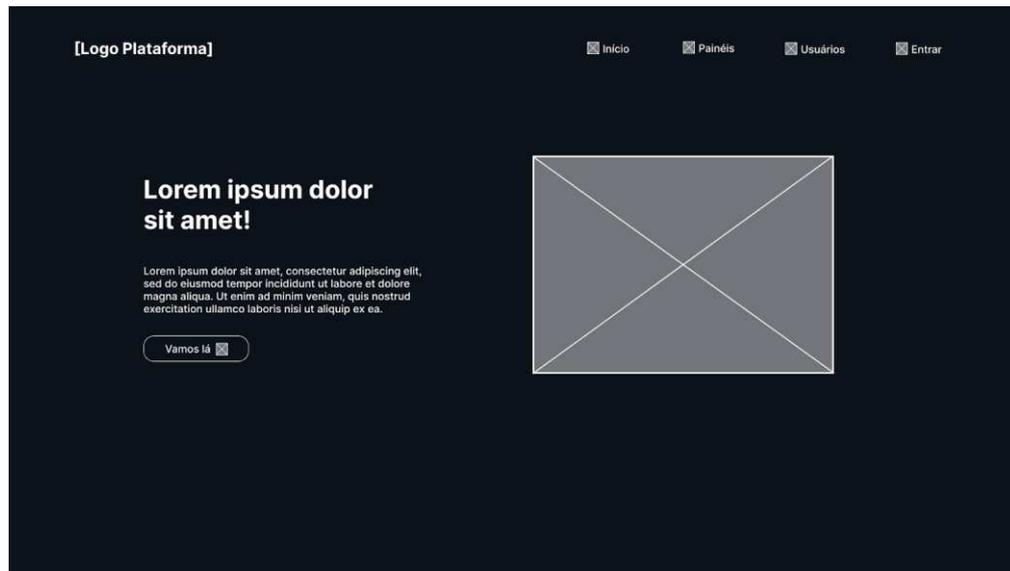
4.2.3.2 *Protótipo da Interface*

Segundo Norman (2004), wireframes permitem que os designers tenham uma visão abrangente e estruturada do projeto, focando nas funcionalidades principais e na usabilidade do sistema. Eles servem como um guia visual para alinhar as expectativas e as necessidades.

Seguindo esse conceito, foram criados protótipos no modelo de *wireframes* para facilitar o entendimento do layout das principais telas da plataforma, que são: Home, painéis (dashboards), visualização de um painel e a tela de “página não encontrada” que será utilizada para as páginas que não entraram no escopo deste projeto.

A Figura 10 mostra o protótipo da página Home, onde há o menu superior de navegação, um texto de boas-vindas e um botão para levar à página de painéis.

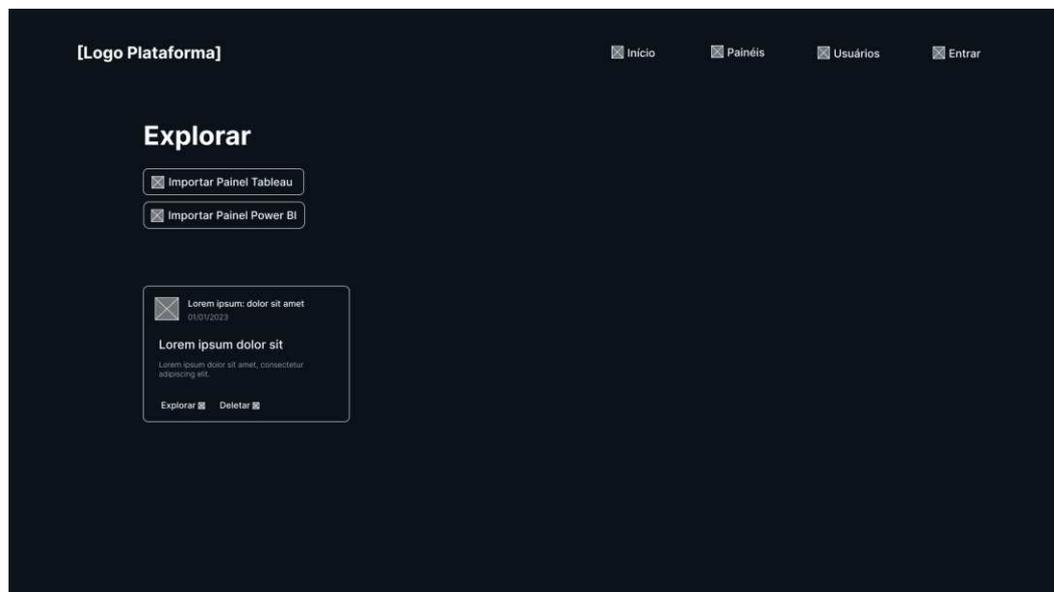
Figura 10 – Protótipo da página Home



Fonte: Elaborado pelo autor.

A Figura 11 mostra o protótipo da página Painéis, onde há o menu superior de navegação, os botões para importação de dashboards e a galeria com cartões de informações dos dashboards já importados.

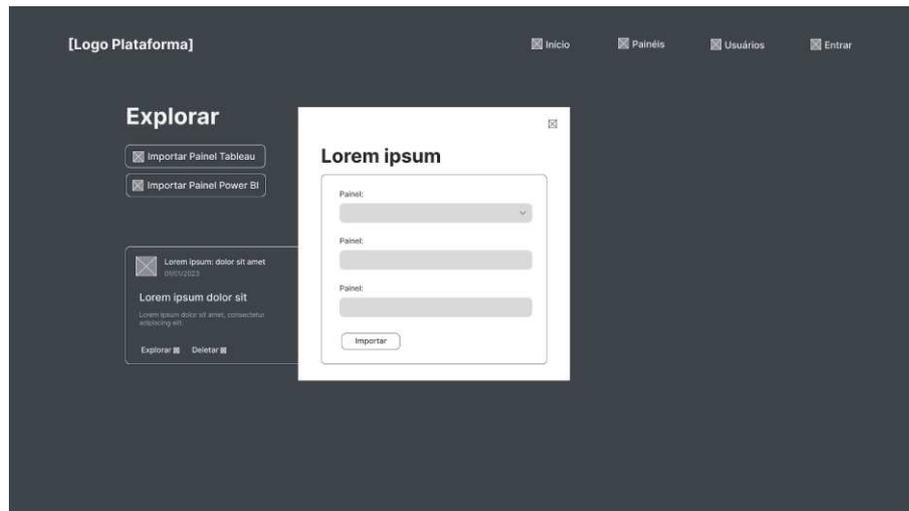
Figura 11 – Protótipo da página Painéis



Fonte: Elaborado pelo autor.

A Figura 12 mostra o protótipo da página Painéis quando um dos botões de importação é clicado e o formulário para importação do dashboard aparece na tela.

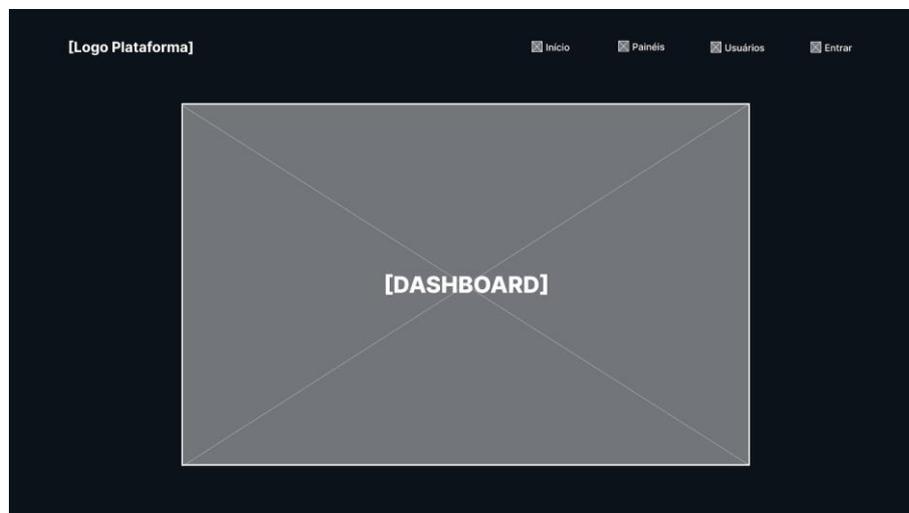
Figura 12 – Protótipo do pop-up da página Painéis



Fonte: Elaborado pelo autor.

A Figura 13 mostra o protótipo da página Explorar, que apresenta o menu superior de navegação e o dashboard que está sendo visualizado.

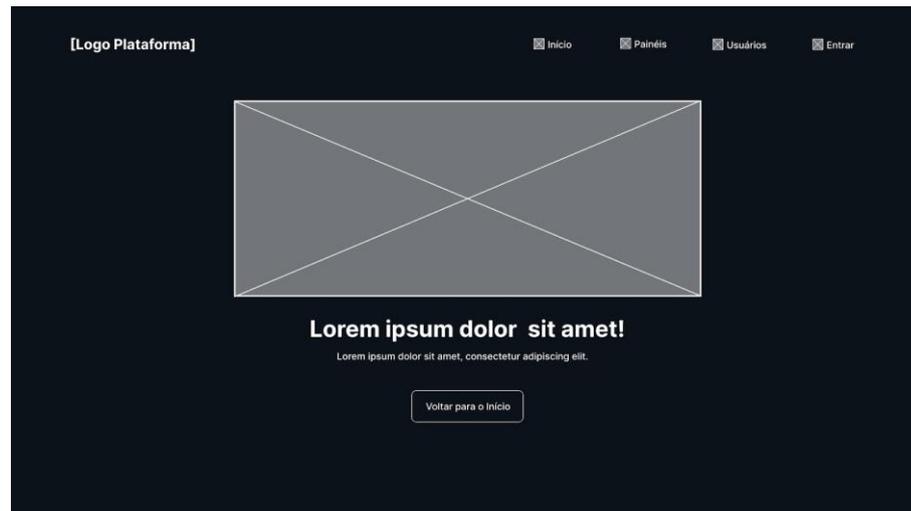
Figura 13 – Protótipo da página Explorar



Fonte: Elaborado pelo autor.

A Figura 14 mostra o protótipo da página “Página não encontrada” que apresenta o menu superior de navegação, um texto explicando a ausência da página e um botão para retornar à Home.

Figura 14 – Protótipo da página “Página não encontrada”

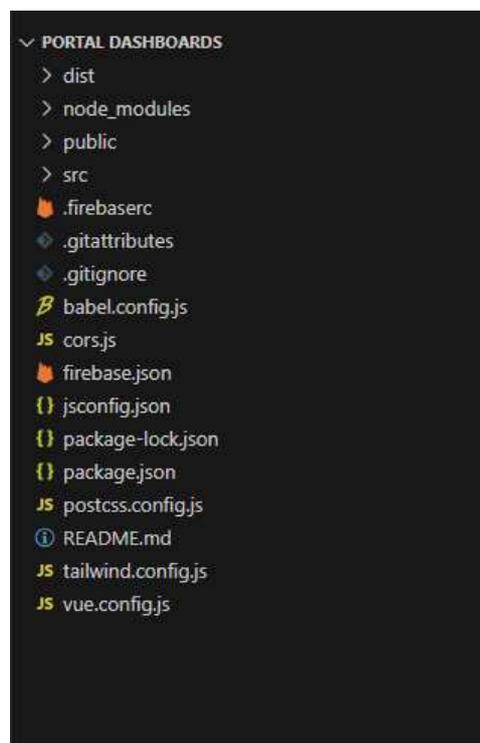


Fonte: Elaborado pelo autor.

4.2.3.3 Estrutura de pastas do projeto

Figura 15 mostra o diretório raiz, que foi criado e organizado pelo próprio vue que, a partir do comando “vue create portal-dashboards”, já prepara toda estrutura necessária para um projeto vue.

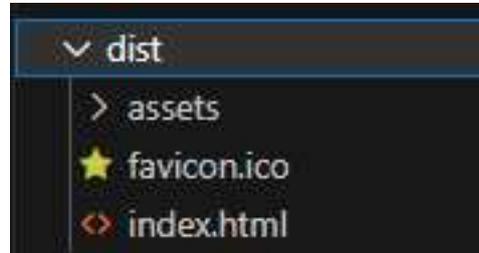
Figura 15 – Diretório raiz do projeto



Fonte: Elaborado pelo autor.

Pasta **dist**: é onde ficam localizados os arquivos gerados no final do projeto, como é mostrado na Figura 16. Estes arquivos são gerados através do comando “npm run build”.

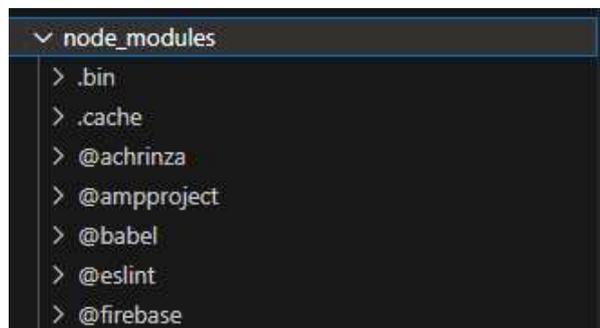
Figura 16 – Diretório pasta “dist”



Fonte: Elaborado pelo autor.

Pasta **node_modules**: é onde ficam localizados os pacotes de todas as bibliotecas necessárias para a aplicação funcionar. Na Figura 17, é mostrado apenas uma parte das bibliotecas, para termos uma idéia de como a pasta node_modules é organizada

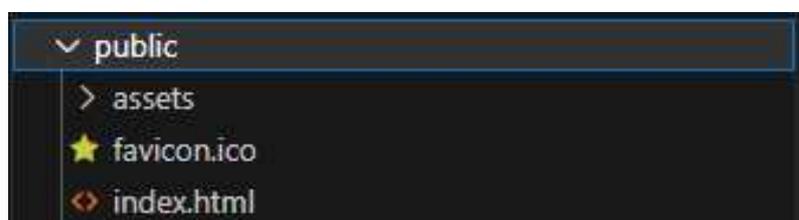
Figura 17 – Diretório pasta “node_modules”



Fonte: Elaborado pelo autor.

Pasta **public**: como mostrado na Figura 18, é onde ficam localizados alguns arquivos importantes, como o arquivo index.html. É neste arquivo que está presente a tag raiz `<div id='app'></div>`, ela faz o vínculo da instância do vue ao HTML, sendo responsável por renderizar as telas da aplicação no browser.

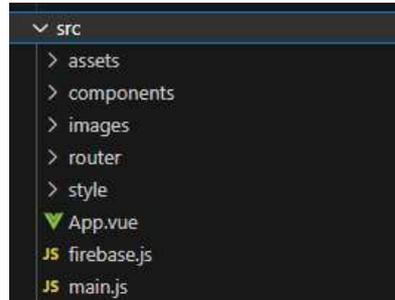
Figura 18 – Diretório pasta “public”



Fonte: Elaborado pelo autor.

Pasta **src**: é a pasta onde se encontram os arquivos de desenvolvimento do projeto, o trabalho de desenvolvimento se concentra principalmente nessa pasta. Como mostrado na Figura 19.

Figura 19 – diretório pasta “src”



Fonte: Elaborado pelo autor.

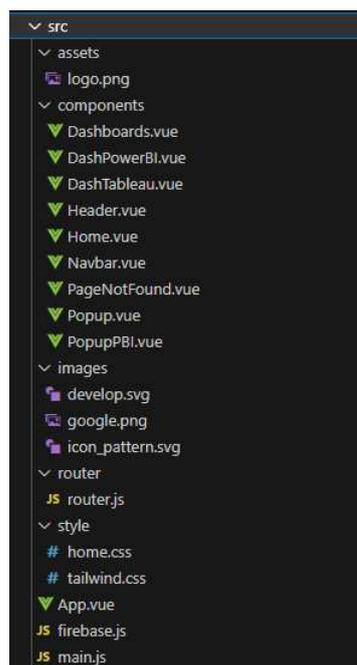
A pasta src é dividida em:

As pastas **assets** e **images** que possuem alguns ícones e imagens utilizados na plataforma (Figura 20).

Na pasta **components** ficam os arquivos de componentes .vue gerais da aplicação (Figura 20). Nestes arquivos estão os códigos-fonte das páginas da plataforma.

Na pasta **styles** ficam alguns arquivos .css que são utilizados para customizar a aparência de elementos nas páginas (Figura 20).

Figura 20 – Diretório expandido da pasta “src”



Fonte: Elaborado pelo autor.

A pasta **router** possui o arquivo **router.js**, que é onde vinculamos cada página a uma rota, como por exemplo a rota “/dashboards” que está vinculada ao componente Dashboards.vue (linhas 17 e 18 da Figura 21), referente à página de painéis. Essas rotas ficam armazenadas e são utilizadas quando chamadas em alguma parte da aplicação, fazendo com que o usuário seja direcionado para a página atrelada àquela rota.

Figura 21 – Código das rotas no arquivo “router.js”

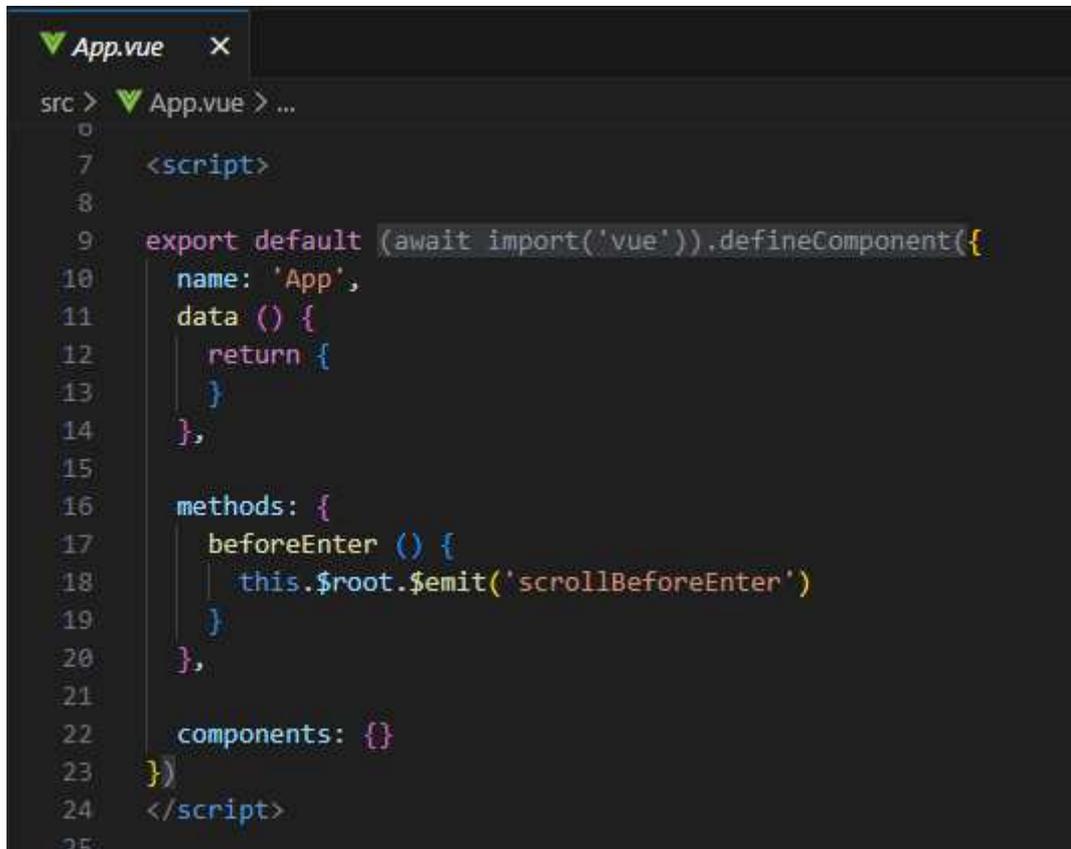
```
JS router.js x
src > router > JS router.js > ...
 1  import { createWebHistory, createRouter } from 'vue-router';
 2
 3  import Home from '../components/Home.vue';
 4  import Dashboards from '../components/Dashboards.vue';
 5  import DashTableau from '../components/DashTableau.vue';
 6  import DashPowerBI from '../components/DashPowerBI.vue';
 7  import PageNotFound from '../components/PageNotFound.vue';
 8
 9
10
11
12  const routes = [{
13    path: "/",
14    component: Home,
15  },
16  {
17    path: "/dashboards",
18    component: Dashboards,
19  },
20  {
21    path: "/dashTableau/:id",
22    component: DashTableau,
23  },
24  {
25    path: "/dashPowerBI/:id",
26    component: DashPowerBI,
27  },
28  {
29    path: '/*',
30    name: "PageNotFound",
31    component: PageNotFound,
32  },
33 ];
34
35
```

Fonte: Elaborado pelo autor.

O arquivo **App.vue**, como mostrado na Figura 22, exporta o código que está vinculado ao arquivo public/index.html, ele que irá se conectar com o HTML de entrada da versão de desenvolvimento da aplicação. Dessa forma, o arquivo App.vue desempenha o papel fundamental de ser o componente central que preenche o conteúdo do HTML inicial.

Ele serve como ponto de partida para todo o desenvolvimento do código, sendo o elemento central na árvore de componentes Vue.

Figura 22 – Código no arquivo “App.vue”



```
App.vue X
src > App.vue > ...
7 <script>
8
9 export default (await import('vue')).defineComponent({
10   name: 'App',
11   data () {
12     return {
13     },
14   },
15
16   methods: {
17     beforeEnter () {
18       this.$root.$emit('scrollBeforeEnter')
19     }
20   },
21
22   components: {}
23 })
24 </script>
25
```

Fonte: Elaborado pelo autor.

O arquivo **firebase.js** possui o código da conexão com o banco de dados Firebase e já foi explicado nas seções 4.2.2.3 e 4.2.2.4.

O arquivo **main.js** é onde a própria instância Vue é inicializada. Ele herda propriedades e métodos (como o “router”) que fazem o Vue funcionar da maneira que funciona (Figura 23).

Figura 23 – Código no arquivo “main.js”

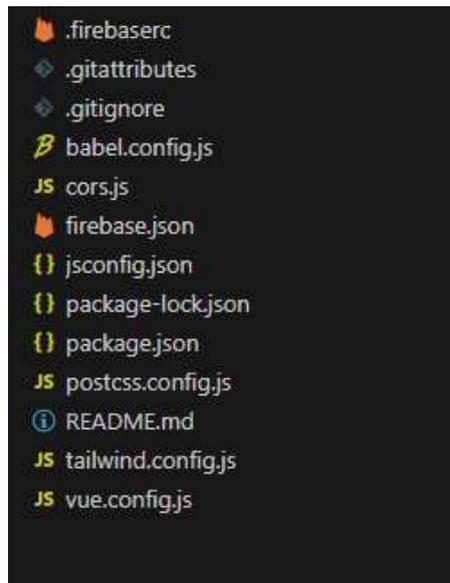
```
JS main.js X
src > JS main.js > ...
1  import { createApp } from 'vue'
2  import App from './App.vue'
3
4  import "bootstrap/dist/css/bootstrap.min.css";
5  import "bootstrap";
6
7  import router from './router/router.js';
8  import './style/home.css';
9  import './style/tailwind.css';
10
11 import { MotionPlugin } from '@vueuse/motion'
12 |
13 const app = createApp(App);
14
15 app.use(MotionPlugin);
16 app.use(router);
17
18
19 app.mount("#app");
```

Fonte: Elaborado pelo autor.

4.2.3.4 Arquivos de configurações

Os demais arquivos presentes na raiz do projeto são arquivos de configurações diversas, desde configurações de bibliotecas específicas como “babel.config.js” (que cuida da interpretação do código JavaScript), até configurações de gerenciamento dos pacotes utilizados na aplicação, como o “package.json” (Figura 24).

Figura 24 – Diretório dos arquivos de configurações



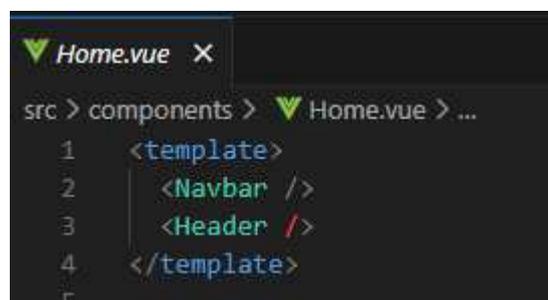
Fonte: Elaborado pelo autor.

4.2.3.5 Anatomia dos componentes principais

Cada componente `.vue` presente na pasta `src/components` é um pedaço de código que representa algo que é renderizado na página. E, utilizando a tag `<template>`, cada componente pode ser exportado para ser utilizado em outros componentes maiores para compor as páginas.

Por exemplo, o componente **Home.vue** (Figura 25) é formado pelos componentes **Navbar.vue**, que representa o menu superior de navegação, e **Header.vue**, que representa o texto de boas-vindas e a imagem que vemos na Home da plataforma.

Figura 25 – Tag template do componente Home.vue



Fonte: Elaborado pelo autor.

E, portanto, os arquivos Navbar.vue e Header.vue possuem o código-fonte que está encapsulado nas tags que vemos com os nomes dos componentes na Home.vue. Podemos ver parte do código do componente Navbar na Figura 26, e do Header na Figura 27.

Figura 26 – Tag template do componente Navbar.vue

```

Navbar.vue X
src > components > ▾ Navbar.vue > ...
1 <template>
2 <nav
3   class="p-3 bg-black">
4   <div class="container flex flex-wrap items-center justify-between mx-auto">
5     <p class="flex items-center hover:no-underline">
6       <img src="" class="h-6 mr-3 sm:h-10" />
7       <span
8         class="self-center text-white text-xl font-semibold whitespace-nowrap ">
9         [ VIZAGE LOGO ]
10      </span>
11    </p>
12  </div>
13  <div class="hidden w-full md:block md:w-auto">...
14 </div>
15 </nav>
16 </template>
17

```

Fonte: Elaborado pelo autor.

Figura 27 – Tag template do componente Header.vue

```

Header.vue X
src > components > ▾ Header.vue > ...
1 <template>
2 <header class="page-banner home-banner">
3   <div class="page-banner home-banner" >
4     <div class="container">
5       <div class="row home-banner" style="padding-top: 130px;">
6         <div class="col-lg-6 align-items-center py-3 wow fadeInUp">
7           <p
8             v-motion
9             :initial="{ opacity: 0, x: 200 }"
10            :enter="{ opacity: 1, x: 0 }"
11            :delay="500"
12            class="text-blue-900 text-2xl lg:text-4xl font-medium lg:font-semibold mb-5 lg:leading-normal leading-snug">
13            Bem-vind@ à nossa Plataforma de Dashboards!
14          </p>
15          <p
16            v-motion
17            :initial="{ opacity: 0, y: 200 }"
18            :enter="{ opacity: 1, y: 0 }"
19            :delay="1200"
20            class="hidden lg:block text-2xl mb-5"
21            style="font-size:20px;">
22            Explore as possibilidades ilimitadas de análise, acompanhe métricas-chave e tome decisões informadas.
23            Estamos empolgados em fazer parte da sua jornada de descoberta e sucesso analítico.
24            Comece a explorar o poder dos seus dados agora mesmo!
25          </p>
26        </div>
27        <router-link...
28      </router-link>
29    </div>
30  </div>
31  <div class="col-lg-6 wow zoomIn">...
32 </div>
33 </div>
34 </div>
35 </header>
36 </template>
37

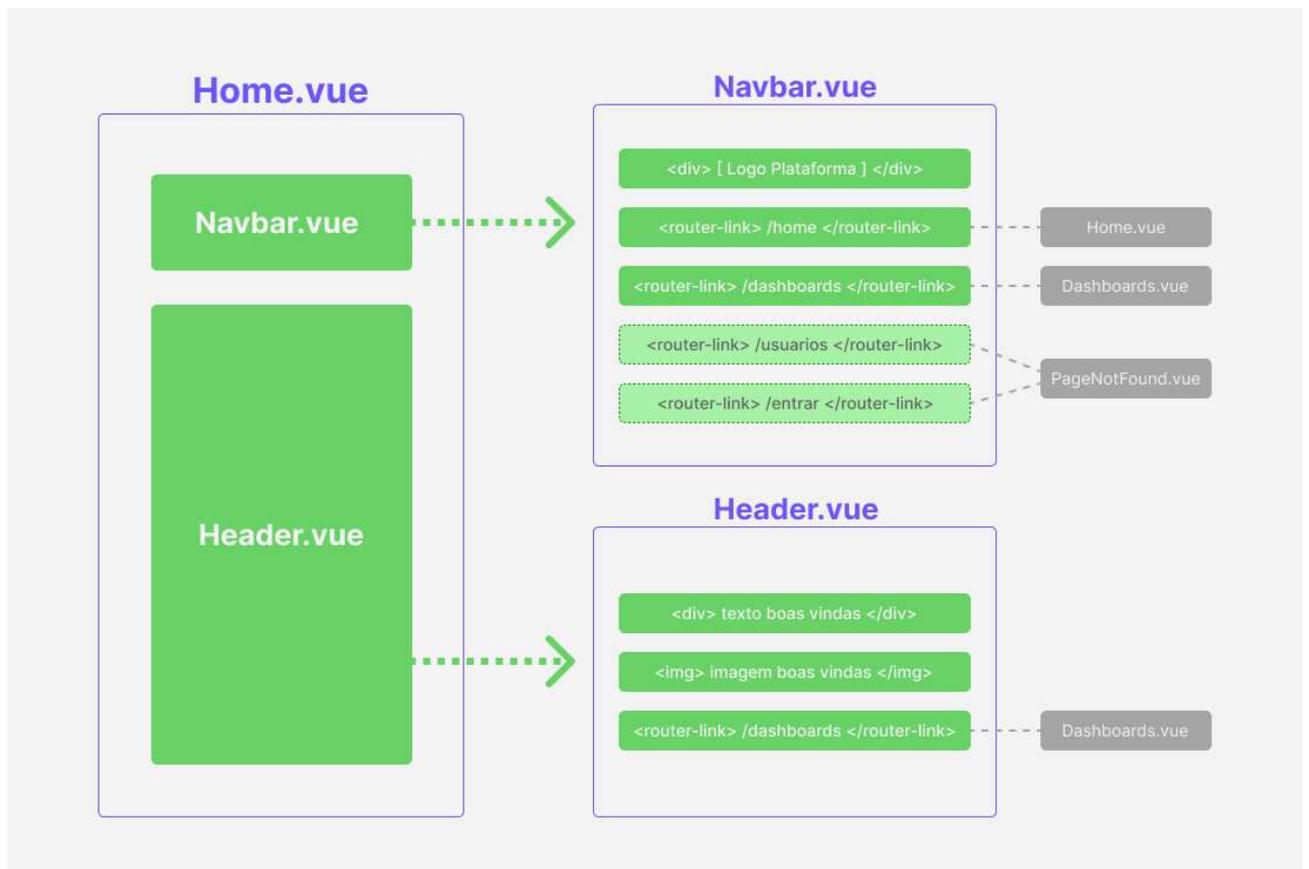
```

Fonte: Elaborado pelo autor.

Desta forma, tendo em mente as seguintes considerações, é possível montar um diagrama simplificado da estrutura do componente **Home.vue** (Figura 28):

- O componente Home.vue é formado pelos componentes Navbar.vue e Header.vue.
- O componente Navbar.vue possui links de navegação para as páginas “Início” (referente ao componente Home.vue), “Painéis” (Dashboards.vue), “Usuários” (para o escopo desse projeto, a página usuários não foi desenvolvida, então sua rota vai levar ao componente PageNotFound.vue) e “Entrar” (mesmo caso de “usuários”).
- O componente Header.vue também possui um link de navegação para a página Painéis (Dashboards.vue) localizado no botão embaixo do texto de boas-vindas.

Figura 28 – Diagrama da estrutura do componente Home.vue



Fonte: Elaborado pelo autor.

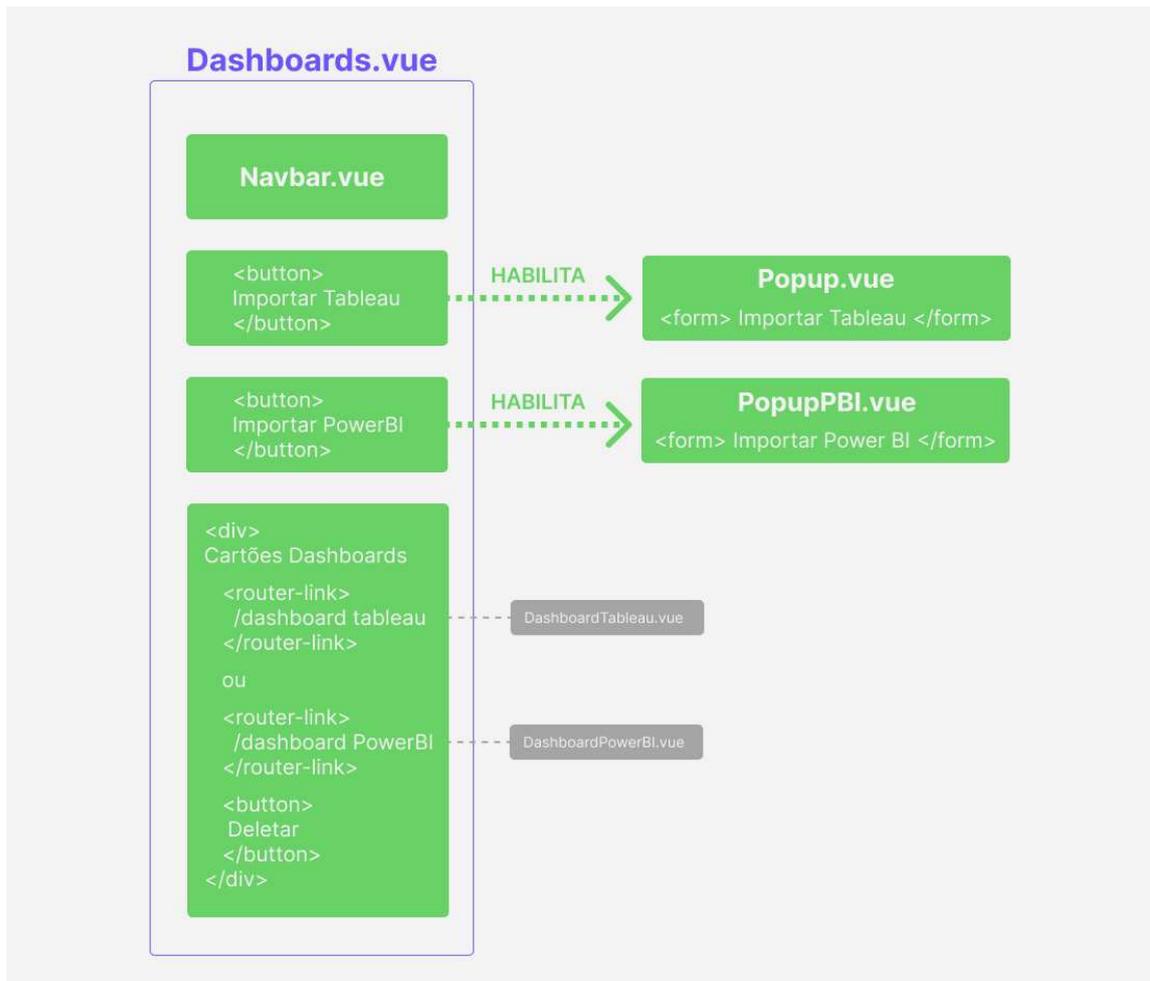
O componente **Dashboards.vue**, assim como o **Home.vue**, possui o componente **Navbar.vue**, porém, o resto do seu conteúdo é diferente. Este componente possui os botões para “Importar Dashboard Tableau” e “Importar Dashboard Power BI”, estes botões habilitam os componentes **Popup.vue** e **PopupPBI.vue**, respectivamente. Estes componentes de pop-up

são responsáveis por compor o po-pup de formulário para preencher as informações para importação dos dashboards.

Abaixo dos botões de importação, existe a galeria de cartões com informações dos dashboards já importados para a plataforma. Estes cartões possuem dois botões:

- O botão “Deletar”, que deleta do banco de dados o dashboard referente àquele cartão.
- E o botão “Explorar”, que possui link para a rota /dashboardTableau (referente ao componente DashboardTableau.vue) se o dashboard do cartão possuir o campo “tipo” igual a “Tableau”, ou, caso o valor de “tipo” for “Power BI”, o link aponta para a rota /dashboardPowerBI (referente ao componente DashboardPowerBI.vue). Na figura 29 é mostrado um diagrama simplificado da estrutura do componente **Dashboards.vue**.

Figura 29 – Diagrama da estrutura do componente Dashboards.vue



Fonte: Elaborado pelo autor.

Além disso, alguns detalhes importantes sobre o funcionamento do componente Dashboards.vue podem ser pontuados:

- Os componentes Popup.vue e PopupPBI.vue são habilitados na página por meio de uma cláusula “v-if”, que possibilita que a renderização de um elemento na página seja condicional. No caso dos pop-ups, eles aparecerão apenas se a variável “popupTriggers.buttonTrigger” (para o pop-up do Tableau) e “popupTriggers.buttonTriggerPBI” (para o pop-up do Power BI) for verdadeira. Como mostra a linha 95 da Figura 30.
- Como mostra a Figura 31, as variáveis para acionamento dos pop-ups são alteradas com a funções TogglePopup() e TogglePopupPBI(). Essas funções são acionadas quando os botões de importar dashboards são clicados.
- Os botões de “Explorar” presentes nos cartões de dashboards já importados também utilizam da cláusula “v-if” para mudar a rota de acordo com o tipo de dashboard do cartão, como foi explicado anteriormente. (linhas 76 e 79 da Figura 32).
- O botão “Deletar” utiliza a função deleteDash() que é importada do arquivo Firebase.js.
- Durante a inicialização do ciclo de vida do componente Dashboards.vue, a função useLoadDashes() (linha 214 da Figura 33) é chamada para carregar as informações fornecidas pelo back-end e preencher as informações dos cartões de dashboards importados.

Figura 30 – Código para abrir ou fechar pop-ups no componente Dashboards.vue

```

94
95     <Popup v-if="popupTriggers.buttonTrigger" :TogglePopup="() => TogglePopup('buttonTrigger')">
96
97     <p class="text-black text-2xl lg:text-4xl font-medium lg:font-semibold lg:leading-normal leading-snug">
98     Importar do Tableau
99     </p>
100

```

Fonte: Elaborado pelo autor.

Figura 31 – Funções para abrir ou fechar pop-ups no componente Dashboards.vue

```

Dashboards.vue X
src > components > Dashboards.vue > {} script > [🔍] default > [📄] setup
220 //Popup Tableau-----
221 const popupTriggers = ref({
222   buttonTrigger: false,
223 });
224
225 const TogglePopup = (trigger) => {
226   popupTriggers.value[trigger] = !popupTriggers.value[trigger]
227   console.log("Toggled")
228 }
229 //Popup Tableau-----
230
231 //PopupPBI-----
232 const popupTriggersPBI = ref({
233   buttonTriggerPBI: false,
234 });
235
236 const TogglePopupPBI = (trigger) => {
237   popupTriggersPBI.value[trigger] = !popupTriggersPBI.value[trigger]
238   console.log("ToggledPBI")
239 }
240 //PopupPBI-----
241

```

Fonte: Elaborado pelo autor.

Figura 32 – Código para criar botões “Explorar” no componente Dashboards.vue

```

Dashboards.vue ●
src > components > Dashboards.vue > ...
75 <div class="footer">
76   <router-link :to="`/dashTableau`" v-if="tipo === 'Tableau'">
77     Explorar
78   </router-link>
79   <router-link :to="`/dashPowerBI`" v-if="tipo === 'PowerBI'">
80     Explorar
81   </router-link>
82   <span class="mai-chevron-forward text-sm"></span>
83
84   <button style="margin-left: 20px; color: #E74848;" @click="deleteDash(id)">
85     Deletar
86   </button>
87   <span class="mai-close text-sm"></span>
88 </div>
89

```

Fonte: Elaborado pelo autor.

Figura 33 – Código para retornar os dashboards do banco de dados

```

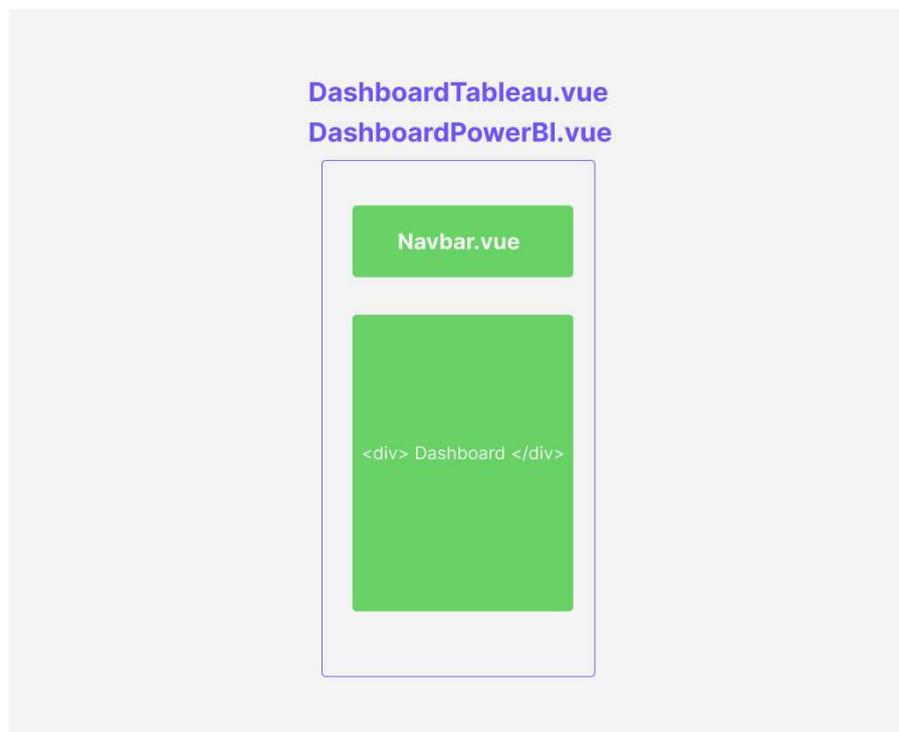
▼ Dashboards.vue ●
src > components > ▼ Dashboards.vue > {} script > [🔗] default > [🔗] setup
209
210 export default (await import('vue')).defineComponent({
211   name: "Dashboards",
212
213   setup() {
214     const dashes = useLoadDashes()
215
216

```

Fonte: Elaborado pelo autor.

Por fim, os componentes **DashboardTableau.vue** e **DashboardPowerBI.vue** possuem a mesma estrutura, onde apresentam o componente **Navbar.vue** no topo da página, e embaixo encontra-se o dashboard propriamente dito, o único aspecto que irá diferenciar esses dois componentes é a forma como os dashboards vão ser incorporados na página, pois isto varia de acordo com a plataforma de origem que o dashboard estava hospedado (Tableau ou Power BI), esse assunto será mais aprofundado na seção 4.2.3.7. Portanto, este é um diagrama simplificado da estrutura dos dois componentes (Figura 34):

Figura 34 – Diagrama da estrutura dos componentes **DashboardTableau.vue** e **DashboardPowerBI.vue**



Fonte: Elaborado pelo autor.

4.2.3.6 Importação de dashboards

No componente **Dashboards.vue**, as informações cadastrados no formulário de importação de dashboards são passadas para o banco de dados através das funções `onSubmitTableau()` e `onSubmitPowerBI()`, dependendo do tipo de dashboard que foi importado. (Figura 35).

Figura 35 – Código das funções `onSubmitTableau` e `onSubmitPowerBI`

```

src > components > Dashboards.vue > {} script > default > setup
241
242 //Form Tableau-----
243 const onSubmitTableau = async () => {
244   await createDash({ ...form })
245   form.dashboard = ''
246   form.titulo = ''
247   form.descricao = ''
248
249   TogglePopup('buttonTrigger')
250 }
251 //Form Tableau-----
252
253 //FormPBI-----
254 const onSubmitPBI = async () => {
255   await createDash({ ...formPBI })
256   formPBI.dashboard = ''
257   formPBI.titulo = ''
258   formPBI.descricao = ''
259
260   TogglePopupPBI('buttonTriggerPBI')
261 }
262 //FormPBI-----
263

```

Fonte: Elaborado pelo autor.

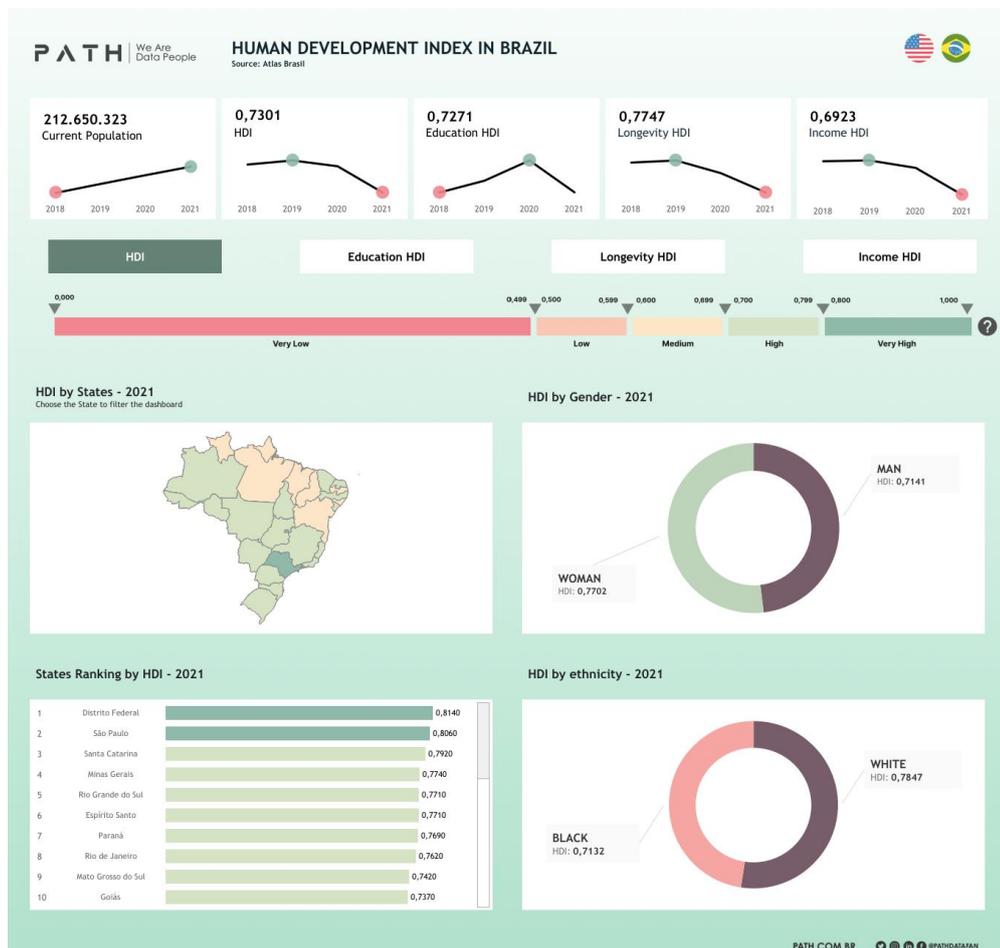
Estas funções são chamadas após o botão de “Importar” do formulário ser clicado, por isso, ao final destas funções, as funções de `TogglePopup()` são chamadas novamente para fechar os pop-ups.

4.2.3.7 Incorporando dashboards na página

Dado ao tempo de desenvolvimento, o escopo foi reduzido para entregar apenas um dashboard do tipo Tableau e um do tipo PowerBI, portanto as escolhas de importação de dashboard foram limitadas a uma para cada tipo de dashboard que pode ser importado.

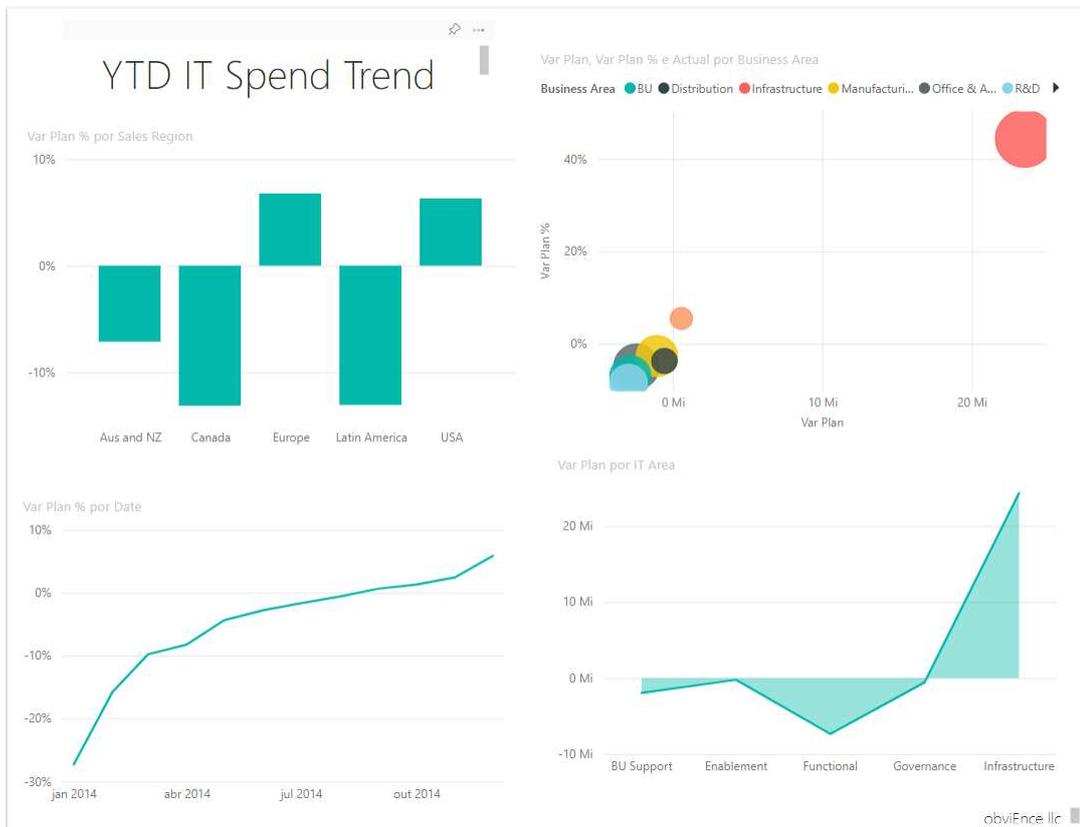
- Para o dashboard do tipo Tableau, foi escolhido o painel **“Human Development Index in Brazil”** (Figura 36) que apresenta sobre o IDH no Brasil ao longo dos anos. O mesmo está disponível no repositório público da Tableau e pode ser acessado pelo link: [Human Development Index in Brazil](#).
- Para o dashboard do tipo PowerBI, foi escolhido o painel **“YTD IT Spend Trend Analysis”** (Figura 37) que apresenta sobre a tendência de gastos da equipe de TI de uma empresa ao longo dos anos. O mesmo está disponível no repositório do Power Bi Online e pode ser acessado pelo link: [YTD IT Spend Trend Analysis](#).

Figura 36 – Painel: Human Development Index in Brazil



Fonte: Elaborado pelo autor.

Figura 37 – Painel: YTD IT Spend Trend Analysis



Fonte: Elaborado pelo autor.

O método de incorporar dashboards na página varia de acordo com a plataforma de origem em que o dashboard estava hospedado. Para a página de dashboard importados do Tableau (componente DashTableau.vue), a inserção ocorre por meio da tag `<tableau-viz id="tableauViz"></tableau-viz>`. Por esta tag, é passado o link do dashboard escolhido pelo atributo "src" (Figura 38).

Figura 38 – HTML tag tableau-viz

```

DashTableau.vue X
src > components > DashTableau.vue > ...
1 <template>
2 <Navbar />
3 <div class="page-banner home-banner" style="background-color: #0B1219">
4 <div class="row align-items-center" style="background-color: #0B1219; padding-top: 60px; padding-left: 150px; ">
5 <div class="col" style="background-color: #0B1219; ref="tableau">
6 <tableau-viz id="tableauViz"
7   src= "https://public.tableau.com/views/HumanDevelopmentIndexinBrazil/DashboardIDHBrasilInglis?:language=pt-BR&:display_count=n&:origin=viz_share_link"
8   height="1400px" width="2000px" toolbar="bottom" hide-tabs>
9 </tableau-viz>
10 </div>
11 </div>
12 </div>
13 </template>
14

```

Fonte: Elaborado pelo autor.

Para a página de dashboard importados do Power BI (componente DashPowerBI.vue), a inserção ocorre por meio da tag `<iframe></iframe>`. Por esta tag, é passado o link do dashboard escolhido pelo atributo “src” (Figura 39).

Figura 39 – HTML tag iframe



```

1 <template>
2 <navbar />
3
4 <div class="page-banner home-banner" style="background-color: #081219; overflow: hidden;">
5 <div class="row align-items-center" style="padding-top: 50px; padding-left: 150px; ">
6 <div class="col" style="background-color: #081219; ref="Power BI">
7 <iframe title="IT Spend Analysis Sample" width="1350" height="750"
8 src="https://app.powerbi.com/reportEmbed?reportId=52e5ff5b-0c70-447a-a088-4fc7bf3bbc94&autoAuth=true&ctid=c84da17b-b660-4390-814f-052a50b5436e"
9 frameborder="0" allowFullScreen="true"></iframe>
10 </div>
11 </div>
12 </template>

```

Fonte: Elaborado pelo autor.

4.3 Reflexão sobre o desenvolvimento

O objetivo do trabalho foi desenvolver uma plataforma para unificar dashboards criados e publicados em diferentes ferramentas de Business Intelligence (BI). Para alcançar esse objetivo, foram adotadas três ferramentas principais: Vue.js para a codificação, Figma para a prototipação da interface e Firebase para o banco de dados.

Durante o processo de desenvolvimento, as ferramentas escolhidas demonstraram-se eficientes e adequadas para as necessidades do projeto. O uso do Vue.js como framework de codificação foi fundamental para a construção da plataforma. Sua arquitetura baseada em componentes permitiu uma organização estruturada do código, facilitando a manutenção e escalabilidade do projeto. Além disso, a ampla documentação e a comunidade ativa em torno do Vue.js proporcionaram suporte valioso durante o desenvolvimento.

A utilização do Figma como ferramenta de prototipação de interface permitiu uma visualização clara e detalhada do design da plataforma. Com o Figma, foi possível criar protótipos interativos, explorar diferentes layouts e validar a usabilidade da interface. A colaboração em tempo real oferecida pela ferramenta também contribuiu para a agilidade e eficiência do processo de design.

O Firebase foi escolhido como banco de dados para a plataforma, fornecendo uma solução eficaz e escalável para armazenar e gerenciar os dados. Sua integração com o Vue.js facilitou a manipulação e sincronização dos dados em tempo real, permitindo uma experiência mais fluida para os usuários da plataforma.

Além disso, a metodologia proposta para o desenvolvimento do trabalho foi aplicada com sucesso. As etapas definidas como: (i) Levantar requisitos, (ii) Projetar e desenvolver e (iii) Avaliar foram satisfatórias para a criação desta primeira versão da plataforma. Os passos do design thinking, como entendimento do problema, ideação, prototipação e teste, foram seguidos ao longo do processo. Essa abordagem centrada no usuário contribuiu para a criação de uma plataforma que atende às necessidades dos usuários e oferece uma experiência intuitiva.

Por fim, é importante destacar algumas limitações encontradas durante o desenvolvimento do protótipo da plataforma. O tempo disponível para a realização do trabalho foi um fator determinante na definição do escopo, impossibilitando a inclusão de todos os recursos desejados.

Algumas possíveis melhorias para a plataforma incluem a implementação de recursos avançados de visualização de dados, a integração com mais ferramentas de BI e a otimização da experiência do usuário.

5 RESULTADOS

Neste capítulo, são apresentados os dois métodos definidos para avaliar o resultado deste trabalho, bem como, apresentar e discutir estes resultados. Além disso, será apresentado o resultado final do desenvolvimento da plataforma, levando em consideração os requisitos funcionais estabelecidos no capítulo referente à metodologia.

5.1 Métodos

Para a etapa de avaliação da usabilidade da plataforma desenvolvida, foram utilizados dois métodos:

- Teste assistido de usabilidade.
- Questionário System Usability Scale (SUS).

O teste tem o objetivo de coletar dados numéricos e mensuráveis sobre o desempenho e a eficácia da plataforma, enquanto o questionário SUS avalia a usabilidade do sistema por meio de uma escala padronizada. Essas técnicas combinadas permitem uma avaliação abrangente e objetiva da plataforma, fornecendo insights valiosos sobre sua usabilidade e efetividade para os usuários. (BOUCINHA; TAROUCO, 2013).

Os participantes da avaliação de usabilidade da plataforma consistem em um grupo de 06 funcionários de uma mesma empresa de Tecnologia. O perfil dos participantes varia em relação à área de atuação na empresa e idade. mas, no geral, todo este grupo representa potenciais usuários da plataforma desenvolvida.

O questionário SUS foi aplicado através da plataforma Google Forms. Esta plataforma, além de ser gratuita, possibilita realizar o processo de coleta de dados de forma flexível, permitindo que os voluntários respondam aos formulários de maneira independente e online, simultaneamente a outros usuários. A plataforma também facilita o processamento dos dados, organizando-os automaticamente e gerando gráficos com métricas relevantes com base nos resultados obtidos.

É fundamental destacar a importância de um Termo de Consentimento Livre e Esclarecido (TCLE) que deve ser apresentado aos voluntários no início da pesquisa. Esse documento garante que os participantes tenham pleno conhecimento e compreensão dos propósitos, procedimentos e possíveis riscos da pesquisa, além de resguardar os direitos da universidade em relação à utilização dos dados coletados. O conteúdo do termo apresentado pode ser visto no Apêndice A.

5.1.1 Teste assistido de usabilidade

No teste assistido, que foi o primeiro teste realizado, os participantes foram monitorados enquanto tentavam completar 04 tarefas com o protótipo desenvolvido da interface. As tarefas elencadas para o teste foram:

1. Navegar da página Home até a página Dashboards.
2. Importar um novo dashboard Tableau.
3. Abrir o dashboard importado.
4. Deletar o dashboard importado.

Em seguida, o resultado de cada participante foi documentado. Este teste de usabilidade apresenta os resultados em porcentagens que são divididas em 3 grupo:

- **Direct Success** (Sucesso direto): Usuários que completaram a tarefa seguindo o fluxo esperado.
- **Indirect Success** (Sucesso indireto): Usuários que completaram a tarefa sem seguir o fluxo esperado.
- **Give-up** (Desistir): Usuários que desistiram ou pularam para próxima tarefa.

5.1.2 Questionário SUS

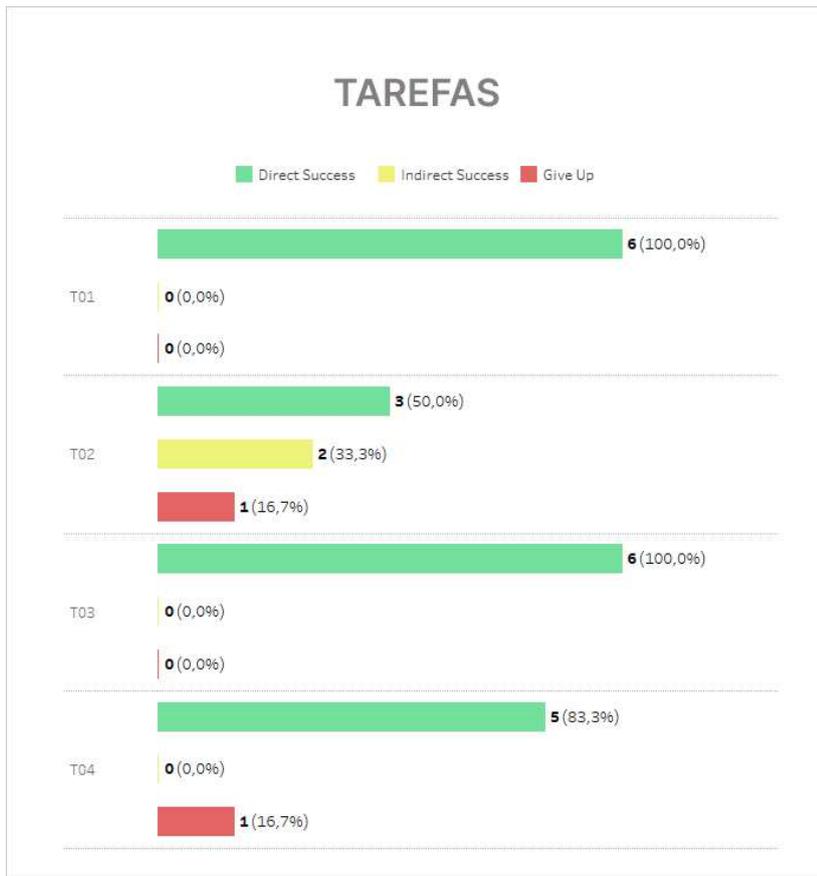
O questionário SUS e seus métodos de avaliação foram aprofundados no capítulo de Referencial Teórico, na seção **2.10**.

5.2 Apresentação dos Resultados

A seguir estão listados os resultados das tarefas do teste de usabilidade (Figura 40):

- **Tarefa 1:** 100% de sucesso direto, 0% de sucesso indireto e 0% de desistência.
- **Tarefa 2:** 50% de sucesso direto, 33,3% de sucesso indireto e 16,7% de desistência.
- **Tarefa 3:** 100% de sucesso direto, 0% de sucesso indireto e 0% de desistência.
- **Tarefa 4:** 83,3% de sucesso direto, 0% de sucesso indireto e 16,7% de desistência.

Figura 40 – Resultados do teste de usabilidade



Fonte: Elaborado pelo autor.

A seguir, na *Tabela 6*, estão os resultados do questionário de usabilidade seguindo o modelo SUS, que foi avaliado de acordo com a explicação encontrada na seção **2.10** deste documento. Seguindo as diretrizes de avaliação, os 6 participantes do questionário tiveram as seguintes pontuações (observação: “P01” significa “Pergunta 01”, e assim por diante. Os números das colunas “Px” representam a resposta do participante àquela pergunta):

Tabela 6 – Resultado do Questionário SUS da plataforma VIZAGE

	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	Pontuação do participante
Participante 01	5	1	5	1	5	1	5	1	5	1	100
Participante 02	5	1	5	1	5	1	5	1	5	1	100
Participante 03	4	1	4	2	4	1	4	2	4	2	80
Participante 04	5	1	4	3	4	1	5	2	5	1	87,5
Participante 05	5	1	5	1	5	1	5	1	5	1	100
Participante 06	2	4	1	5	3	4	3	5	1	2	25

Fonte: Elaborado pelo autor.

A análise destes resultados será aprofundada posteriormente na seção 5.5 deste documento.

5.3 Atendimento dos requisitos funcionais: Front-end

A seguir estão apresentados os resultados finais do desenvolvimento e como foram atendidos os requisitos definidos anteriormente, na seção 4.2.1 deste documento:

RF01 Menu de navegação (Figura 41).

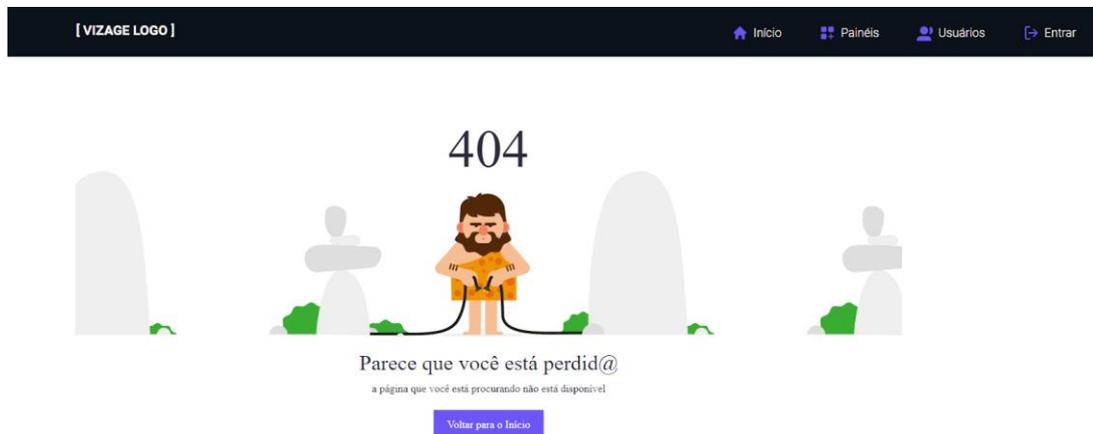
Figura 41 – Menu de navegação



Fonte: Elaborado pelo autor.

RF02 Conteúdo “Em breve” para páginas fora do escopo (Figura 42).

Figura 42 – Página “Em breve”



Fonte: Elaborado pelo autor.

RF03 - Página “Início” com texto de boas-vindas, uma imagem e um botão para nos levar à página de dashboards. (Figura 43).

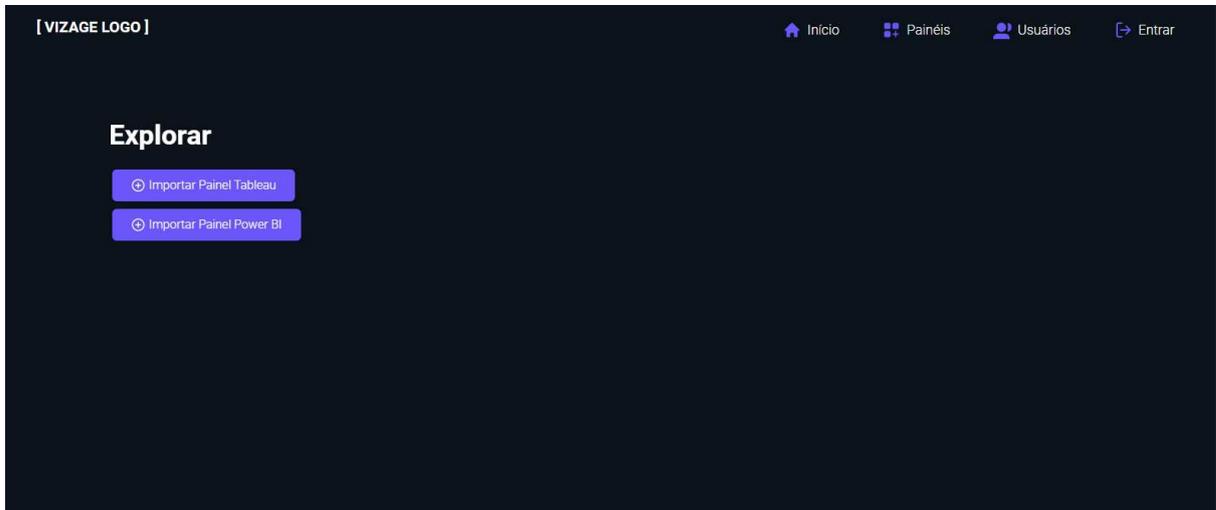
Figura 43 – Página Início



Fonte: Elaborado pelo autor.

RF04 - Página “Painéis” com botões para importar os dashboards (Figura 44).

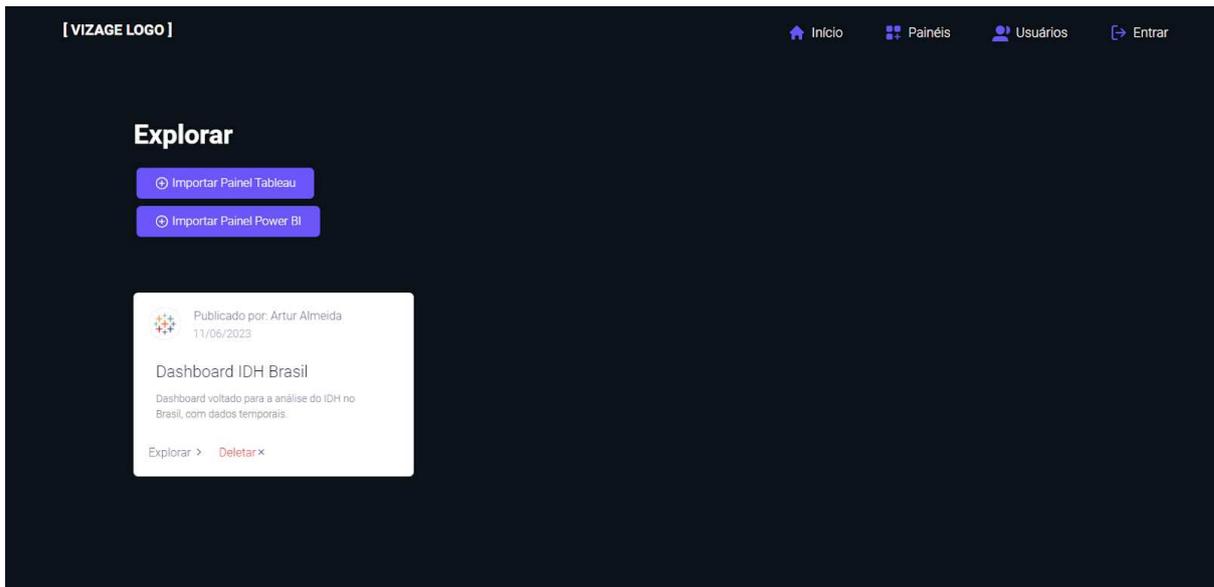
Figura 44 – Página Painéis



Fonte: Elaborado pelo autor.

RF05 - Galeria de dashboards já importados, cada dashboard já presente na galeria deve conter uma seção com os seguintes elementos: Ícone de qual plataforma pertence, título, autor, data da publicação, um botão para ver o dashboard e um botão para deletá-lo (Figura 45).

Figura 45 – Galeria de dashboards na página Painéis

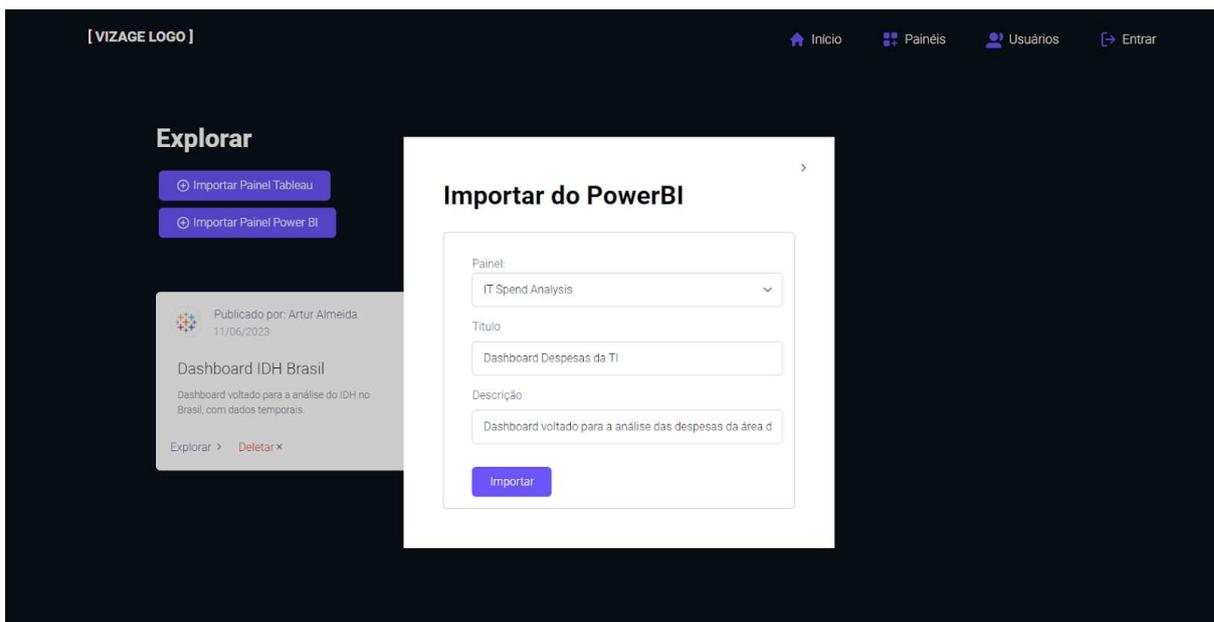


Fonte: Elaborado pelo autor.

RF06 - Ao clicar em adicionar um novo dashboard, deve abrir um pop-up com um formulário para ser populado com as informações do dashboard que será incluído na galeria (Figura 46).

RF07 - Ao final do formulário, deve existir um botão de confirmação, que, ao clicado, deverá validar o formulário e criar o dashboard com as informações preenchidas (Figura 46).

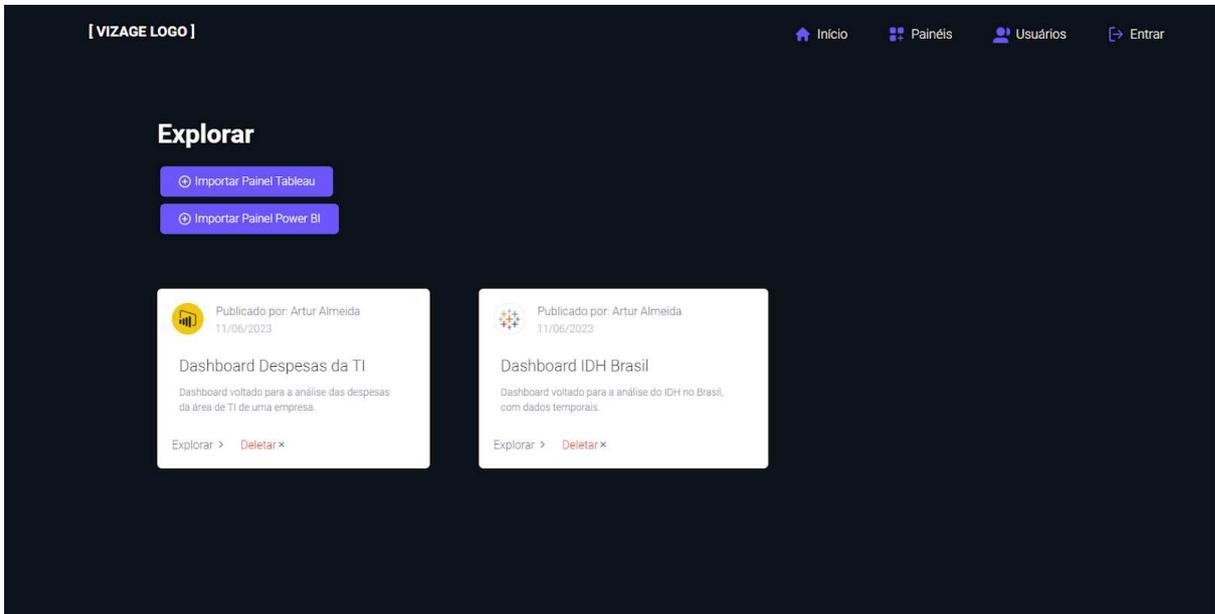
Figura 46 – Formulário para importação de dashboard



Fonte: Elaborado pelo autor.

RF08 - Ao ser criado um novo dashboard, deve aparecer na galeria a seção com suas informações (Figura 47).

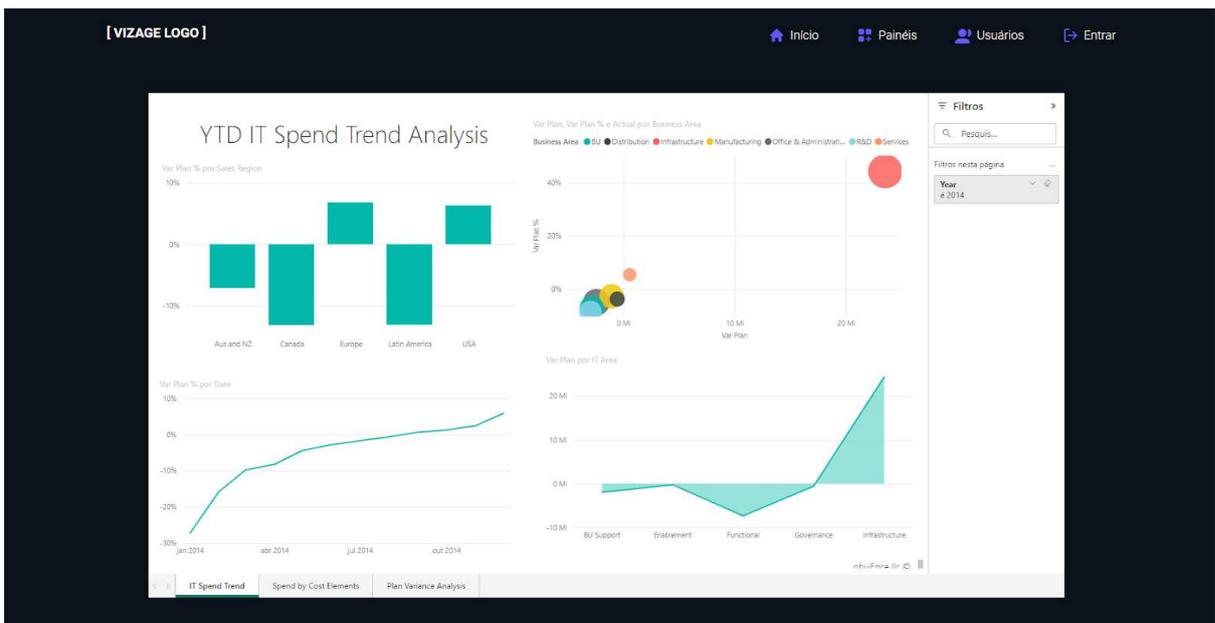
Figura 47 – Novo dashboard importado para a galeria



Fonte: Elaborado pelo autor.

RF09 I - Visualizar dashboard importado (Figura 48).

Figura 48 – Visualizando dashboard Power BI recém importado



Fonte: Elaborado pelo autor.

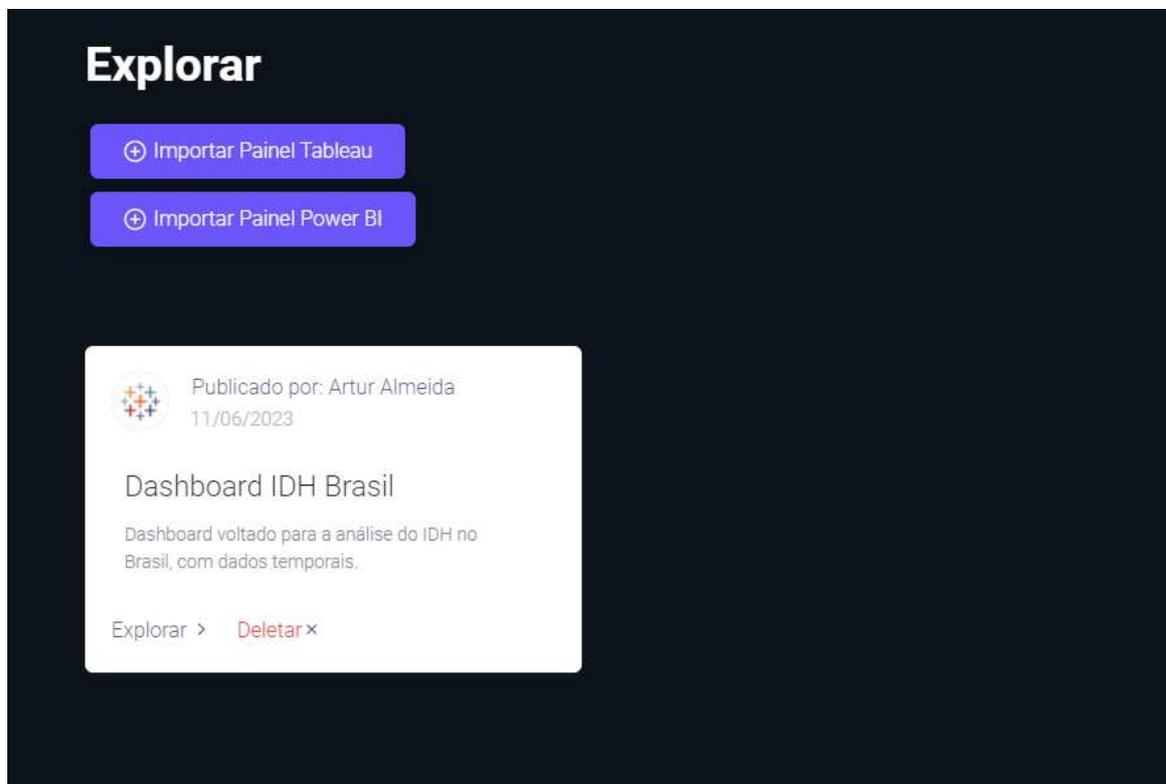
RF09 II - Deletar dashboard importado (Figuras 49 e 50).

Figura 49 – Destaque no botão de deletar dashboard



Fonte: Elaborado pelo autor.

Figura 50 – Página painéis atualizada sem o dashboard que foi deletado



Fonte: Elaborado pelo autor.

5.4 Atendimento dos requisitos funcionais: Back-end

RF01 Realizar a persistência e coleta de dados de um banco externo não relacional (FireBase).
 [Observação]: Nessa versão do projeto, o escopo foi pensado de modo reduzido, portanto, a aplicação deve ser estruturada para armazenar e manipular apenas dashboards do tipo Tableau e Power BI. (Concluído)

RF02 Definição de um modelo de dados para os dashboards. (Concluído)

RF03 Realização das operações C.R.U.D para dashboards importados.

- CREATE - Criar um dashboard. (Concluído)
- GET - Retornar um dashboard específico do banco. (Concluído)
- UPDATE - Alterar um dashboard específico do banco. (Concluído)
- DELETE - Deletar dashboard específico do banco. (Concluído)
- LOAD - Retornar todos os dashboards do banco. (Concluído)

5.5 Discussão

Nesta seção, discutiremos os resultados obtidos a partir do teste e questionário de usabilidade. O objetivo dessas etapas foi avaliar a eficiência e a facilidade de uso da plataforma, bem como identificar possíveis áreas de melhoria em termos de usabilidade.

O teste de usabilidade consistiu em quatro atividades que foram realizadas pelos participantes. Na primeira e terceira atividades, observamos um desempenho muito satisfatório, com 100% de acerto direto por parte dos participantes. Isso indica que essas tarefas foram facilmente compreendidas e executadas pelos usuários, demonstrando a efetividade da interface e da interação proposta.

No entanto, na segunda atividade, encontramos um índice de acerto direto de apenas 50%. No processo de realização desta atividade, a dificuldade mais comum foi acessar o formulário e completá-lo. Isso sugere que essa tarefa pode apresentar desafios para os usuários, indicando a necessidade de uma melhor clareza nas instruções ou de possíveis ajustes na interface. A análise desses resultados aponta para a importância de realizar uma avaliação mais aprofundada dessa atividade específica, a fim de identificar as possíveis dificuldades enfrentadas pelos usuários e propor melhorias correspondentes.

Já na quarta atividade, observamos um índice de acerto direto de 83,3%. Embora seja um resultado positivo, indica que ainda há margem para aprimoramentos na usabilidade dessa tarefa.

Além disso, o questionário de usabilidade SUS nos deu alguns pontos importantes para serem analisados. Porém, primeiro precisamos utilizar os dados apresentados na *Tabela 6*, mostrada na seção 5.2, para calcular a pontuação SUS total da plataforma desenvolvida:

- Soma dos scores dos participantes = 492,5
- Score SUS Total da plataforma VIZAGE = $492,5 / 6 = 82$

Como relatado na seção a *Tabela 3*, apresentada na seção 2.10, segundo Bangor, Kortum e Miller (2009), o **Score SUS de 82** representa a categoria “Bom”. Isto indica que, apesar de ser um protótipo, a usabilidade da plataforma está aceitável, visto que pontuações acima de 70 se enquadram na categoria de resultados aceitáveis e pontuações entre os valores 80-90 seriam correspondentes à nota “B” do sistema de escala de notas americano.

Ademais, é importante ressaltar o *outlier* representado pelo participante 06 mostrado na tabela de resultados do questionário SUS (*Tabela 6*). Este participante teve um resultado que chamou atenção, visto que, além de ter o menor score SUS entre os participantes, ele também teve dificuldades para realizar as tarefas 2 e 4 do teste de usabilidade. Chegamos à conclusão de que o valor atípico observado no caso do participante 06 pode ser atribuído à diferença de perfil desse participante em relação aos demais. Primeiramente, destaca-se que o participante 06 pertence ao setor comercial da empresa, enquanto os demais participantes são do setor técnico. Essa diferença de perfil pode ter influenciado na forma como o participante interagiu e percebeu a plataforma durante o estudo.

Além disso, é relevante mencionar que o participante 06 faz parte de uma faixa etária mais avançada, entre 50 e 60 anos. Essa característica demográfica pode ter impactado sua familiaridade e adaptação aos aspectos tecnológicos da plataforma, resultando em respostas diferentes em relação aos demais participantes.

Portanto, com base nos resultados do teste e do questionário de usabilidade, podemos concluir que a plataforma apresenta um bom desempenho geral em termos de usabilidade. No entanto, as atividades com índices de acerto direto mais baixos e os resultados obtidos com o participante 06, que respondeu de forma negativa ao questionário, destacam a necessidade de realizar ajustes e aprimoramentos. É necessário aprimorar a visibilidade das funções principais e a usabilidade da plataforma, levando em consideração as características específicas dos usuários. Isso significa que, ao realizar melhorias na plataforma, é crucial

considerar as particularidades e as demandas dos vários possíveis usuários da plataforma.

Para melhorar a visibilidade das funções principais, é recomendado otimizar a organização dos elementos da interface, tornando-os mais intuitivos e de fácil acesso. Além disso, é importante considerar a clareza e a simplicidade das instruções e orientações presentes na plataforma, garantindo que os usuários de diferentes perfis possam compreender e utilizar as funcionalidades de forma eficaz.

No que diz respeito à usabilidade, é fundamental simplificar os processos e fluxos de interação, reduzindo a complexidade e fornecendo feedback adequado aos usuários. A usabilidade deve ser pensada levando em conta as habilidades e a familiaridade tecnológica dos usuários.

Essas descobertas são importantes para direcionar futuras melhorias na interface e na interação da plataforma, visando proporcionar uma experiência mais intuitiva e eficiente para todos.

6 CONCLUSÕES E O FUTURO DO PROJETO

6.1 Considerações finais

Este trabalho levantou a discussão sobre a facilitação da publicação e compartilhamento de painéis de BI criados em diferentes plataformas. O mesmo documentou o processo de elaboração e desenvolvimento da primeira versão da plataforma Vizage, que tem como objetivo ser uma opção para solucionar o problema da integração de visualizações, explorando o potencial das tecnologias apresentadas (Vue e Firebase).

O percurso do desenvolvimento deste trabalho abrangeu diferentes etapas essenciais. Inicialmente, foi realizado o levantamento de requisitos, permitindo uma compreensão aprofundada das necessidades dos usuários e a definição dos principais objetivos da plataforma. Em seguida, aplicou-se a abordagem do Design Thinking para orientar a concepção e o design da interface, com foco na usabilidade.

A seguir, foi realizado a prototipação e desenvolvimento da plataforma, com a adoção do software Figma para a prototipação e do framework Vue.js para codificação de todo o ambiente Front-end.

Os resultados dos testes com usuários reais e do questionário SUS revelaram a efetividade da plataforma em questões de usabilidade, apesar de que, como vimos anteriormente, ainda é necessária a implementação de melhorias para atender usuários com menos familiaridade técnica. A plataforma, em sua essência, busca preencher uma lacuna no mercado, ao proporcionar uma solução acessível para a análise de dados, com foco em atender a um público diverso.

Em conclusão, a motivação por trás da criação desta solução gratuita vai além da simples disponibilidade de recursos acessíveis. Almeja-se democratizar o acesso à análise de dados e proporcionar uma ferramenta poderosa e de fácil utilização para um público amplo, inclusive para pequenas empresas e empreendedores que, muitas vezes, não possuem recursos para investir em soluções pagas.

6.2 O Futuro da plataforma

É importante destacar que o escopo atendido foi limitado às necessidades do prazo de construção desse trabalho, uma vez que o objetivo era montar um protótipo. Um ponto central para a continuidade do projeto nos seus próximos passos, é a exploração de

funcionalidades que foram pensadas, mas não foram desenvolvidas nessa primeira versão da plataforma, como por exemplo:

- Importação de dashboards de outras ferramentas de BI, como ThoughtSpot e Qlik.
- Implementação de uma aba de favoritos.
- Implementação de funcionalidades para editar informações dos painéis publicados.
- Utilização de APIs para integração com as plataformas de BI, possibilitando a listagem real dos painéis publicados em seus repositórios.
- Implementação de Login e usuários.
- Implementação da criação de grupos de usuários.
- Implementação da visualização/ocultamento de dashboards para usuários ou grupos de usuários selecionados.
- Implementação de uma barra de pesquisa na página de dashboards, a fim de facilitar a busca por dashboards específicos.
- Implementação de uma funcionalidade de compartilhamento dos painéis.

REFERÊNCIAS

- DILIGENT. **The Benefits of Centralized Data: A Single Source of Information for Your Business Entities**. Disponível em: <https://www.diligent.com/insights/data-management/the-benefits-of-centralized-data-a-single-source-of-information-for-your-business-entities/>. 2018. Acesso em: 10 de abril de 2023.
- HARMATIUK DA SILVA, Lucas José; COELHO, Taiane Ritta. **O uso das ferramentas de BI sob o ponto de vista dos profissionais de TI e de Negócio**. 2020.
- PRIMAK, Fábio Vinícius. **Decisões com bi (business intelligence)**. Fabio Vinicius Primak, 2008.
- CORTEX INTELLIGENCE. **Estudo dos Concorrentes**. Blog da Cortex Intelligence, [S.l.], 2023. Disponível em: <https://www.cortex-intelligence.com/blog/inteligencia-de-mercado/estudo-dos-concorrentes>. Acesso em: 17 maio 2023.
- CURATOR. **All Your Analytics In One Place**. 2023. Disponível em: <https://curator.interworks.com/>. Acesso em: 17 de maio de 2023.
- LOOMESOFTWARE. **Connecting people to the data they need**. 2023. Disponível em: <https://www.loomesoftware.com/index>. Acesso em: 17 de maio de 2023.
- THE REPORTING HUB. **White Label Business Intelligence Platform For Power BI**. 2023. Disponível em: <https://thereportinghub.com/>. Acesso em: 17 de maio de 2023.
- SOUZA, W. C. **Construtor de Sistemas Web**. 2018. Monografia de Graduação. Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Instituto Federal de Santa Catarina, Gaspar, Brasil.
- FELIZARDO, A. **O que é Vue.js**. 2018. Disponível em: <http://www.andrefelizardo.com.br/blog/o-que-e-vue-js>. Acessado em: 17 de maio de 2023.
- VUEJS. **O Framework JavaScript Progressivo**. 2023. Disponível em: <https://vuejsbr-docs-next.netlify.app/>. Acessado em: 19 de maio de 2023.
- SILVA, Andréia Silva da. **Business intelligence**. 2017.
- LAPA, Joaquim Manuel Oliveira. **Estudo e Avaliação de Plataformas de Business Intelligence em contexto empresarial**. 2015. Tese de Doutorado. Instituto Politecnico do Porto (Portugal).

BROWN, Tim. **Design thinking**. Harvard business review, v. 86, n. 6, p. 84-92, 2008.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Addison-Wesley, 2011.

TABLEAU SOFTWARE. **Modern Analytics Workflow**. 2021. Disponível em: https://help.tableau.com/current/blueprint/en-us/bp_modern_analytics_workflow.htm. Acesso em: 17 de maio de 2023.

BROOKE, J. **Smart phone applications for people with brain injury**. United Kingdom: Agency for Clinical Innovation, 1986.

BANGOR, Aaron; KORTUM, Philip; MILLER, James. **Determining what individual SUS scores mean: Adding an adjective rating scale**. Journal of usability studies, v. 4, n. 3, p. 114-123, 2009.

NORMAN, Donald A. **Emotional design: Why we love (or hate) everyday things**. Civitas Books, 2004.

CRESWELL, John W.; CRESWELL, J. David. **Research design: Qualitative, quantitative, and mixed methods approaches**. Sage publications, 2017.

BOUCINHA, R. M.; TAROUCO, L. M. R. **Avaliação de ambiente virtual de aprendizagem com o uso do sus-system usability scale**. 2013. Disponível em: <<https://www.seer.ufrgs.br/renote/article/view/44479>>.

APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

Você está sendo convidado por Alysson Diniz dos Santos, professor da Universidade Federal do Ceará - Campus Pici, e por Artur de Freitas Almeida, aluno do curso de Sistemas e Mídias Digitais da Universidade Federal do Ceará - Campus Pici, como participante da pesquisa intitulada "Vizage: O processo de desenvolvimento de uma plataforma para centralizar a hospedagem de painéis de BI".

Você não deve participar contra a sua vontade. Leia atentamente as informações a seguir e faça qualquer pergunta que desejar, para que todos os procedimentos desta pesquisa sejam esclarecidos.

O objetivo desta pesquisa é utilizar uma ferramenta para a importação de painéis de BI publicados em diferentes plataformas, e avaliar sua usabilidade sob diversos aspectos.

Basicamente você será solicitado a realizar 4 tarefas em um protótipo da plataforma, e em seguida deverá preencher um questionário com 10 perguntas.

As respostas deverão ser baseadas em sua experiência na utilização da ferramenta.

(i) Eu declaro que é de livre e espontânea vontade que estou participando desta pesquisa.

(ii) Eu declaro que li cuidadosamente este Termo de Consentimento Livre e Esclarecido e que, após sua leitura, tive a oportunidade de fazer perguntas sobre o seu conteúdo, como também sobre a pesquisa, e recebi explicações que responderam por completo minhas dúvidas.