



**INSTITUTO FEDERAL DO PIAUÍ – IFPI
PRÓ-REITORIA DE ENSINO
DIRETORIA DE ENSINO
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

RELATÓRIO TÉCNICO DE SOFTWARE
ORÇAMETRO: TECNOLOGIA E ACESSIBILIDADE NA CRIAÇÃO DE ORÇAMENTOS

ÉZIO FEITOSA OLIVEIRA FONTES

AISLAN RAFAEL RODRIGUES DE SOUSA
ORIENTADOR

PICOS, PI
2025

ÉZIO FEITOSA OLIVEIRA FONTES

ORÇAMETRO: TECNOLOGIA E ACESSIBILIDADE NA CRIAÇÃO DE ORÇAMENTOS

Relatório Técnico de Software apresentado como exigência parcial para obtenção do diploma do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Picos.

Orientador: Prof. Me. Aislan Rafael Rodrigues de Sousa

Aprovado em: 08/04/2025

BANCA EXAMINADORA

Prof. Aislan Rafael Rodrigues de Sousa
Instituto Federal do Piauí

Prof. Jesiel Viana da Silva
Instituto Federal do Piauí

Prof. João Paulo Lima do Nascimento
Instituto Federal do Piauí

LISTA DE ILUSTRAÇÕES

Figura 1 - Diagrama de caso de uso	10
Figura 2 - Diagrama de arquitetura do software	12
Figura 3 - Diagrama de entidade relacionamento	13
Figura 4 - Diagrama de navegação e nível de usuário do sistema	14
Figura 5 - Interface de login do aplicativo	15
Figura 6 - Cadastro de usuário	15
Figura 7 - Contratação de plano	16
Figura 8 - Página de vendedores	17
Figura 9 - Edição de vendedores	17
Figura 10 - Exclusão de vendedores	18
Figura 11 - Cadastro de vendedores	18
Figura 12 - Listagem de serviços	19
Figura 13 - Criação de serviços	19
Figura 14 - Edição de serviços	20
Figura 15 - Exclusão de serviços	20
Figura 16 - Interface de produtos	21
Figura 17 - Cadastro de produtos	21
Figura 18 - Edição de produtos	22
Figura 19 - Exclusão de produtos	22
Figura 20 - Listagem de orçamentos	23
Figura 21 - Visualização, envio e validação de orçamento	23
Figura 22 - Criação de Orçamentos	24
Figura 23 - Exclusão de orçamento	24
Figura 24 - Dashboard de análise de dados	25
Figura 25 - Diagrama de implantação	27
Figura 26 - Teste SUS, gráfico da primeira pergunta	28
Figura 27 - Teste SUS, gráfico da segunda.....	29
Figura 28 - Teste SUS, gráfico da terceira pergunta	29
Figura 29 - Teste SUS, gráfico da quarta pergunta	30
Figura 30 - Teste SUS, gráfico da quinta pergunta	30
Figura 31 - Teste SUS, gráfico da sexta pergunta	31
Figura 32 - Teste SUS, gráfico da sétima pergunta	31
Figura 33 - Teste SUS, gráfico da oitava pergunta	32
Figura 34 - Teste SUS, gráfico da nona pergunta	32
Figura 35 - Teste SUS, gráfico da décima pergunta	33

RESUMO

Este relatório técnico apresenta o desenvolvimento do Orçametro, um *software* acessível para a geração de orçamentos, voltado tanto para pequenas e médias empresas quanto para profissionais autônomos. O projeto surgiu da necessidade de oferecer uma solução moderna e de baixo custo para aqueles que não dispõem de recursos para adquirir *softwares* robustos e caros. O desenvolvimento seguiu uma abordagem iterativa, baseada no levantamento de requisitos, modelagem do sistema e testes de usabilidade, garantindo uma interface intuitiva e funcionalidades essenciais. O Orçametro permite a criação ágil de orçamentos, o cadastro de serviços e produtos, a aplicação de descontos e a geração de documentos em PDF, que podem ser enviados diretamente por *e-mail* ou WhatsApp. Os resultados obtidos indicam que a ferramenta atende às necessidades do público-alvo, proporcionando eficiência e acessibilidade na gestão financeira. Conclui-se que o Orçametro contribui para a modernização de pequenos negócios e trabalhadores autônomos, oferecendo uma solução tecnológica acessível e eficaz.

Palavras-chave: Orçametro; *software* de orçamentos; pequenas empresas; autônomos; tecnologia acessível; produtos; serviços; orçamento; vendedores; cadastro; edição; exclusão; visualização.

SUMÁRIO

1.INTRODUÇÃO.....	7
1.1.OBJETIVOS.....	7
1.1.1.Geral.....	7
1.1.2.Específicos.....	8
2.TECNOLOGIAS ENVOLVIDAS.....	8
3.MODELAGEM DO PROJETO.....	9
3.1.LEVANTAMENTO DE REQUISITOS.....	9
3.2.DIAGRAMAS DE CASOS DE USO.....	11
3.4.ARQUITETURA DO SISTEMA.....	12
3.5.DIAGRAMAS DE ENTIDADES-RELACIONAMENTOS.....	13
3.6.INTERFACE.....	15
4.SOFTWARE.....	27
4.1.IMPLANTAÇÃO.....	27
4.2.TESTES.....	28
4.3 ACESSO AO	
CÓDIGO-FONTE.....	34
4.4 ACESSO À	
APLICAÇÃO.....	35
5.CONSIDERAÇÕES FINAIS.....	35
6.REFERÊNCIA.....	36

1. INTRODUÇÃO

A modernização da gestão empresarial é um desafio para pequenas e médias empresas, especialmente para aquelas que não possuem recursos para investir em *softwares* caros e robustos. Muitos desses negócios ainda realizam seus orçamentos manualmente ou utilizam ferramentas genéricas que não atendem plenamente às suas necessidades. Com o avanço da tecnologia, a digitalização desses processos tornou-se essencial para melhorar a eficiência e a competitividade no mercado (REZENDE, 2018).

Diante desse cenário, este relatório técnico apresenta o desenvolvimento do Orçametro, um *software* acessível e intuitivo voltado para a criação e gerenciamento de orçamentos. A ferramenta busca oferecer uma solução prática e de baixo custo, permitindo que pequenos empreendedores e trabalhadores autônomos digitalizem seus processos sem a necessidade de um grande investimento.

O desenvolvimento do sistema envolveu a utilização de tecnologias modernas para garantir facilidade de uso, desempenho e acessibilidade. Além disso, foram considerados aspectos como usabilidade e experiência do usuário para assegurar que o *software* atenda às demandas de seu público-alvo (NIELSEN, 2012). Conforme a ISO 9241-11 (2018), a usabilidade também compreende a eficácia, eficiência e satisfação com que usuários alcançam objetivos específicos em um sistema, dentro de um contexto determinado. Este relatório detalha as tecnologias utilizadas, a modelagem do projeto, a implementação do *software* e os resultados obtidos, além de apresentar as principais conclusões sobre a viabilidade e eficiência da solução proposta.

1.1 OBJETIVOS

Apresentam-se, a seguir, as metas que norteiam o desenvolvimento deste projeto, buscando alinhar as necessidades técnicas de software à demanda de mercado identificada.

1.1.1. Geral

Desenvolver um software acessível para a criação e gerenciamento de orçamentos, voltado a pequenas empresas e profissionais autônomos.

1.1.2. Específicos

- Projetar e desenvolver uma interface intuitiva para facilitar o uso do software, independentemente do nível de experiência do usuário com a tecnologia.
- Implementar funcionalidades essenciais, como cadastro de produtos e serviços, aplicação de descontos e geração de orçamentos em PDF.
- Permitir o envio de orçamentos diretamente por e-mail ou WhatsApp, otimizando o contato com os clientes.
- Garantir a segurança e integridade dos dados armazenados no sistema.
- Testar e validar a usabilidade do *software* com usuários reais para garantir sua eficiência e praticidade.

2. TECNOLOGIAS ENVOLVIDAS

Para a viabilização do Orçametro, foi selecionado um conjunto de tecnologias modernas que priorizam a escalabilidade, a segurança e a agilidade no desenvolvimento. O quadro a seguir detalha as ferramentas, frameworks e bibliotecas que compõem a *stack* tecnológica do projeto, especificando suas versões ou finalidades dentro do ecossistema da aplicação.

Categoria	Tecnologia	Versão / Descrição
Framework Principal	Next.js	v14.2.18 (App Router)
Linguagem / Biblioteca	React & TypeScript	v18.0.0 / v5.0.0
Estilização / UI	Tailwind CSS & ShadCN	Interface consistente e responsiva

Banco de Dados (ORM)	Prisma	v5.21.1
Autenticação	Clerk	Integração com Login Social (Google)
Pagamentos	Stripe	Gestão de assinaturas e checkout
Editor de Código	VS Code	Ambiente de desenvolvimento (IDE)
Versionamento	Git / GitHub	Controle de versão e repositório

Fonte: Elaborado pelo autor (2026).

O desenvolvimento do Orçametro foi realizado utilizando Next.js¹ na sua versão 14.2.18, React² na versão 18.0.0 e TypeScript³ na sua versão 5.0.0, garantindo uma aplicação moderna, escalável e com alto desempenho. A estilização e a composição dos componentes foram feitas com ShadCN, proporcionando uma interface visual consistente e eficiente. O Prisma⁴ na versão 5.21.1 foi utilizado como ORM para facilitar a comunicação com o banco de dados, garantindo uma estrutura flexível e segura.

Para o gerenciamento de autenticação e integração com *login* do Google, foi utilizado o Clerk⁵, uma solução que simplifica a autenticação e a gestão de usuários. O sistema de assinaturas foi implementado por meio do Stripe, permitindo o gerenciamento de planos e pagamentos de forma segura e automatizada.

¹ <https://nextjs.org>

² <https://react.dev>

³ <https://www.typescriptlang.org>

⁴ <https://www.prisma.io>

⁵ <https://clerk.com>

O Visual Studio Code⁶ foi escolhido como editor de código devido à sua ampla personalização, suporte a extensões e depuração eficiente. O versionamento do projeto foi realizado utilizando o GitHub⁷, permitindo o controle de versões, colaboração e integração contínua.

O banco de dados foi hospedado no NeonDB⁸, um serviço em nuvem para PostgreSQL, garantindo escalabilidade e desempenho na gestão dos dados. Já a hospedagem da aplicação foi feita na Vercel⁹, que oferece uma infraestrutura otimizada para projetos baseados em Next.js, proporcionando facilidade de *deploy* e alto desempenho.

Essas tecnologias foram escolhidas para garantir um desenvolvimento ágil, seguro e eficiente, atendendo às necessidades dos usuários do Orçametro.

3. MODELAGEM DO PROJETO

A modelagem do projeto é fundamental para estruturar a arquitetura e os requisitos do Orçametro. Conforme aponta **Guedes (2018, p. 18)**, o uso de modelos permite que o desenvolvedor visualize o comportamento e a estrutura do sistema de forma abstrata, facilitando a comunicação entre os envolvidos e a detecção precoce de falhas. Sob essa ótica, o sistema foi projetado para permitir a criação e gerenciamento de orçamentos. O sistema foi projetado para permitir a criação e gerenciamento de orçamentos, contemplando a gestão de serviços, produtos, clientes e usuários da organização, que podem ser administradores ou vendedores.

Nesta seção, são descritos os requisitos funcionais e não funcionais, acompanhados de diagramas que ilustram o funcionamento do sistema, incluindo casos de uso e a arquitetura adotada. A modelagem também contempla a estrutura do banco de dados, representada pelo diagrama de entidades e relacionamentos, garantindo a integridade das informações e a eficiência na manipulação dos dados.

⁶ <https://code.visualstudio.com>

⁷ <https://github.com>

⁸ <https://neon.tech>

⁹ vercel.com

3.1. LEVANTAMENTO DE REQUISITOS

A modelagem do **Orçametro** foi baseada na análise das necessidades dos usuários, visando oferecer uma solução eficiente e acessível para a geração de orçamentos. Os requisitos foram categorizados em **funcionais** e **não funcionais**, garantindo que o sistema atenda tanto às funcionalidades esperadas quanto aos aspectos de desempenho e segurança.

- **Requisitos funcionais:**

Os requisitos funcionais descrevem as funcionalidades diretas que o sistema deve oferecer para atender às necessidades do usuário. Eles definem as ações, entradas de dados e comportamentos esperados do software durante a sua operação. O quadro a seguir detalha as funções principais implementadas no projeto.

ID	Requisito	Descrição
RF001	Gerenciamento de itens	Permitir o cadastro, edição e exclusão de serviços e produtos para composição de orçamentos.
RF002	Elaboração de orçamentos	Possibilitar a criação de orçamentos com descrições detalhadas, valores unitários e aplicação de descontos.
RF003	Exportação de documentos	Gerar orçamentos em formato PDF para compartilhamento via e-mail ou WhatsApp.

RF004	Autenticação de usuários	Prover acesso seguro via Google Authentication integrado à plataforma Clerk.
RF005	Gestão de assinaturas	Controlar planos e pagamentos recorrentes através da integração com o Stripe.
RF006	Controle de níveis de acesso	Diferenciar permissões entre Admin (acesso total) e Vendedor (operacional de orçamentos).

- **Requisitos não-funcionais:**

Os requisitos funcionais descrevem as funcionalidades diretas que o sistema deve oferecer para atender às necessidades do usuário. Eles definem as ações, entradas de dados e comportamentos esperados do software durante a sua operação. O quadro a seguir detalha as funções principais implementadas no projeto.

ID	Categoria	Descrição
RNF001	Disponibilidade	O sistema deve operar em regime 24/7, garantindo acesso contínuo aos usuários.

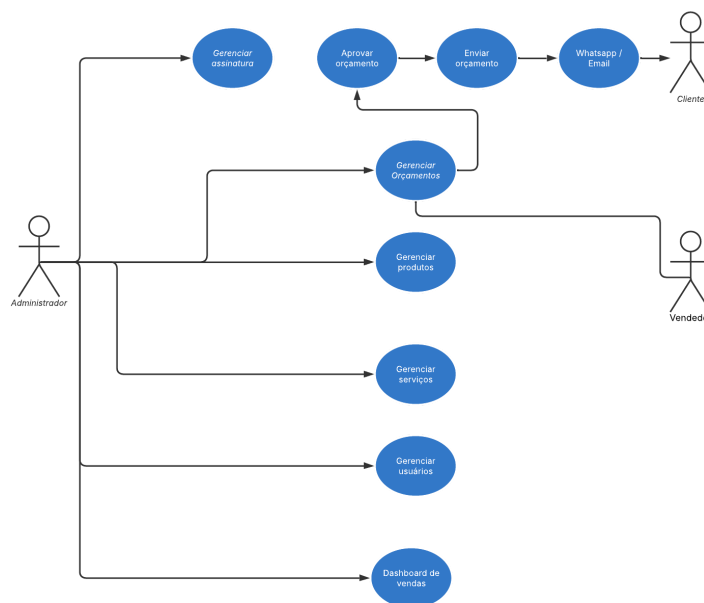
RNF002	Escalabilidade	Suportar até 1.000 usuários simultâneos e 10.000 registros via NeonDB (PostgreSQL) sem perda de performance.
RNF003	Segurança	Implementação de protocolo HTTPS, OAuth 2.0 e criptografia bcrypt para proteção de dados e sessões.
RNF004	Desempenho	Tempo de resposta inferior a 1 segundo em 90% das requisições, otimizado pelo ecossistema Next.js/Vercel.
RNF005	Usabilidade	Interface intuitiva e responsiva adaptada para diferentes dispositivos (Desktop e Mobile).

3.2. DIAGRAMAS DE CASOS DE USO

Conforme ilustrado na Figura 1, o diagrama de casos de uso apresenta os principais atores do sistema e suas interações, demonstrando como o usuário administrador, o vendedor e o cliente utilizam as funcionalidades do Orçametro.

O primeiro caso representa o usuário administrador, o mesmo é responsável pela gestão completa do sistema, incluindo o gerenciamento de serviços, produtos, usuários e configurações gerais da sua organização. O segundo caso mostra um ator com papel de vendedor, ele pode criar e gerenciar orçamentos, mas não possui permissões para modificar as configurações gerais da organização. O terceiro caso representa um ator cliente, responsável por receber os orçamentos gerados no sistema por meio dos canais de envio disponíveis, e-mail e Whatsapp.

Figura 1 - Diagrama de caso de uso



Fonte: Autor do relatório

3.3. ARQUITETURA DO SISTEMA

A arquitetura de *software* é um dos pilares fundamentais no desenvolvimento de sistemas escaláveis, modulares e de fácil manutenção. Segundo Bass, Clements e Kazman (2012), a definição da arquitetura influencia diretamente na qualidade do *software*, incluindo aspectos como desempenho, segurança e escalabilidade. O Orçametro foi desenvolvido utilizando um modelo arquitetural baseado em componentes e serviços. Conforme afirma **Pressman (2016, p. 242)**, "a engenharia de software baseada em componentes permite que o arquiteto de software estruture o sistema como uma série de unidades de software independentes que se comunicam através de interfaces bem definidas". Essa abordagem garante flexibilidade na evolução do sistema e facilita a manutenção. O sistema foi construído com Next.js e TypeScript, adotando uma estrutura modular para facilitar a organização do código. De acordo com Fowler (2002), a separação de responsabilidades dentro de um sistema facilita a reutilização de código e melhora a testabilidade.

A estrutura do diretório do projeto segue um padrão organizado:

app/ – Diretório principal da aplicação, contendo páginas e componentes.

_components/ui/ – Conjunto de componentes reutilizáveis, como botões, *inputs* e modais.

api/ – *Endpoints* responsáveis pela comunicação entre *frontend* e *backend*, incluindo integrações com *Stripe* e envio de e-mails.

budgets/ – Módulo de gerenciamento de orçamentos.

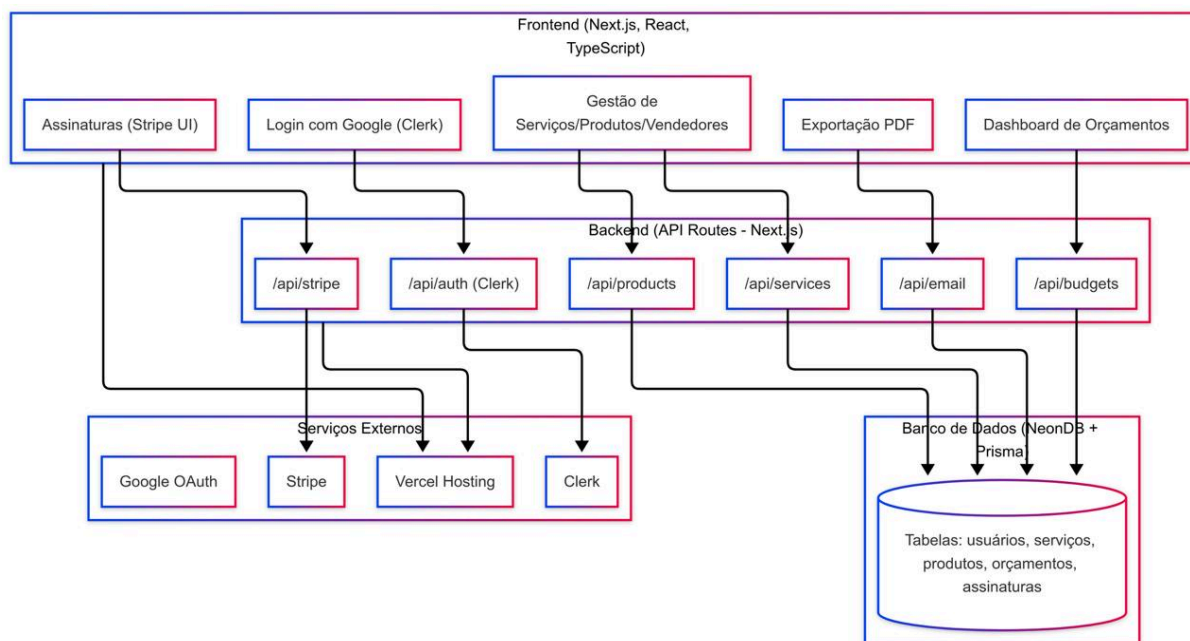
sellers/ – Gestão de vendedores cadastrados na plataforma.

services/ – Controle dos serviços e produtos cadastrados.

subscriptions/ – Gerenciamento de assinaturas integradas ao *Stripe*.

A Figura 2 exibe a arquitetura do *software*, evidenciando a separação dos módulos e a comunicação entre o *frontend*, *backend* e o banco de dados, o que garante escalabilidade e modularidade ao sistema.

Figura 2 - Diagrama de arquitetura do software

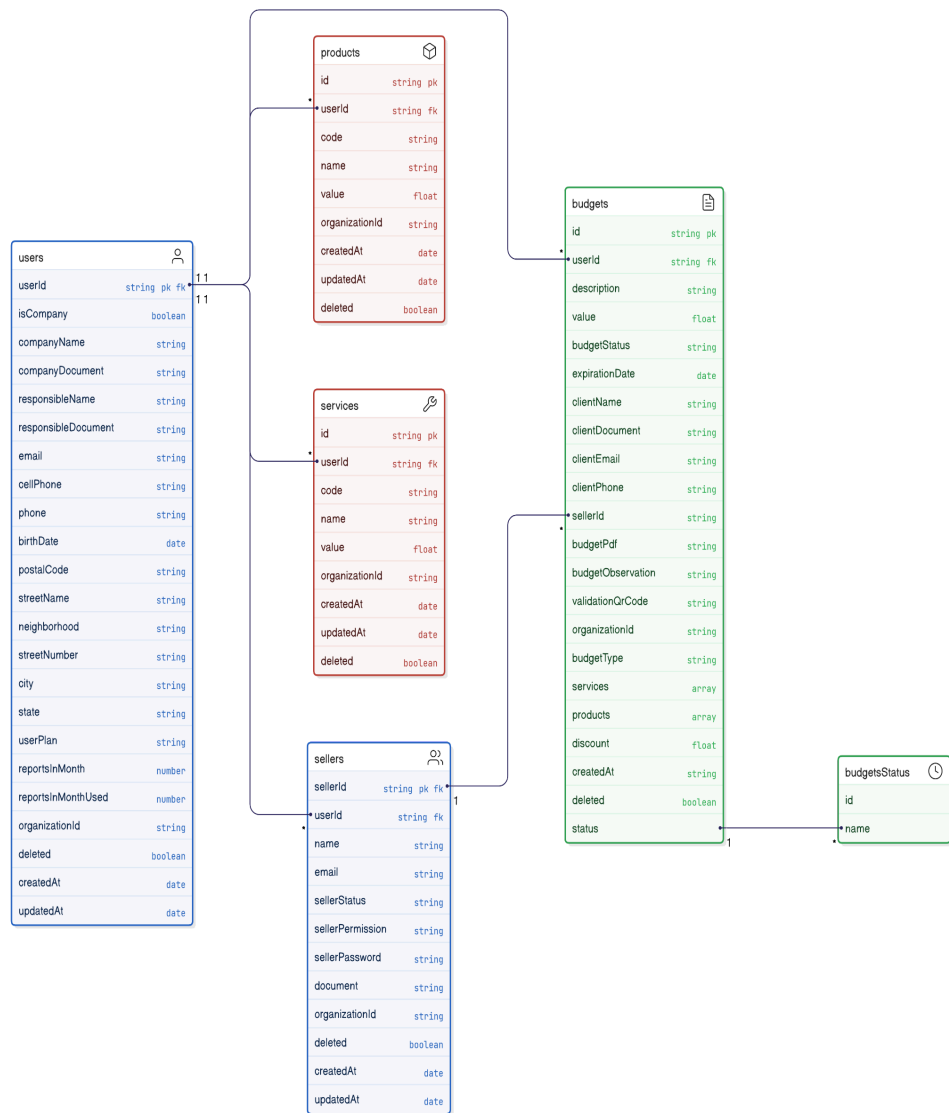


Fonte: Autor do relatório

3.4. DIAGRAMAS DE ENTIDADES-RELACIONAMENTOS

Na Figura 3, o Diagrama Entidade-Relacionamento ilustra as principais entidades do sistema, bem como seus relacionamentos, proporcionando uma visão detalhada da modelagem orientada a objetos adotada para o desenvolvimento do Orçametro. De acordo com Sommerville (2019), os diagramas de entidades e relacionamentos são fundamentais para a estruturação de sistemas, pois permitem a definição clara das interações entre os componentes, facilitando a manutenção e a escalabilidade do *software*. Dessa forma, o diagrama auxilia na compreensão da arquitetura do sistema, garantindo um design mais coeso e bem estruturado.

Figura 3 - Diagrama de entidade relacionamento



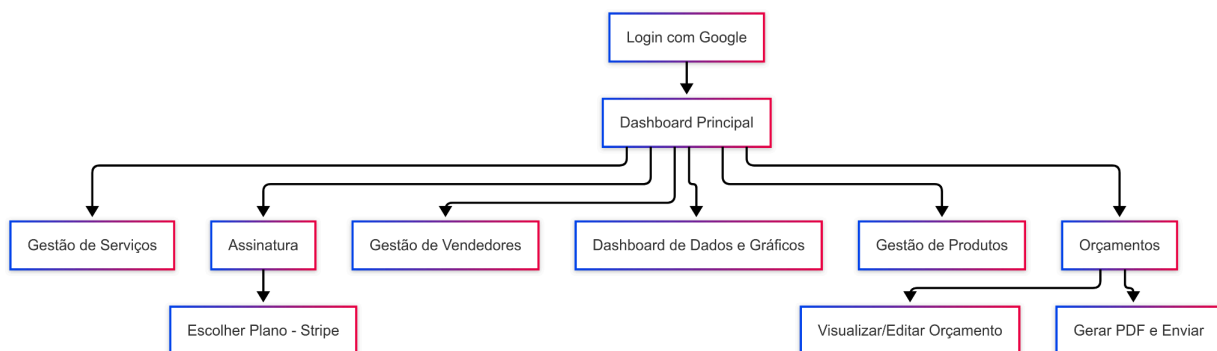
Fonte: Autor do relatório

3.5. INTERFACE

Toda a interface do sistema foi construída utilizando Typescript, React.js e Next.js. O intuito da aplicação é ser o mais simples e intuitivo possível, com ícones que tentam traduzir sua respectiva função para o usuário e utilizando na maior parte de suas funcionalidades a regra dos 3 cliques, princípio informal citado por Krug (2014), sugere que os usuários devem conseguir acessar qualquer funcionalidade do sistema com no máximo três interações.

Conforme demonstrado na Figura 4, o diagrama de navegação define a estrutura e o fluxo de interação do usuário, destacando as diferentes telas e níveis de acesso dentro do sistema.

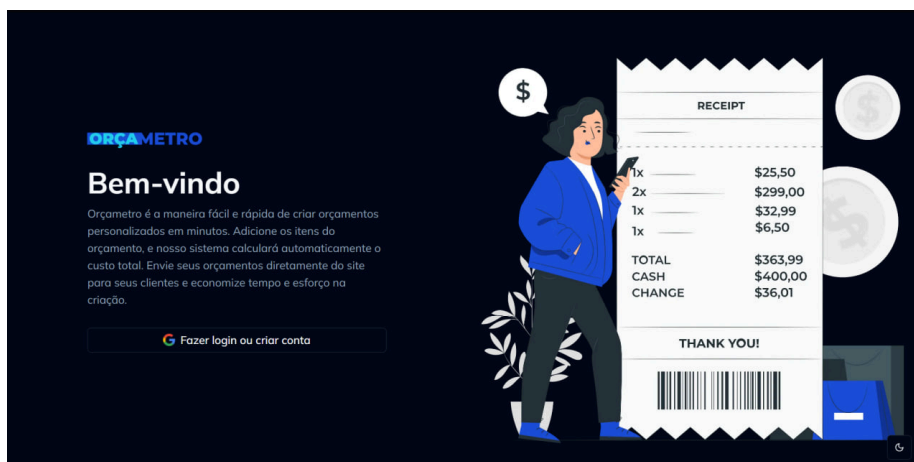
Figura 4 - Diagrama de navegação e nível de usuário do sistema



Fonte: Autor do relatório

A Figura 5 ilustra a tela inicial de login da aplicação, onde o usuário pode realizar *login* por sua conta Google. A tela dispõe de um texto introdutório sobre o sistema e um botão para realizar login.

Figura 5 - Interface de *login* do aplicativo



Fonte: Autor do relatório

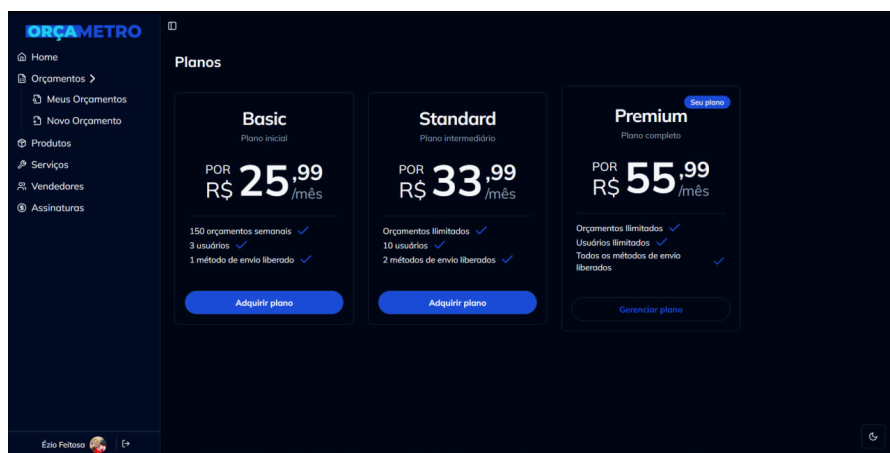
A Figura 6 demonstra a página de cadastro do usuário na aplicação, na tela são solicitados dados pessoais e caso seja uma empresa, empresariais, como nome e CNPJ.

Figura 6 - Cadastro de usuário

Fonte: Autor do relatório

A Figura 7 exibe a página de contratação de assinatura, efetuando pagamento integrado com Stripe. Nessa tela, é possível realizar a gestão da assinatura através do *gateway* de pagamento Stripe, integração feita diretamente via API. Os valores e funcionalidades de cada plano são informados no seu card específico. Também é possível gerenciar seu plano, como alterar o método de pagamento, trocar de plano e cancelar.

Figura 7 - Contratação de plano

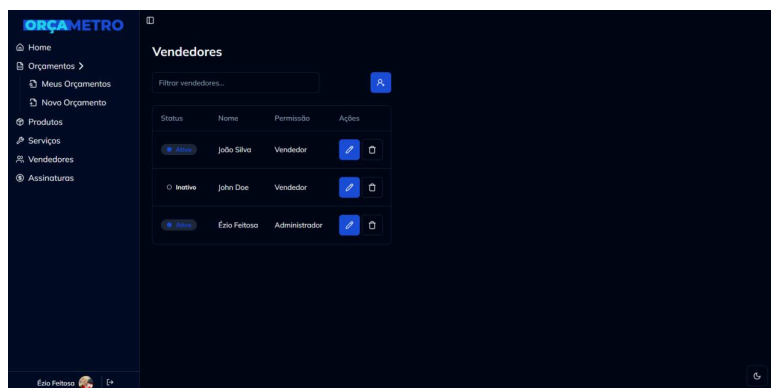


Fonte: Autor do relatório

O processo de gestão de usuários por empresa é ilustrado das figuras 8 a 11, sendo possível visualizar, editar, criar e excluir. Ao realizar uma criação ou edição, um *drawer* lateral é mostrado para o usuário solicitando as informações necessárias, já na exclusão de usuário, um *modal* de confirmação aparece ao centro da tela.

A Figura 8 mostra a página destinada ao gerenciamento de vendedores, possibilitando a visualização dos cadastros já realizados no sistema.

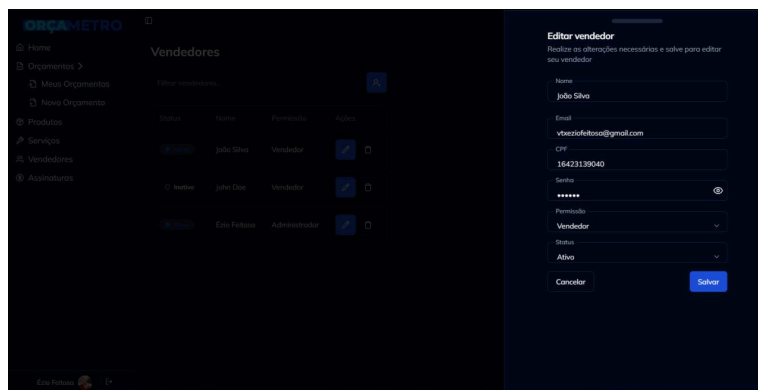
Figura 8 - Página de vendedores



Fonte: Autor do relatório

A figura 9 representa a edição de um vendedor, representada por um *drawer* lateral que exibe e possibilita a edição de informações como a senha para aprovação de orçamentos e o nível de acesso do vendedor.

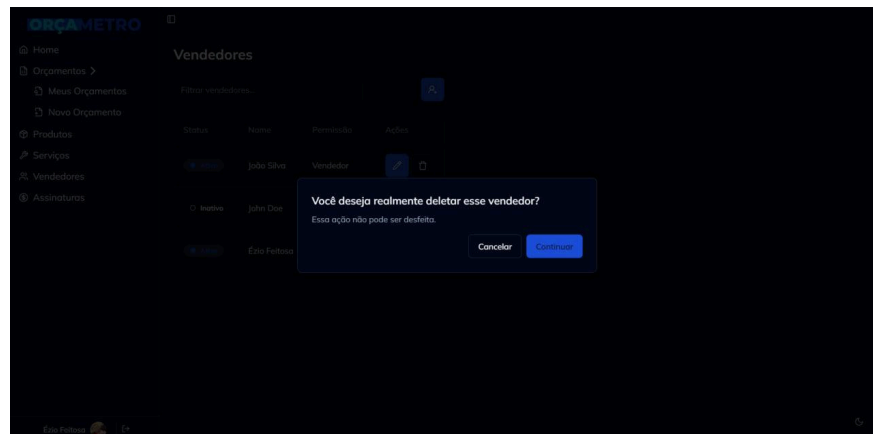
Figura 9 - Edição de vendedores.



Fonte: Autor do relatório

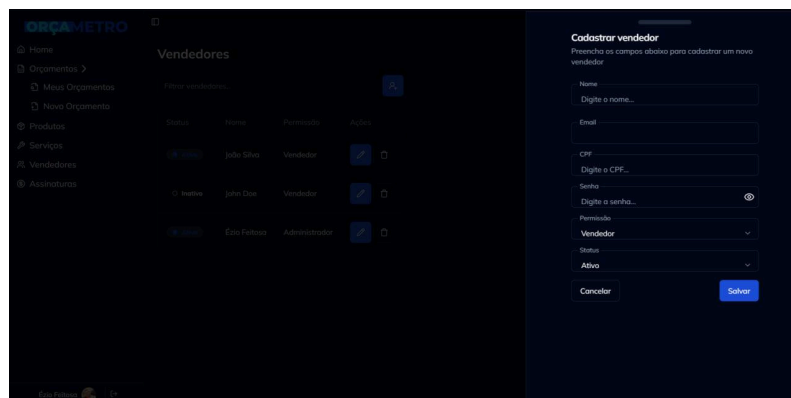
A figura abaixo ilustra a exclusão de um vendedor, representada por um *modal* de confirmação.

Figura 10 - Exclusão de vendedores



Fonte: Autor do relatório

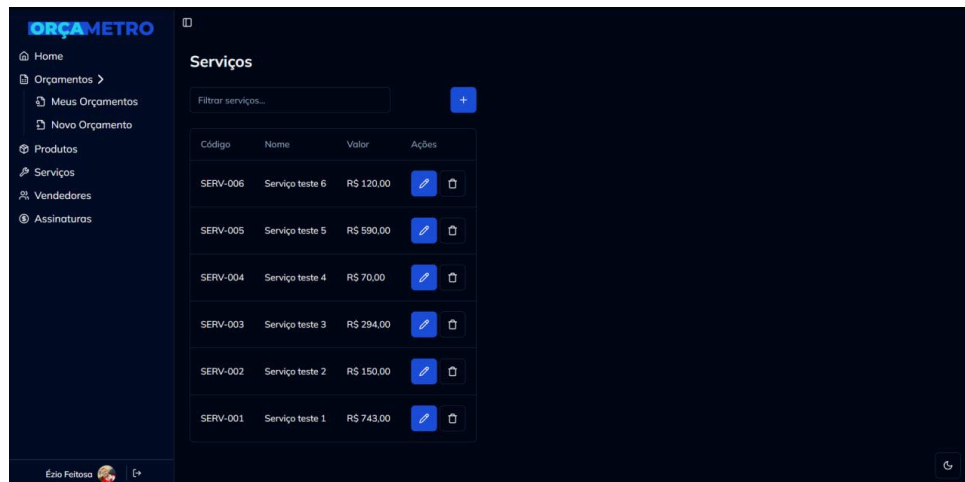
Figura 11 - Cadastro de vendedores.



Fonte: Autor do relatório

As ilustrações 12 a 15 são referentes a gestão de serviços, funcionalidades responsáveis por listar, criar, editar e excluir os mesmos. Seguindo a mesma lógica da gestão de usuários, a criação e edição são através do *drawer* lateral que aparecem ao clicar nos ícones de edição e criação, representados pelo lápis e pelo sinal de mais, respectivamente. Já a exclusão é representada pela lixeira.

Figura 12 - Listagem de serviços.

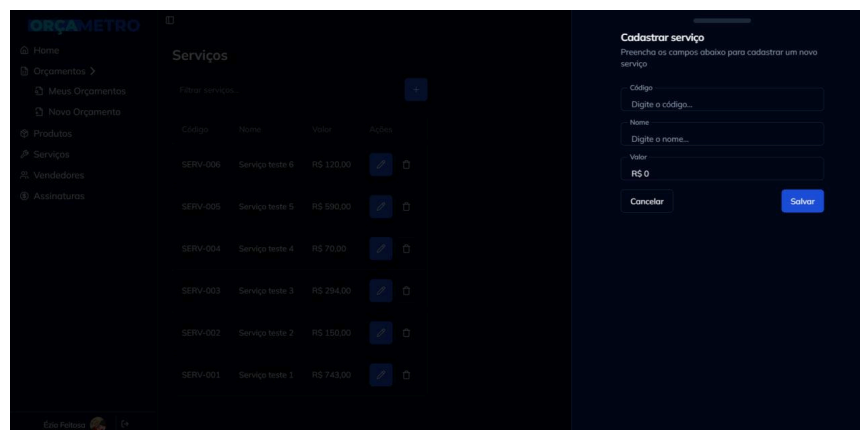


Código	Nome	Valor	Ações
SERV-006	Serviço teste 6	R\$ 120,00	 
SERV-005	Serviço teste 5	R\$ 590,00	 
SERV-004	Serviço teste 4	R\$ 70,00	 
SERV-003	Serviço teste 3	R\$ 294,00	 
SERV-002	Serviço teste 2	R\$ 150,00	 
SERV-001	Serviço teste 1	R\$ 743,00	 

Fonte: Autor do relatório

A figura abaixo demonstra um *drawer* de criação de um serviço, a criação tem como objetivo ser simples, solicitando apenas dados essenciais para a utilização do mesmo.

Figura 13 - Criação de serviços.



Cadastrar serviço
Preencha os campos abaixo para cadastrar um novo serviço.

Código
Digite o código...

Nome
Digite o nome...

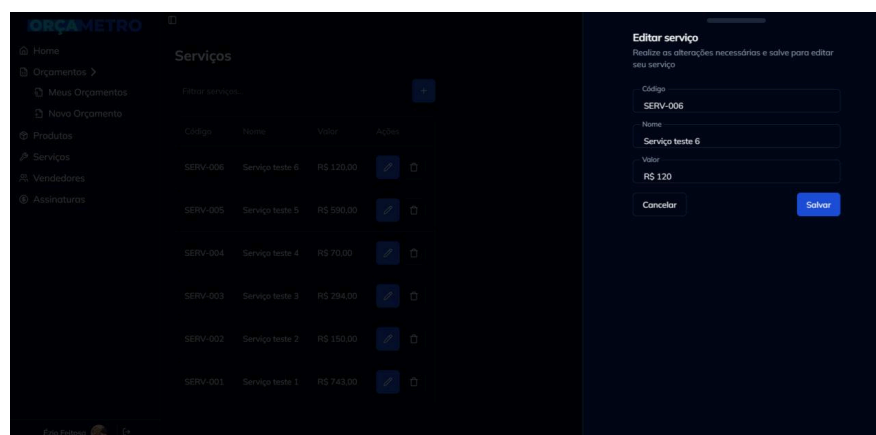
Valor
R\$ 0

Cancelar Salvar

Fonte: Autor do relatório

A figura 14 representa a edição de um serviço, na tela em questão foi utilizado uma lógica de *upsert*, onde é possível visualizar e também editar o serviço.

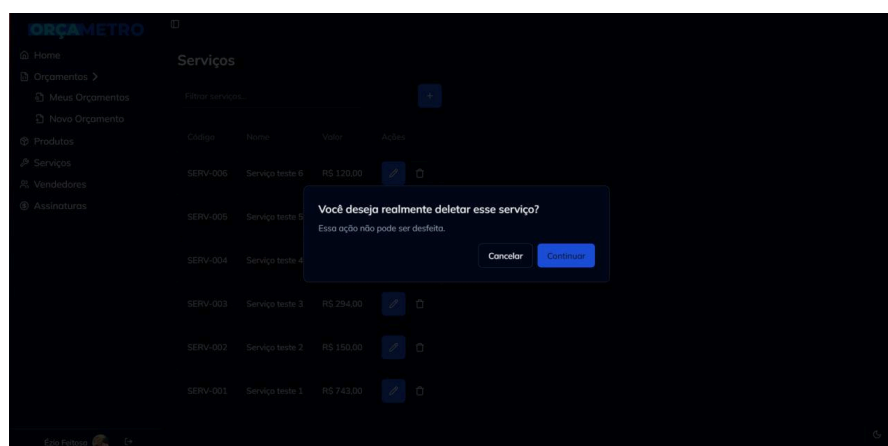
Figura 14 - Edição de serviços.



Fonte: Autor do relatório

A próxima Figura ilustra a exclusão de um serviço, solicitando através de um modal uma confirmação para realizar a ação.

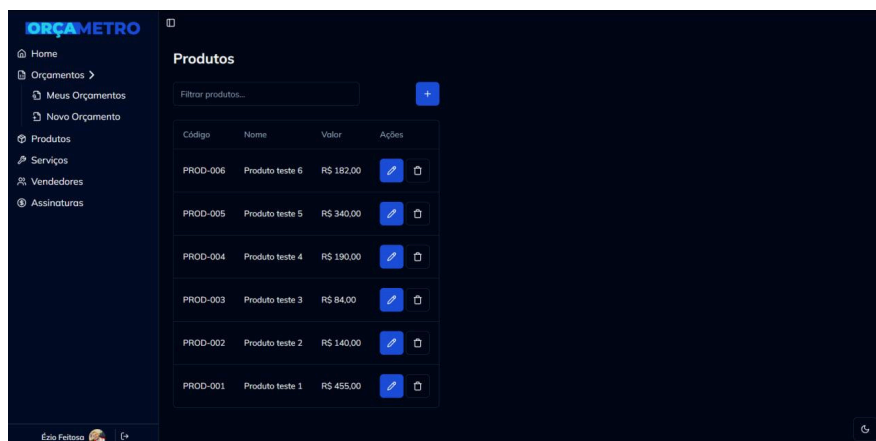
Figura 15 - Exclusão de serviços.



Fonte: Autor do relatório

As Figuras 16 a 19 exibem as telas responsáveis pela gestão de produtos da aplicação, possibilitando realizar um CRUD completo. A mesma lógica das outras funcionalidades é seguida na tela abaixo, tendo como objetivo deixar o sistema mais intuitivo e fácil de ser utilizado.

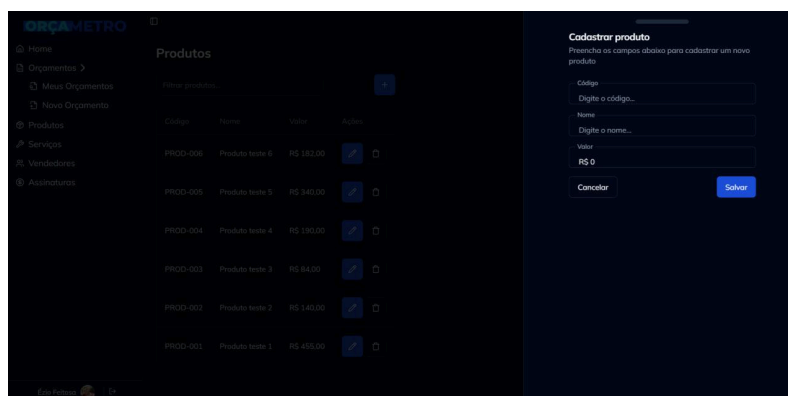
Figura 16 - Interface de produtos.



Código	Nome	Valor	Ações
PROD-006	Produto teste 6	R\$ 182,00	 
PROD-005	Produto teste 5	R\$ 340,00	 
PROD-004	Produto teste 4	R\$ 190,00	 
PROD-003	Produto teste 3	R\$ 84,00	 
PROD-002	Produto teste 2	R\$ 140,00	 
PROD-001	Produto teste 1	R\$ 455,00	 

Fonte: Autor do relatório

Figura 17 - Cadastro de produtos



Cadastrar produto
Preencha os campos abaixo para cadastrar um novo produto

Código
Digite o código...

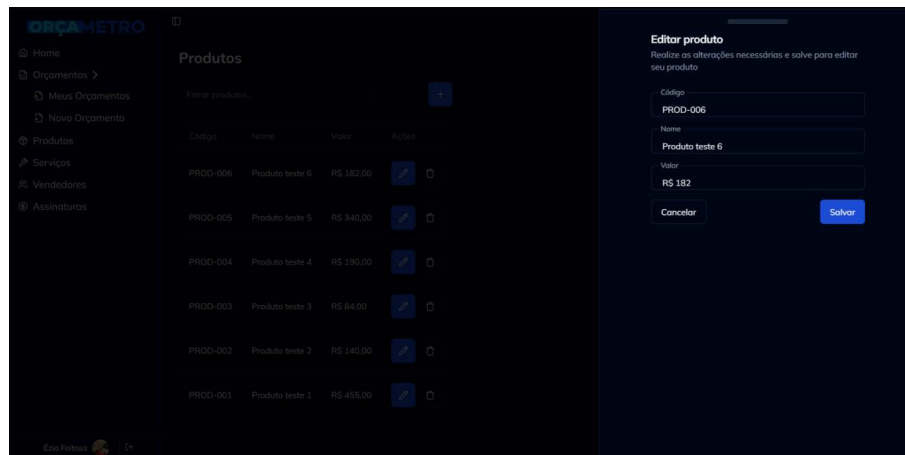
Nome
Digite o nome...

Valor
R\$ 0

Cancelar Salvar

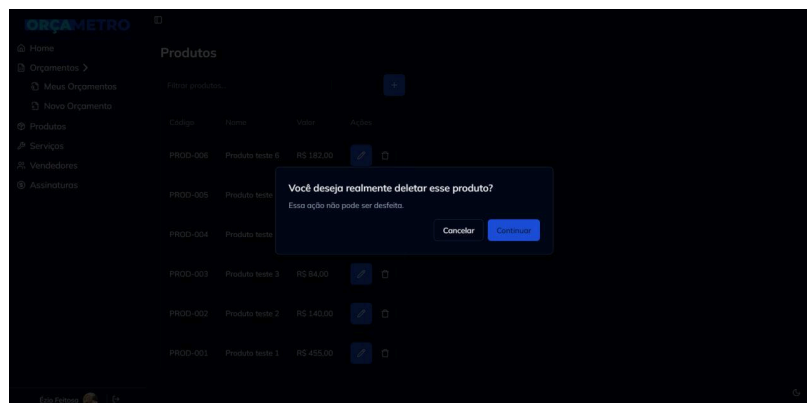
Fonte: Autor do relatório

Figura 18 - Edição de produtos



Fonte: Autor do relatório

Figura 19 - Exclusão de produtos



Fonte: Autor do relatório

As figuras 20 a 23 exibem as funcionalidades possíveis para um orçamento, como listagem, exclusão, validação, criação, *download* e exclusão e envio do mesmo. A tela de listagem de orçamento possui ícones, *badges* e cores representativas, utilizando o verde para aprovado, vermelho para cancelado e branco para pendente na representação do status do orçamento.

Todos os orçamentos possuem prazo de validade que expira após um mês da criação. Também é possível saber qual o valor e o tipo do orçamento, representado por produto, serviço e híbrido. O ícone de olho representa a visualização das informações de um

orçamento específico, a função abre um *drawer* com informações detalhadas do mesmo e botões que possibilitam realizar as ações referentes ao orçamento.

Figura 20 - Listagem de orçamentos

Status	Cliente	Descrição	Valor	Tipo	Ações
Cancelado	Ézio Feltosa	Orçamento - 079.178.703-67 - 24/03/2025	R\$ 530,00	Produto	
Pendente	Ézio Feltosa	Orçamento - 079.178.703-67 - 24/03/2025	R\$ 190,00	Produto	
Aprovado	Ézio Feltosa	Orçamento - 079.178.703-67 - 24/03/2025	R\$ 935,00	Produto	
Aprovado	Ézio Feltosa	Orçamento - 079.178.703-67 - 24/03/2025	R\$ 1.226,00	Híbrido	
Aprovado	Ézio Feltosa	Orçamento - 079.178.703-67 - 24/03/2025	R\$ 1.256,00	Híbrido	
Aprovado	Ézio Feltosa	Orçamento - 079.178.703-67 - 24/03/2025	R\$ 1.227,00	Serviço	
Aprovado	Ézio	Orçamento - 079.178.703-67 - 24/03/2025	R\$ 721,00	Produto	

Fonte: Autor do relatório

Figura 21 - Visualização, envio e validação de orçamento

Informações do Orçamento

Orçamento - 079.178.703-67 - 24/03/2025 Aprovado Produto

Informações do Cliente

Nome: Ézio Feltosa Documento: 079.178.703-67

E-mail: eziof.dev@gmail.com Telefone: (89) 99423-9661

Informações do Orçamento

Vendedor: João Silva Valor: R\$ 935,00

Produtos: Visualizar os produtos Serviços: Visualizar os serviços

Data de criação: 24/03/2025 Data de expiração: 23/04/2025

Observações: Sem observações

Fechar

Fonte: Autor do relatório

Figura 22 - Criação de Orçamentos

ORÇAMETRO

Home
Orçamentos >
Meus Orçamentos
Novo Orçamento
Produtos
Serviços
Vendedores
Assinaturas

Novo Orçamento

Nome do Cliente: Ézio Feltosa
CPF do responsável: 713.084.920-73
Email do cliente: teste@gmail.com
Celular: (89) 99999-9999
Vendedor: Ézio Feltosa
Produtos: Produto teste 1, Produto teste 2, Produto teste 3
Serviços: Serviço teste 1, Serviço teste 2, Serviço teste 3
TESTE

PROD-003	Produto teste 3	R\$ 84,00	1	R\$ 84,00
SERV-001	Serviço teste 1	R\$ 743,00	1	R\$ 743,00
SERV-002	Serviço teste 2	R\$ 150,00	1	R\$ 150,00
SERV-003	Serviço teste 3	R\$ 294,00	1	R\$ 294,00

Desconto %: 10
Subtotal: R\$ 1.866,00
Total: R\$ 1.679,40

Simulação de valores do orçamento
Gerar Orçamento

Fonte: Autor do relatório

Figura 23 - Exclusão de orçamento

ORÇAMETRO

Home
Orçamentos >
Meus Orçamentos
Novo Orçamento
Produtos
Serviços
Vendedores
Assinaturas

Meus Orçamentos

Filtrar orçamentos...

Status	Cliente	Descrição	Valor	Tipo	Ações
Pendente	Ézio Feltosa	Orçamento - 079.178.703-67 - 24/03/2025	R\$ 530,00	Produto	Editar, Excluir
Pendente	Ézio Feltosa	Orçamento - 079.178.703-67 - 24/03/2025	R\$ 1.226,00	Híbrido	Editar, Excluir
Pendente	Ézio Feltosa	Orçamento - 079.178.703-67 - 24/03/2025	R\$ 1.256,00	Híbrido	Editar, Excluir
Pendente	Ézio Feltosa	Orçamento - 079.178.703-67 - 24/03/2025	R\$ 1.227,00	Híbrido	Editar, Excluir
Pendente	Ézio Feltosa	Orçamento - 079.178.703-67 - 24/03/2025	R\$ 721,00	Produto	Editar, Excluir

Você deseja realmente deletar esse orçamento?
Essa ação não pode ser desfeita.
Cancelar Continuar

Fonte: Autor do relatório

A figura 24 exibe uma *dashboard* de análise de dados que possui informações da empresa em tempo real, gráficos com informações importantes que auxiliam na tomada de decisão e gestão da empresa. As informações expostas na tela são, faturamento total e mensal da empresa, vendas pendentes e realizadas. Um gráfico de barras que representa os valores vendidos por vendedor e um gráfico de área representando as vendas e faturamento mensal de todo o ano.

Figura 24 - Dashboard de análise de dados.



Fonte: Autor do relatório

4. SOFTWARE

A implantação e os testes do Orçametro foram conduzidos de forma estruturada para garantir um sistema funcional, confiável e alinhado às necessidades dos usuários. A implementação do *software* seguiu uma abordagem gradual, permitindo ajustes antes do lançamento oficial. Paralelamente, diversas metodologias de teste foram aplicadas para validar cada aspecto da aplicação, desde a verificação de componentes individuais até a avaliação da experiência do usuário.

A fase de testes seguiu diretrizes bem definidas, contemplando testes unitários, de integração, funcionais e de usabilidade, assegurando que o Orçametro estivesse livre de falhas e proporcionasse uma experiência satisfatória. Com essa abordagem, foi possível identificar e corrigir eventuais inconsistências, resultando em um sistema robusto e eficiente. A seguir, são apresentados os detalhes do processo de implantação e os testes realizados.

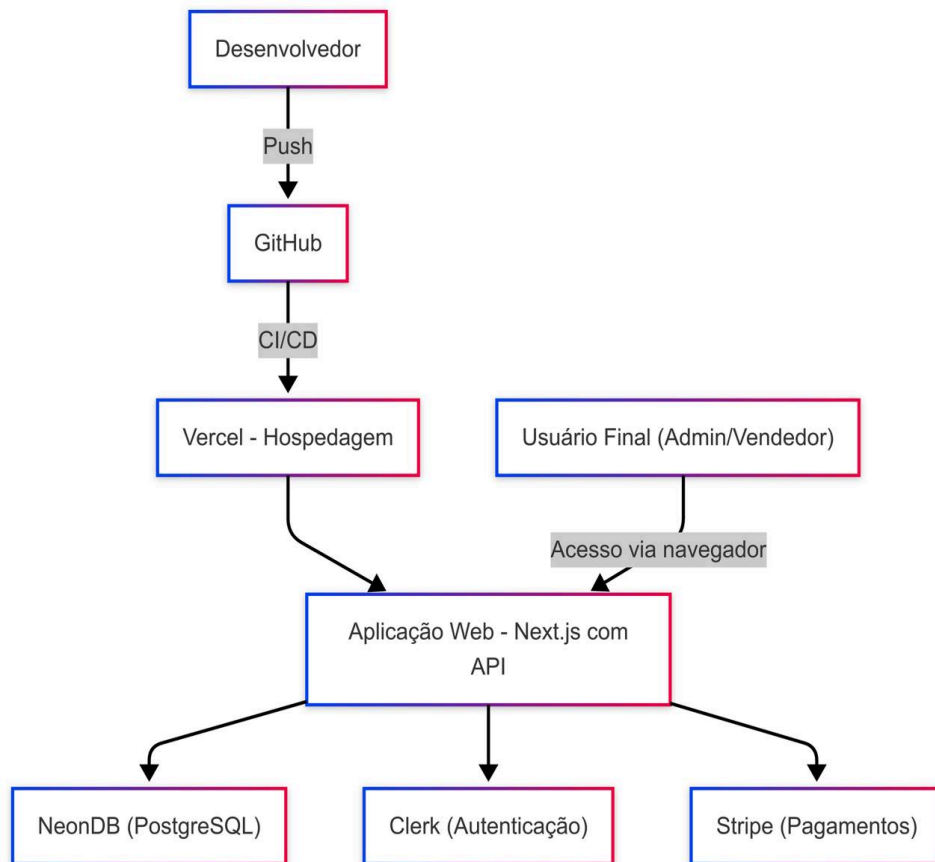
4.1. IMPLANTAÇÃO

A implantação do Orçametro foi realizada de maneira estratégica e gradual, garantindo uma adaptação eficiente dos usuários ao sistema. Inicialmente, foi disponibilizada uma versão beta para um grupo seleto de empresas, permitindo a coleta de *feedbacks* e ajustes necessários antes do lançamento completo.

A Figura 25 apresenta o diagrama de implantação do sistema Orçametro, destacando o fluxo de desenvolvimento, hospedagem e utilização da aplicação. O processo se inicia com o desenvolvedor enviando alterações para o repositório no GitHub, que, por meio de um pipeline CI/CD, realiza a implantação automática na Vercel, plataforma responsável pela hospedagem do sistema.

O usuário final, seja administrador ou vendedor, acessa a aplicação via navegador, interagindo com a interface desenvolvida em Next.js. A aplicação se comunica com diferentes serviços essenciais para seu funcionamento, como o NeonDB (PostgreSQL) para gerenciamento do banco de dados, Clerk para autenticação de usuários e Stripe para processamento de pagamentos, garantindo uma infraestrutura moderna, segura e escalável.

Figura 25 - Diagrama de implantação



Fonte: Autor do relatório

4.2. TESTES

Os testes de *software* desempenham um papel essencial no desenvolvimento do Orçametro, garantindo que o sistema funcione conforme esperado e atenda aos requisitos definidos. De acordo com Sommerville (2019), o processo de testes está diretamente relacionado aos conceitos de verificação e validação, assegurando que o *software* esteja livre de falhas e adequado ao seu propósito.

Durante o desenvolvimento, foram aplicadas diferentes abordagens de teste:

- **Testes unitários:** Implementados para validar funcionalidades individuais do sistema, garantindo que cada componente, como os serviços de autenticação (Clerk) e

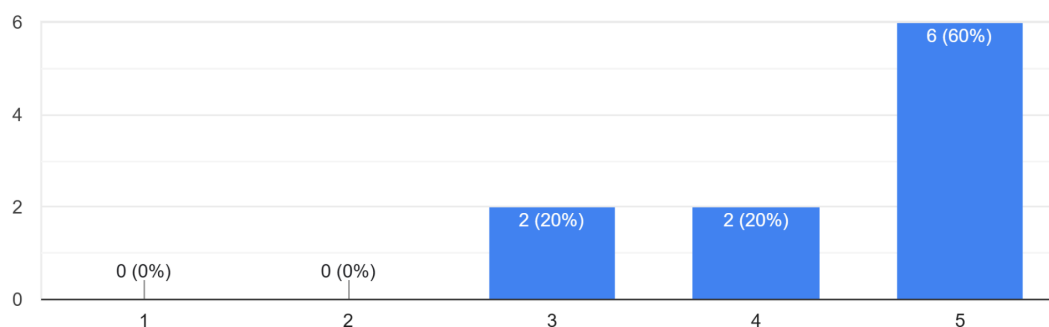
processamento de pagamentos (Stripe), funcionasse corretamente de forma isolada.

- **Testes de integração:** Realizados para verificar a comunicação entre os módulos da aplicação, como a interação entre os endpoints da API e o banco de dados gerenciado pelo NeonDB.
- **Testes funcionais:** Testes baseados nos requisitos do sistema para garantir que as funcionalidades, como criação de orçamentos e gerenciamento de assinaturas, operassem conforme esperado.
- **Testes de usabilidade SUS:** Aplicados a um grupo restrito de usuários, com o objetivo de avaliar a experiência e acessibilidade do sistema, garantindo uma interface intuitiva e eficaz. O teste foi realizado via Google Forms em um formulário de 10 perguntas em escala linear que vai de 1 a 5, sendo 1 referente a discordo totalmente e 5 referente a concordo totalmente. As respostas recebidas estão exibidas abaixo em gráficos.

Figura 26 - Teste SUS, gráfico da primeira pergunta

Eu acho que usaria este sistema com frequência.

10 respostas

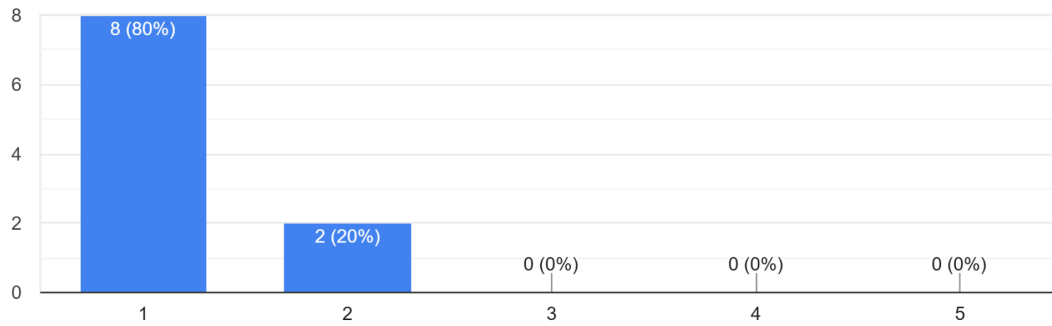


Fonte: Autor do relatório

Figura 27 - Teste SUS, gráfico da segunda

Achei o sistema desnecessariamente complexo.

10 respostas

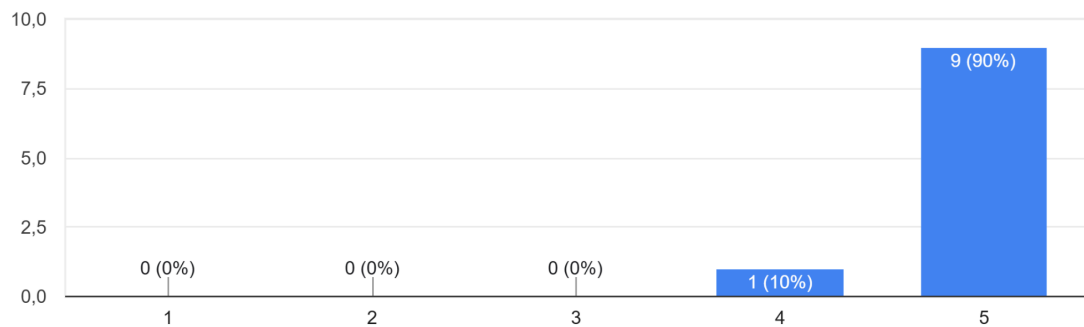


Fonte: Autor do relatório

Figura 28 - Teste SUS, gráfico da terceira pergunta

Achei o sistema fácil de usar.

10 respostas

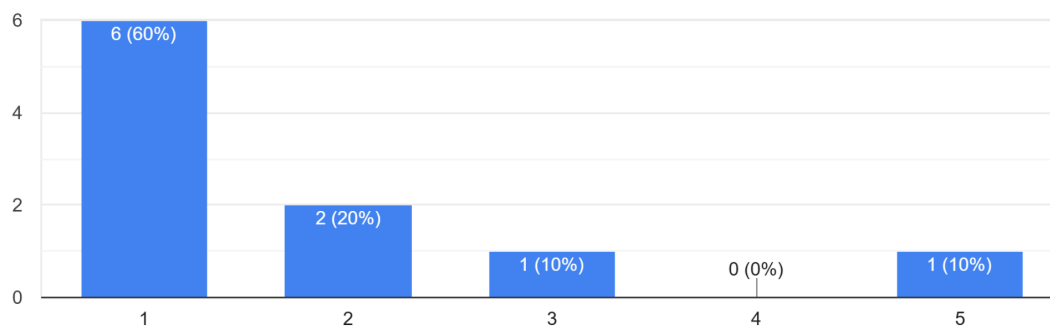


Fonte: Autor do relatório

Figura 29 - Teste SUS, gráfico da quarta pergunta

Acho que precisaria da ajuda de um técnico para conseguir usar o sistema.

10 respostas

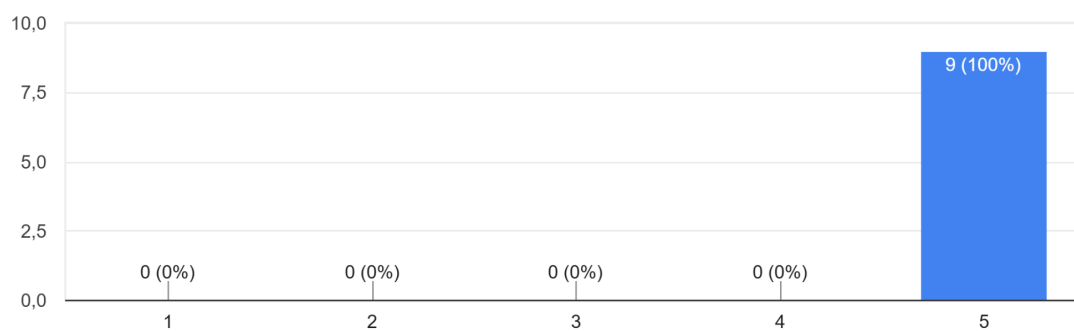


Fonte: Autor do relatório

Figura 30 - Teste SUS, gráfico da quinta pergunta

As funcionalidades do sistema estão bem integradas.

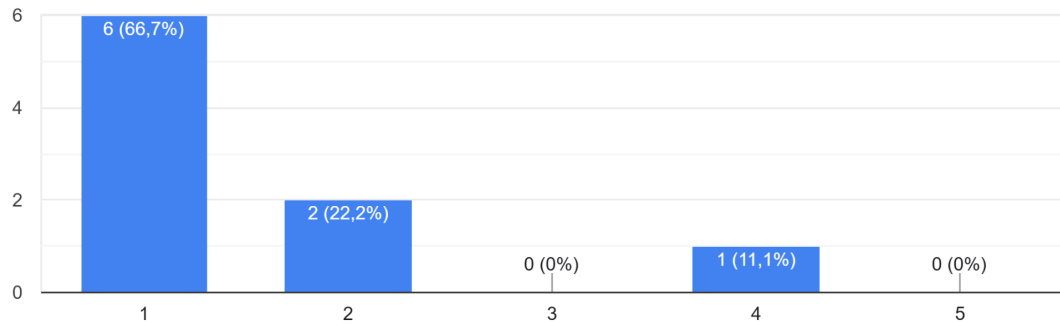
9 respostas



Fonte: Autor do relatório

Figura 31 - Teste SUS, gráfico da sexta pergunta

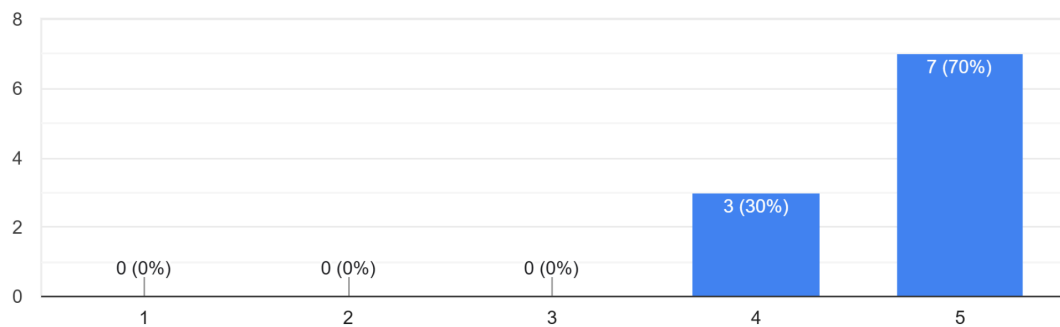
Achei que havia muita inconsistência no sistema.
9 respostas



Fonte: Autor do relatório

Figura 32 - Teste SUS, gráfico da sétima pergunta

Imagino que a maioria das pessoas aprenderia a usar este sistema rapidamente.
10 respostas

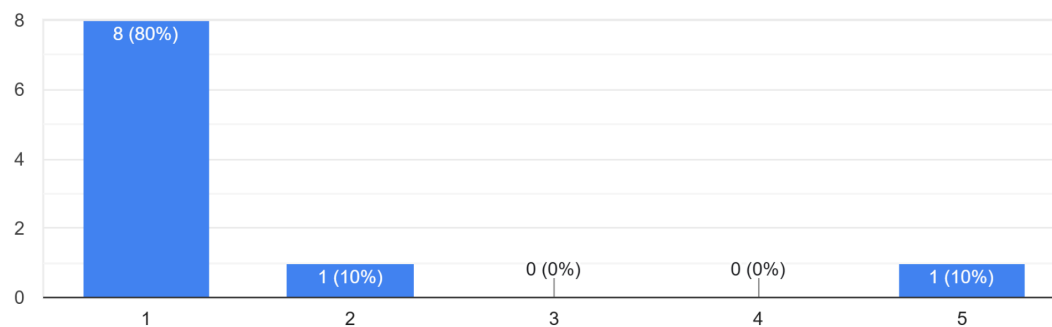


Fonte: Autor do relatório

Figura 33 - Teste SUS, gráfico da oitava pergunta

Achei o sistema muito difícil de usar.

10 respostas

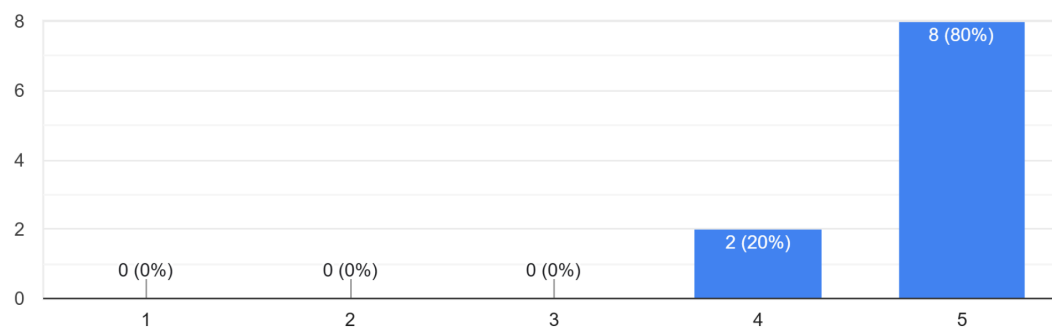


Fonte: Autor do relatório

Figura 34 - Teste SUS, gráfico da nona pergunta

Me senti confiante usando o sistema.

10 respostas

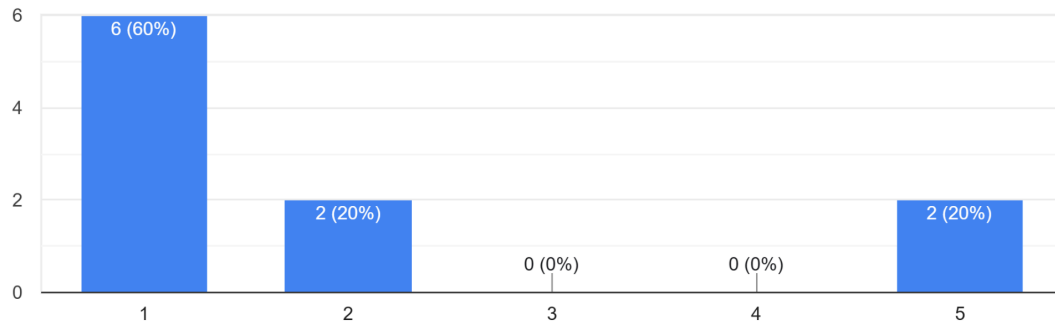


Fonte: Autor do relatório

Figura 35 - Teste SUS, gráfico da décima pergunta

Precisei aprender muitas coisas antes de conseguir usar o sistema.

10 respostas



Fonte: Autor do relatório

A validação da usabilidade do Orçametro foi realizada por meio do questionário SUS (System Usability Scale), aplicado a um grupo de usuários reais. A média geral obtida foi de 75,78 pontos, o que, segundo a escala proposta por Bangor, Kortum e Miller (2009), é considerado um nível “bom” de usabilidade. Esse resultado confirma que a aplicação é bem aceita pelos usuários em termos de navegação, simplicidade e eficiência.

4.3. ACESSO AO CÓDIGO-FONTE

O código-fonte do Orçametro está disponível publicamente no GitHub, permitindo o acesso completo ao sistema, desde sua camada de frontend, backend e integrações com APIs externas como Stripe e Clerk. A disponibilização do repositório visa garantir transparência, facilitar futuras colaborações e permitir que o projeto seja utilizado ou adaptado por outros desenvolvedores e empreendedores interessados.

O repositório pode ser acessado pelo link:

<https://github.com/ezin1/orcametro-fullstack>

4.4 ACESSO À APLICAÇÃO

A aplicação está hospedada na Vercel, uma plataforma que oferece infraestrutura otimizada para aplicações front-end e back-end, garantindo alta disponibilidade e desempenho. O acesso ao sistema pode ser realizado por meio do link disponibilizado abaixo, permitindo que os usuários utilizem as funcionalidades do Orçametro de maneira simples e eficiente, sem necessidade de instalação local.

A aplicação está hospedada e pode ser acessada através do link:

<https://orcometro-fullstack.vercel.app>

5. CONSIDERAÇÕES FINAIS

Com a conclusão do projeto Orçametro, foi possível cumprir o objetivo geral de desenvolver uma plataforma digital acessível, moderna e eficiente para a criação e gestão de orçamentos, voltada a pequenas e médias empresas e profissionais autônomos. A aplicação entrega as funcionalidades planejadas, como cadastro e controle de serviços, produtos e clientes; geração de orçamentos com aplicação de descontos personalizados; exportação em PDF; e envio direto via e-mail e WhatsApp, facilitando a comunicação com os clientes.

Os objetivos específicos também foram atendidos ao longo do desenvolvimento. A interface do sistema foi projetada para ser simples, responsiva e acessível, utilizando as tecnologias Next.js, TypeScript, Prisma ORM e NeonDB para garantir performance, segurança e facilidade de manutenção. A autenticação de usuários foi implementada por meio do serviço Clerk (OAuth 2.0), com controle de acesso por perfis distintos (administrador e vendedor). Além disso, a usabilidade da plataforma foi validada por meio da aplicação do teste SUS (System Usability Scale) com usuários reais, alcançando nota média de 75,78 pontos, o que indica boa aceitação da interface.

O sistema também se destaca por seu potencial de impacto social e econômico, ao oferecer uma alternativa de baixo custo para a digitalização de processos que antes eram manuais. Segundo o SEBRAE (2021), a adoção de soluções digitais é um fator determinante para a competitividade e sustentabilidade de pequenos negócios, especialmente no contexto

atual de transformação digital. A entrega do Orçametro contribui para esse cenário ao tornar a tecnologia mais acessível e eficiente.

Apesar dos resultados positivos, o sistema apresenta pontos que podem ser aprimorados em versões futuras. Entre as melhorias sugeridas, destacam-se: a criação de dashboards e relatórios gerenciais para visualização de desempenho financeiro; a integração com meios de pagamento locais; e a ampliação da camada de segurança com criptografia de dados sensíveis e backups automáticos. Tais recursos podem tornar a ferramenta ainda mais robusta, ampliando sua utilidade e alcance.

REFERÊNCIA

BANGOR, A.; KORTUM, P.; MILLER, J. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, [S.l.], v. 4, n. 3, p. 114–123, 2009. Disponível em: <https://uxpajournal.org/determining-what-individual-sus-scores-mean/>. Acesso em: 02 abr. 2025.

BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. *Software Architecture in Practice*. 3. ed. Boston: Addison-Wesley, 2012.

FOWLER, Martin. *Patterns of Enterprise Application Architecture*. Boston: Addison-Wesley, 2002.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO 9241-11: Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts*. 2. ed. Geneva: ISO, 2018.

KRUG, Steve. *Não me faça pensar: uma abordagem de bom senso à usabilidade na web*. 2. ed. Rio de Janeiro: Alta Books, 2014.

NIELSEN, Jakob. *Designing Web Usability: The Practice of Simplicity*. Berkeley: New Riders Publishing, 2012.

REZENDE, Denis Alcides. *Tecnologia da Informação Aplicada a Sistemas de Informação Empresariais*. São Paulo: Atlas, 2018.

SEBRAE. *Transformação Digital nas MPEs Brasileiras*. Brasília: SEBRAE, 2021. Disponível em: <https://datasebrae.com.br>. Acesso em: 31 mar. 2025.

SOMMERVILLE, Ian. *Software Engineering*. 10. ed. Boston: Pearson, 2019.

GUEDES, Gilleanes T. A. *UML 2.5 do Princípio ao Exemplo*. 1. ed. São Paulo: Érica, 2018.

PRESSMAN, Roger S.; MAXIM, Bruce R. *Engenharia de Software: Uma Abordagem Profissional*. 8. ed. Porto Alegre: AMGH, 2016.