



Instituto Federal Catarinense
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
Campus Blumenau

DANIEL DA CRUZ VIZOLI

**SISTEMA DE GERAÇÃO DE ORÇAMENTOS PARA PEQUENAS E MÉDIAS
EMPRESAS DE MANUFATURA**

Blumenau

2023

DANIEL DA CRUZ VIZOLI

**SISTEMA DE GERAÇÃO DE ORÇAMENTOS PARA PEQUENAS E MÉDIAS
EMPRESAS DE MANUFATURA**

Artigo apresentado como requisito parcial à
conclusão do curso de Tecnologia em Análise
e Desenvolvimento de Sistemas, *campus* Blumenau,
Instituto Federal Catarinense.

Orientador: Prof.^(a). Hylson Vescovi Netto, Dr

Blumenau

2023

DANIEL DA CRUZ VIZOLI

**SISTEMA DE GERAÇÃO DE ORÇAMENTOS PARA PEQUENAS E MÉDIAS
EMPRESAS DE MANUFATURA**

Este artigo foi julgado adequado para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistema e aprovado em sua forma final pelo curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal Catarinense – *Campus* Blumenau.

Orientador: Prof.^(a). Hylson Vescovi Netto, Dr

autenticação eletrônica na folha de assinaturas

Prof. Hylson Vescovi Netto, Dr.

Orientador – IFC *campus* Blumenau

BANCA EXAMINADORA

autenticação eletrônica na folha de assinaturas

Prof. Paulo Cesar Rodacki Gomes, Dr.

Instituto Federal Catarinense, campus Blumenau

autenticação eletrônica na folha de assinaturas

Prof. Aldelir Fernando Luiz, Dr.

Instituto Federal Catarinense, campus Blumenau

Blumenau

2023



DECLARAÇÃO Nº 5/2023 - CCCOMP/BLU (11.01.09.22)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 13/12/2023 19:06)

ALDELIR FERNANDO LUIZ

DIRETOR GERAL - TITULAR

DG/BLU (11.01.09.01)

Matrícula: ###010#2

(Assinado digitalmente em 14/12/2023 08:18)

HYLSON VESCOVI NETTO

PROFESSOR ENS BASICO TECN TECNOLOGICO

CCCOMP/BLU (11.01.09.22)

Matrícula: ###100#1

(Assinado digitalmente em 14/12/2023 13:55)

PAULO CESAR RODACKI GOMES

PROFESSOR ENS BASICO TECN TECNOLOGICO

CGE/BLU (11.01.09.01.03.07)

Matrícula: ###299#3

Visualize o documento original em <https://sig.ifc.edu.br/documentos/> informando seu número: 5, ano: 2023, tipo:
DECLARAÇÃO, data de emissão: 13/12/2023 e o código de verificação: **a7383ea27c**

SISTEMA DE GERAÇÃO DE ORÇAMENTOS PARA PEQUENAS E MÉDIAS EMPRESAS DE MANUFATURA

Daniel da Cruz Vizoli

RESUMO

Este trabalho apresenta um sistema de geração de orçamentos voltado para pequenas e médias empresas com pouca experiência ou ligação com computadores e novas tecnologias. O sistema foi criado com o objetivo de auxiliar esses empreendedores na geração e envio de orçamentos para seus clientes de forma simplificada e eficiente. O sistema permite a criação de orçamentos personalizados, exportação para o formato PDF, inclusão de seções e itens, além de facilitar o armazenamento e recuperação desses orçamentos.

Palavras-chave: Pequenas empresas, orçamentos, PDF, geração de documentos, empreendedorismo

ABSTRACT

This work presents a budget generation system aimed at small and medium-sized companies with little experience or connection with computers and new technologies. The system was created with the aim of assisting these entrepreneurs in generating and sending quotes to their clients in a simplified and efficient way. The system allows the creation of personalized budgets, export to PDF format, inclusion of budgets and items, in addition to facilitating the storage and retrieval of these budgets.

Keywords: Small Businesses, Budgets, PDF, Document Generation, Entrepreneurship

1 1 INTRODUÇÃO

A sociedade contemporânea está intrinsecamente ligada à evolução tecnológica, que não apenas simplifica, mas também revoluciona a maneira como enfrentamos desafios cotidianos. Através da integração de soluções tecnológicas, muitos obstáculos que antes pareciam insuperáveis agora podem ser abordados de maneiras inovadoras e acessíveis. "A tecnologia é nada. O importante é que você tenha fé nas pessoas, que elas são basicamente boas e inteligentes, e se você der a elas ferramentas, elas farão coisas maravilhosas com elas" (JOBS, 2013). Essa visão ressalta a transformação significativa que a tecnologia tem proporcionado em nossas vidas diárias, capacitando-nos a enfrentar desafios complexos de maneiras anteriormente inimagináveis.

No cenário brasileiro, a relevância da tecnologia para empresas, especialmente as de pequeno e médio porte no setor de manufatura, assume um papel crucial. Contudo, é notório que a maior parte dos empreendedores e empresas que estão começando ou possuem vínculos com a manufatura enfrentam desafios em relação ao domínio das tecnologias modernas. Muitos desses empreendedores se veem diante da carência de recursos para contratar profissionais especializados ou investir em treinamento, o que limita sua capacidade de adotar soluções inovadoras e eficazes.

A elaboração de orçamentos é uma atividade crítica para as empresas, especialmente para as de pequeno e médio porte no setor de manufatura. Entretanto, muitas vezes, essas empresas enfrentam restrições financeiras que dificultam a contratação de profissionais especializados ou a realização de treinamentos para executar esse processo de maneira eficiente. Nesse contexto, o presente trabalho oferece um sistema que contém uma perspectiva genérica para a geração simplificada de orçamentos, atendendo às necessidades específicas de várias dessas empresas e possibilita uma gestão financeira mais eficaz.

2 1.1 TEMA/PROBLEMA

O tema abordado neste trabalho é o desenvolvimento de um sistema que permita a geração de orçamentos em formato PDF de maneira simplificada para pequenas e médias empresas e empresários com pouca experiência em tecnologia. O problema que se pretende resolver é a dificuldade enfrentada por esses empreendedores ao criar orçamentos de maneira eficiente e profissional, devido à falta de conhecimento técnico e familiaridade com ferramentas de geração de documentos.

1.2 OBJETIVOS PROPOSTOS/SOLUÇÃO DOS PROBLEMAS

O principal objetivo deste trabalho é desenvolver um sistema que possibilite a geração de orçamentos em formato PDF de maneira fácil e acessível para pequenas e médias empresas. Os objetivos específicos incluem:

- Criar uma interface intuitiva e amigável que permita aos usuários inserirem informações relevantes para a geração do orçamento.
- Implementar um mecanismo para adicionar seções e itens ao orçamento, permitindo a personalização do documento.
- Integrar uma funcionalidade para armazenar os orçamentos gerados, permitindo seu acesso posterior.
- Garantir a geração de um documento PDF esteticamente agradável e profissional a partir dos dados inseridos pelos usuários.

3 1.3 ESCOPO

O sistema desenvolvido engloba a criação de uma interface web que permite aos usuários, por meio de autenticação, gerar orçamentos com base em informações do cliente, da empresa e dos itens a serem oferecidos, como mão de obra e produtos. O orçamento e todas as informações ficam armazenadas no sistema, associadas a um *login*.

1.4 VIABILIDADE DO PROJETO

O sistema, acessível via navegadores web, é projetado para operar em computadores pessoais, tablets e celulares. A acessibilidade multiplataforma amplia a utilidade do sistema, permitindo que os usuários gerem orçamentos de maneira conveniente. Além disso, a seleção de tecnologias gratuitas e de código aberto reflete uma abordagem econômica, atendendo às restrições financeiras comuns entre as empresas brasileiras, podendo inclusive ser comercializado a um baixo custo. O sistema pode ser hospedado em um provedor de nuvem e dessa maneira ficar disponível para o público da Internet.

1.5 MÉTODO DE TRABALHO (ARQUITETURA, FERRAMENTAS, TECNOLOGIAS APLICADAS)

Neste capítulo, serão apresentados o método de trabalho, a arquitetura geral do sistema e as tecnologias aplicadas durante o desenvolvimento do projeto. A metodologia utilizada foi a de prototipação, uma abordagem iterativa que permite a criação de protótipos funcionais do sistema para testar e refinar suas funcionalidades. Essa técnica guia a evolução do projeto conforme as necessidades identificadas, permitindo que ocorra “uma colaboração estreita entre desenvolvedores e usuários finais, garantindo que o sistema atenda às necessidades reais.” (BØDKER,1991).

No início da fase de prototipagem, deve-se compreender tanto as necessidades dos usuários quanto os requisitos funcionais do sistema de geração de orçamentos. Essa compreensão proporciona a base para a definição das principais funcionalidades que serão incorporadas ao protótipo. “A única maneira de realmente saber se uma ideia é razoável é testá-la. Crie um protótipo rápido ou uma maquete de cada solução em potencial.” (NORMAN, 2013). O desenvolvimento do protótipo funcional envolve a criação de uma representação da interface do sistema e de suas principais funcionalidades. Focalizando especificamente nos pontos críticos do fluxo de geração de orçamentos, como o preenchimento de informações, cálculos automatizados e a geração de arquivos PDF, a construção do protótipo é realizada por meio da utilização de tecnologias como HTML, CSS e JavaScript, viabilizando a criação de uma interface interativa.

Uma vez criado o protótipo, o próximo passo envolve a sua apresentação aos *stakeholders* (partes interessadas), incluindo tanto os usuários finais quanto os clientes. Nesse estágio, eles têm a oportunidade de interagir com o protótipo e compartilhar suas impressões a respeito da usabilidade, *design*, fluxo de interação e funcionalidades oferecidas. Esse processo de *feedback* permite a identificação de áreas que requerem ajustes para assegurar a conformidade do protótipo com as expectativas. Com base nas observações e avaliações dos *stakeholders*, o protótipo é aprimorado de maneira iterativa. Essa fase envolve alterações na interface, ajustes no fluxo de trabalho e aprimoramento das funcionalidades, todos voltados a alinhar o protótipo às necessidades e desejos dos usuários.

Após diversas iterações de refinamento, o protótipo é submetido a uma validação final junto aos *stakeholders*. Nesse momento, é assegurado que todas as funcionalidades críticas

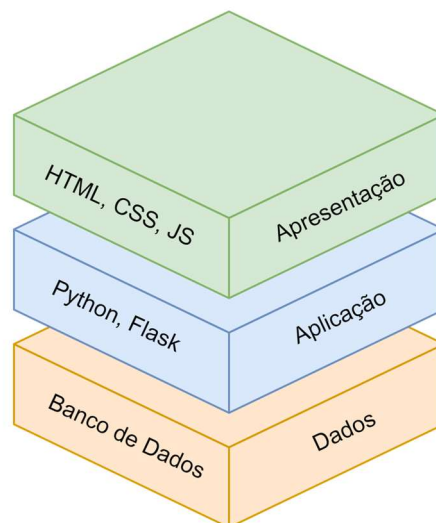
estão adequadamente representadas e que o fluxo de geração de orçamentos se apresenta de forma intuitiva e eficaz. Uma vez que o protótipo recebe a validação e aprovação dos *stakeholders*, inicia-se o processo de implementação completa do sistema web integrado. Utilizando as tecnologias previamente planejadas, como Flask, Jinja, MongoDB, entre outras, a solução final é desenvolvida com base nas lições e melhorias derivadas das iterações do protótipo. Com a implementação finalizada, uma rodada de testes extensivos é conduzida para garantir que todas as funcionalidades do sistema operem conforme o esperado. Após a aprovação desses testes, o sistema está pronto para ser disponibilizado aos usuários finais.

A prototipagem, ao incorporar um ciclo contínuo de desenvolvimento, *feedback* e refinamento desde as etapas iniciais do projeto, se apresenta como uma abordagem eficaz para a obtenção de retornos valiosos dos usuários. Esse processo minimiza o risco de erros e ajustes onerosos durante as fases finais de desenvolvimento. A aplicação dessa abordagem foi vantajosa no caso do sistema web para geração de orçamentos, onde a usabilidade e a funcionalidade estão intrinsecamente ligadas ao êxito do projeto.

1.5.1. ARQUITETURA

O sistema de geração de orçamentos foi projetado em uma arquitetura em camadas, visando a modularidade e a escalabilidade. A arquitetura consiste em três camadas principais: apresentação, aplicação e dados (Figura 1), detalhadas a seguir.

Figura 1. Arquitetura de sistema em camadas.



Adaptado de (LOGI, 2021).

A camada de Apresentação engloba a interface de usuário acessada por navegadores web. Essa camada foi implementada utilizando HTML, CSS e Javascript para fornecer uma experiência amigável e responsiva aos usuários. Foi considerada a versão mais recente de HTML (5.0), a folha de estilos foi desenvolvida a partir do zero e o Javascript utilizado nas páginas teve por objetivo tornar mais fluida e intuitiva a navegação, e para realizar validações, tornando o sistema mais seguro.

A camada de Aplicação hospeda a lógica de negócios do sistema e foi desenvolvida usando a linguagem de programação Python, com o *framework* Flask. A utilização do Flask permitiu a criação de rotas para diferentes funcionalidades, como autenticação de usuários, geração de orçamentos e manipulação de dados do banco.

Por fim, na camada de Dados o sistema utiliza o banco de dados não relacional MongoDB. Foram criadas duas coleções principais: uma para armazenar informações de *login* dos usuários e outra para guardar os detalhes de cada geração de orçamento, incluindo informações de cliente, empresa, seções e itens.

1.5.1. FERRAMENTAS

As tecnologias gratuitas se encontram presentes no cotidiano dos desenvolvedores, trazendo flexibilidade e eficiência para o processo de desenvolvimento. Uma menção notável à flexibilidade é o Visual Studio Code (VSCode), que se destaca como uma ferramenta indispensável no presente desenvolvimento do projeto de geração de orçamentos. O VSCode não apenas oferece uma interface moderna e amigável, mas também suporta simultaneamente várias linguagens de programação. No contexto do projeto de geração de orçamentos, onde a geração de PDFs e o gerenciamento de dados de clientes e itens são fundamentais, a flexibilidade do VSCode permite que desenvolvedores trabalhem com HTML, CSS, JavaScript e Python (utilizado para a criação do PDF via Flask) de maneira integrada. Isso resulta em um ambiente de desenvolvimento coeso, onde diferentes componentes do sistema podem ser abordados simultaneamente, aumentando a eficiência e a coesão do projeto.

No contexto do desenvolvimento do projeto de geração de orçamentos, o MongoDB emerge como uma ferramenta relevante devido a sua natureza como banco de dados não relacional. Ao contrário dos bancos de dados tradicionais, o MongoDB oferece uma abordagem flexível e escalável para o armazenamento de dados. No projeto de geração de orçamentos,

onde a estrutura de dados pode variar e evoluir ao longo do desenvolvimento, o MongoDB é apropriado por permitir que os desenvolvedores modelem e reestruturem os dados com agilidade. A capacidade de armazenar informações em documentos sem a rigidez de um esquema fixo simplifica a adaptação às mudanças nos requisitos do sistema, proporcionando uma eficiência significativa no desenvolvimento e na evolução do projeto.

Outro ponto positivo no contexto desse trabalho é que o MongoDB é altamente escalável, o que o torna ideal para lidar com o crescimento dos dados ao longo do tempo, um aspecto crítico em um projeto de geração de orçamentos que pode acumular uma variedade de informações de clientes, empresas e itens ao longo do tempo. Em última análise, a natureza flexível e escalável do MongoDB contribui para uma implementação mais suave do sistema de geração de orçamentos, oferecendo uma infraestrutura sólida para armazenamento e gerenciamento de dados.

1.5.2. TECNOLOGIAS APLICADAS

No decorrer do processo de desenvolvimento, diversas tecnologias foram empregadas para assegurar tanto a funcionalidade quanto a segurança do sistema. Neste trabalho de conclusão de curso foi utilizada a linguagem Python para o desenvolvimento do sistema e o banco de dados MongoDB para o armazenamento de dados, tendo sido a interface criada para a plataforma Web. O restante desta seção apresenta em detalhes cada uma dessas tecnologias.

A linguagem de programação Python foi selecionada como a base para a implementação da lógica de negócios. Por meio de sua sintaxe clara e versátil, Python viabilizou a criação de um código coeso e eficiente. Juntamente com o *framework* Flask, foi possível conceber uma aplicação web robusta, caracterizada pela modularidade e adaptabilidade.

No contexto da interface de usuário, as tecnologias HTML, CSS e JavaScript assumiram um papel essencial. Esses componentes, quando empregados em conjunto, permitiram a criação de uma interface interativa e de alta usabilidade. Através de HTML, a estrutura da página foi moldada; o CSS, por sua vez, conferiu estilo e design às páginas; e o JavaScript possibilitou a criação de interações dinâmicas e responsivas.

A integração do motor de *template* Jinja ao Python, por meio do *framework* Flask, foi responsável pela geração dinâmica de páginas HTML. Tal funcionalidade foi explorada para

inserir informações oriundas do banco de dados nos *templates*, conferindo dinamismo e personalização às páginas exibidas aos usuários.

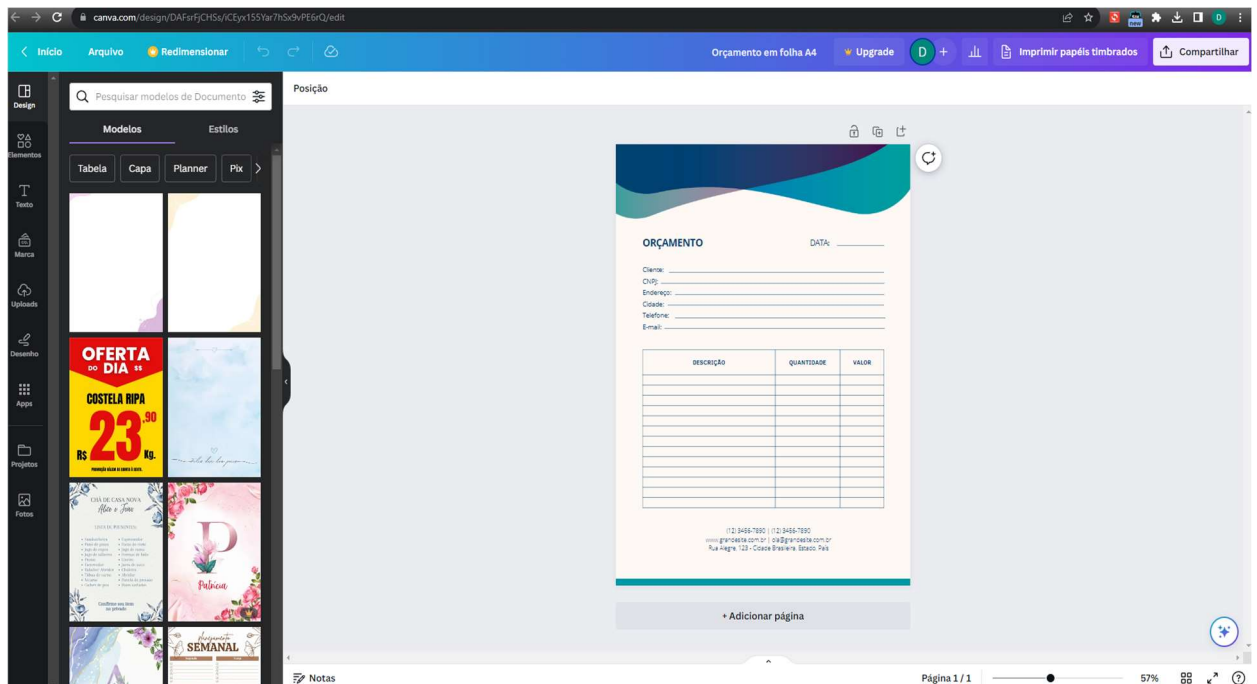
Para a gestão dos dados do sistema, a escolha recaiu sobre o banco de dados não relacional MongoDB. Essa decisão foi motivada pela capacidade do MongoDB em armazenar informações de maneira eficiente e flexível, características essenciais para um sistema voltado à geração de orçamentos, que lida com variados tipos de dados.

A combinação dessas tecnologias resultou na criação de um sistema que se alinha com as exigências das pequenas e médias empresas do setor manufatureiro. Assim, o processo de elaboração de orçamentos foi simplificado de maneira acessível e altamente eficaz, permitindo que as empresas atendam suas necessidades com agilidade e precisão.

4 2 TRABALHOS CORRELATOS

No cenário atual, não se identifica uma solução específica no mercado voltada exclusivamente para a geração de orçamentos com seções, itens e edição simplificada com banco de dados. As alternativas disponíveis muitas vezes se aproximam do objetivo, como é o caso de ferramentas de criação de *templates*, a exemplo do Canva (Figura 2). Contudo, tais ferramentas geralmente possuem foco diversificado e abrangente, não proporcionando a mesma amplitude de funcionalidades e personalização esperada em um sistema dedicado à geração de orçamentos para pequenas e médias empresas do setor de manufatura.

Figura 2 – Site Canva



(CANVA, 2023)

Outro site que pode ser citado como exemplo é o site Gerador de Orçamento PDF (Figura 3). Há alguns campos de itens disponíveis para o uso em geração de orçamentos, porém é pouco intuitivo, não possui login para diferenciar usuários e seu orçamento é salvo apenas em *cache*, não sendo possível ter acesso aos orçamentos gerados a partir de outros dispositivos. Após gerar o PDF é verificado que também não existe um campo de assinatura de aceite do cliente e da empresa, não havendo a possibilidade do cliente que está recebendo o orçamento e da empresa terem um documento em mãos com o aceite de ambas as partes. Outro ponto bem importante é a formatação das informações no cabeçalho, em que não sabemos o que é cada informação por conta da falta de um *label* especificando a informação, como “Telefone:”, “CNPJ:”, entre outros, ao gerar o orçamento (Figura 4).

Figura 3 – Gerador de Orçamento PDF

[Termos de Uso](#)

Endereço

Itens

Quantidade

Descrição

R\$ 0,00

Adicionar

08/09/2023

10/09/2023

Teste Geração

R\$ 1.500,00

R\$ -1500.00

Editar Gerar PDF Arquivar

© 2023 Gerador de Orçamento

(GERADORDEORCAMENTO, 2023)

Figura 4 – Orçamento gerado

Orçamento

Emissão 08/09/2023
Validade 10/09/2023

Empresa
Teste
121321231231
Contato
(11) 1123-4567
123@gmail.com

Cliente
Teste Cliente
Contato
(47) 9999-7888
sadasoidal@gmail.com

Produto/Serviço	Qty.	Val. Unit.	Total
Produto3	3	R\$ 100.00	R\$ 300.00
Produto2	2	R\$ 600.00	R\$ 1200.00
Produto1	1	R\$ 500.00	R\$ 500.00

Total de produtos 6

Sub-total R\$ 2000.00

Desconto R\$ 1500.00

Total R\$ 500.00

Observações
Teste Geração

(GERADORDEORCAMENTO, 2023)

Por fim, se encontra disponível outra opção para geração de orçamentos no site Wibushi (Figura 5). O Wibushi é muito mais complexo do que os sistemas comentados anteriormente, tirando do foco o intuito principal da presente proposta, que é prover simplicidade e interface intuitiva. O gerador de orçamentos com *templates* desse site deixa ainda mais complexa a operação, apesar de fornecer um recurso interessante que é o envio de e-mail com a proposta, dentre outras funções.

Figura 5 – Gerador Wibushi



(WIBUSHI, 2023)

Para explicitar as diferenças entre os trabalhos similares e a presente proposta, foi criada uma tabela comparativa entre os sistemas apresentados anteriormente e o sistema apresentado neste trabalho (Tabela 1). A tabela é dividida por funcionalidades e focos principais de cada sistema, trazendo pontos positivos e comparações quanto a algumas características. No cabeçalho da tabela está o nome dos sistemas que estão sendo comparados e na lateral esquerda da tabela estão as funcionalidades que estão sendo comparadas; no centro da tabela existem as marcações para cada funcionalidade vinculada aos sistemas, sendo elas “X” para o sistema que não adere a funcionalidade e “Ok” para o sistema aderente à funcionalidade. Todos os sistemas considerados são para a plataforma Web e possuem um mecanismo de edição rápida, isto é, uma forma de recuperar as informações de orçamentos anteriores, podendo realizar alterações e gerar um novo documento. Desta forma há uma “reciclagem” de informação para o usuário, que não precisa inserir todos os campos novamente.

Tabela 1 - Comparação dos Sistemas de Geração de Orçamento

Gerador Genérico PDF (Apresentado)	CANVA	GERADORDEORCAMENTO	WIBUSHI
Ok	Ok	Cache	Ok
Ok	X	Ok	X
Ok	Ok	X	Ok
Ok	Ok	Ok	Ok
X	Ok	X	Ok
Ok	Ok	X	Ok

Fonte: o autor.

Pode-se observar na tabela que o sistema proposto (Gerador Genérico PDF) possui todas as funcionalidades consideradas, com exceção do uso de *Templates*. O uso de um orçamento genérico inicial já resolve o problema inicial do cliente, e o reaproveitamento de orçamentos anteriores funciona como uma espécie de *Template*, provendo informações já existentes em novos orçamentos. Não é propósito deste trabalho fornecer diversos modelos de orçamento, mas um modelo padronizado que seja prático e eficiente. Para o presente trabalho foram considerados aspectos prioritários, como o armazenamento de informações separado por usuários por meio do *login*, a integração com banco de dados e a edição rápida. Além disso, desde a concepção até os últimos detalhes de implementação o aspecto de simplicidade de operação esteve presente, guiando todo o processo de desenvolvimento.

3 REQUISITOS

Os requisitos em um projeto de desenvolvimento de software são as especificações detalhadas das funcionalidades, comportamentos e características que o sistema deve apresentar. Eles atuam como um guia para orientar o processo de *design*, implementação e teste do sistema, garantindo que o resultado final atenda às expectativas dos usuários e cumpra os objetivos do projeto. Os requisitos podem ser classificados em três categorias principais:

requisitos de usuário, requisitos funcionais e requisitos não funcionais (SOMMERVILLE, 2019).

Requisitos de usuário, também conhecidos como requisitos de alto nível, são declarados na linguagem dos usuários finais e descrevem o que os usuários esperam do sistema. Eles focam nas necessidades e objetivos dos usuários, muitas vezes sem detalhar a implementação técnica. Os requisitos de usuário são vitais para criar um sistema que verdadeiramente atenda às demandas e expectativas dos usuários.

Requisitos funcionais descrevem as funcionalidades específicas que o sistema deve oferecer. Eles definem o que o sistema deverá fazer em termos de entradas, saídas e comportamento esperado. Geralmente, são descritos em termos de ações que o sistema deve realizar em resposta a determinados eventos: "requisitos representam uma comunicação formal e compartilhada, necessária entre os *stakeholders* para entender as necessidades, limitações e oportunidades do sistema em desenvolvimento." (SOMMERVILLE, 2019). Requisitos não funcionais dizem respeito às características não diretamente ligadas às funcionalidades específicas do sistema, mas que impactam sua qualidade e desempenho. Isso inclui aspectos como desempenho, segurança, usabilidade, escalabilidade, entre outros.

3.1 REQUISITOS DE USUÁRIO

São descritos a seguir 13 requisitos de usuário que abordam a geração de PDF, gerenciamento de usuários, captura e manutenção dos dados e organização do documento, facilidade e rapidez na geração e interatividade com o sistema.

RU01: Oferecer acesso intuitivo e simples via celulares e notebooks.

RU02: Permitir a geração de orçamentos em formato PDF.

RU03: Incluir assinaturas do cliente e da empresa no PDF.

RU04: Exibir de forma clara a data de geração do orçamento.

RU05: Dispor de interface que apresente informações detalhadas do cliente e da empresa.

RU06: Capacitar a inserção de múltiplos itens em diferentes seções.

RU07: Proporcionar fácil acesso à lista de orçamentos.

RU08: Viabilizar a seleção e edição de orçamentos existentes.

RU09: Garantir atualização eficiente das informações nos orçamentos.

RU10: Permitir acesso descomplicado à lista de orçamentos para seleção de reimpressão.

RU11: Realizar geração veloz e eficaz de PDFs para reimpressão.

RU12: Disponibilizar notificação clara e eficiente em caso de bloqueio e desbloqueio.

RU13: Assegurar recuperação segura de senha em cenários de bloqueio.

3.2 REQUISITOS FUNCIONAIS

A implementação bem-sucedida dos requisitos funcionais a seguir desempenha um papel crucial na eficácia e na utilidade do sistema de gestão de orçamentos proposto. Em conjunto, esses requisitos funcionais garantem que o sistema seja seguro, eficiente e amigável, atendendo às necessidades tanto dos clientes quanto das empresas na gestão de orçamentos.

RF01: Implementar sistema de autenticação de usuários.

RF02: Desenvolver interface de usuário para preenchimento de informações.

RF03: Possibilitar a geração de PDF com todas as informações relevantes.

RF04: Incluir campo no PDF para assinatura do cliente e da empresa.

RF05: Exibir claramente a data de geração do orçamento no PDF.

RF06: Automatizar cálculos de valores.

RF07: Dividir o processo em seções para inserção de múltiplos itens.

RF08: Realizar validação das informações antes da geração do PDF.

RF09: Oferecer visualização da lista completa de orçamentos.

RF10: Permitir a seleção e edição de orçamentos existentes.

RF11: Habilitar a atualização das informações nos orçamentos.

RF12: Garantir recálculo automático dos valores nos orçamentos editados.

RF13: Disponibilizar opção para seleção e reimpressão de orçamentos existentes.

RF14: Implementar mecanismo de contagem de tentativas de login.

RF15: Realizar bloqueio automático após um número predefinido de tentativas malsucedidas.

RF16: Dispor de um processo de solicitação de desbloqueio.

3.3 REQUISITOS NÃO-FUNCIONAIS

A incorporação dos requisitos não funcionais a seguir é importante para o sucesso e a eficiência do sistema de gestão de orçamentos proposto. Esses requisitos não funcionais garantem que o sistema seja robusto e de fácil utilização, atendendo às expectativas dos usuários e aos padrões de qualidade estabelecidos.

RNF01: Assegurar plataforma web responsiva para acesso por meio de celulares e notebooks.

RNF02: Criar interface intuitiva e de fácil utilização.

RNF03: Suportar múltiplos usuários simultâneos.

RNF04: Realizar a geração eficiente de arquivos PDF.

RNF05: Garantir a segurança das informações dos clientes.

RNF06: Possibilitar navegação fluída para acesso à lista de orçamentos.

RNF07: Manter integridade das informações durante a edição.

RNF08: Prover atualização dinâmica dos valores nos orçamentos.

RNF09: Alcançar velocidade na geração ágil de PDFs para reimpressão.

RNF10: Implementar segurança na recuperação de senhas e no desbloqueio.

RNF11: Fornecer notificações eficazes ao administrador em caso de bloqueio e desbloqueio.

RNF12: Assegurar rapidez na geração de novas senhas e no processo de desbloqueio.

4 DIAGRAMAS UML

O Diagrama de Modelagem de Linguagem Unificada (UML) é uma linguagem visual utilizada para modelar, especificar, visualizar e documentar sistemas de software. Ele oferece uma variedade de diagramas que representam diferentes aspectos de um sistema, permitindo uma comunicação eficiente entre os membros da equipe de desenvolvimento, analistas, designers e stakeholders. O UML é amplamente adotado na engenharia de software para capturar a estrutura, comportamento, entre outros elementos cruciais de sistemas complexos:

"O UML é uma linguagem visual que oferece uma maneira padrão de visualizar o design de um sistema. Ela permite que desenvolvedores criem diagramas claros e concisos que podem ser compartilhados entre diferentes equipes e stakeholders, auxiliando na compreensão e comunicação de conceitos complexos." (FOWLER, 2003)

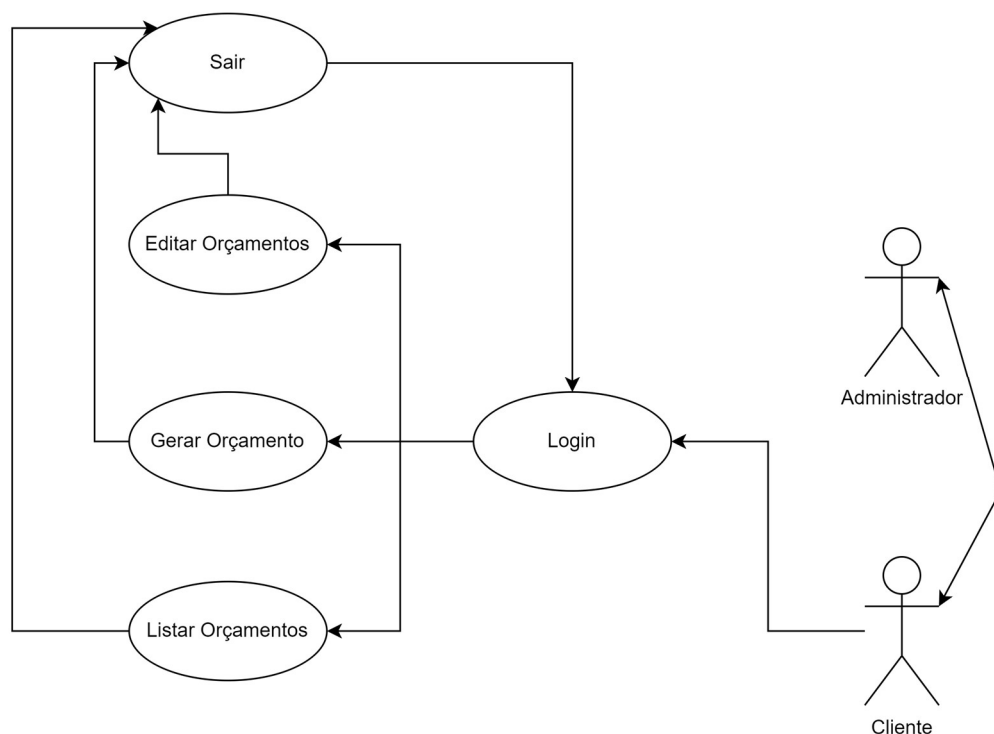
4.1 DIAGRAMAS DE CASO DE USO

O caso de uso é uma ferramenta da engenharia de software que auxilia na representação das interações funcionais entre os atores (usuários ou sistemas externos) e o sistema em desenvolvimento. Ele captura os cenários de uso, descrevendo as ações que os atores executam e as respostas do sistema. Essa abordagem fornece uma compreensão clara dos requisitos e comportamentos do sistema a partir da perspectiva dos usuários.

"O caso de uso é uma forma poderosa e eficiente de capturar e comunicar as interações entre atores e sistemas. Ele facilita a comunicação entre diferentes partes interessadas, permitindo uma compreensão compartilhada dos objetivos e funcionalidades do sistema." (LARMAN, 2007).

Um caso de uso que envolve os usuários do presente sistema (administrador e cliente) ilustra de forma concisa o fluxo completo que engloba a compra do sistema, a liberação do acesso pelo administrador e as atividades executadas pelo cliente após a liberação (Figura 6). Tem-se, então, uma visão clara de como o sistema atende as necessidades do usuário final e de como as operações se interconectam para oferecer uma experiência integrada.

Figura 6 - Diagrama de Casos de Uso.

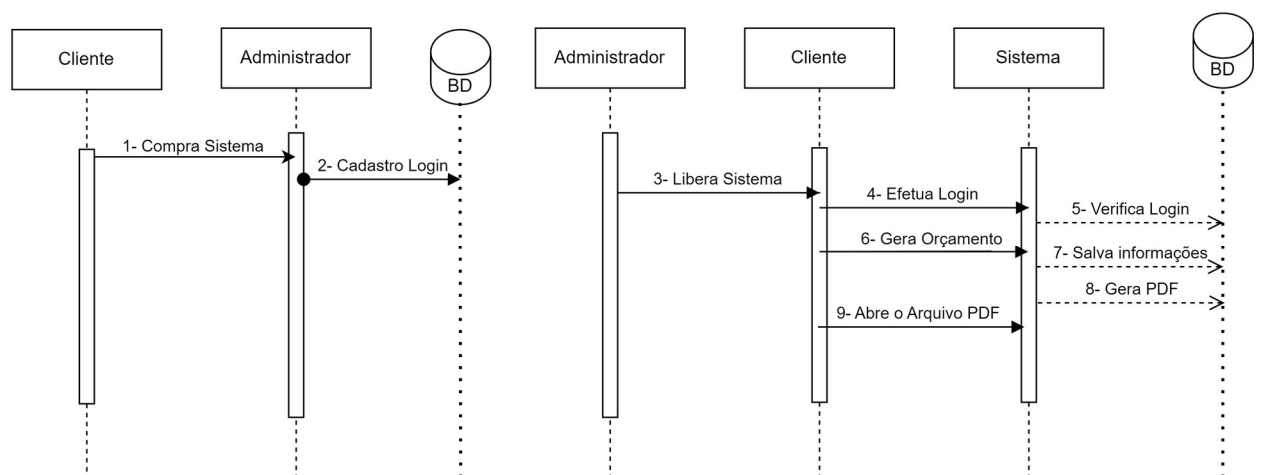


Fonte: o autor.

4.1 DIAGRAMA DE SEQUÊNCIA

O Diagrama de Sequência é uma ferramenta gráfica da UML que representa a interação temporal entre os objetos do sistema ao longo do tempo. Ele descreve a ordem das mensagens trocadas entre os objetos, ajudando a ilustrar como os componentes do sistema colaboram para realizar um cenário específico. No sistema genérico de orçamentos, o diagrama de sequência ilustra de forma clara as interações e a ordem das mensagens entre os atores e o sistema, proporcionando uma visão detalhada de como o fluxo de trabalho se desenvolve (Figura 1). O processo se inicia na solicitação do sistema por compra ou por aquisição estratégica (ação 1) como exemplo período de teste do sistema; após a solicitação, o administrador ou dono do sistema realiza o cadastro das credenciais do cliente no banco de dados (ação 2) e libera o acesso ao usuário (ação 3). No acesso via login (ação 4), o usuário passa pela criação (ação 6), preenchendo os campos do orçamento, persistindo os dados (ação 7) e concluindo o processo com a geração do arquivo PDF (Figura 7). O diagrama ajuda a entender a dinâmica do sistema e como os diferentes componentes colaboram para atingir os objetivos do usuário.

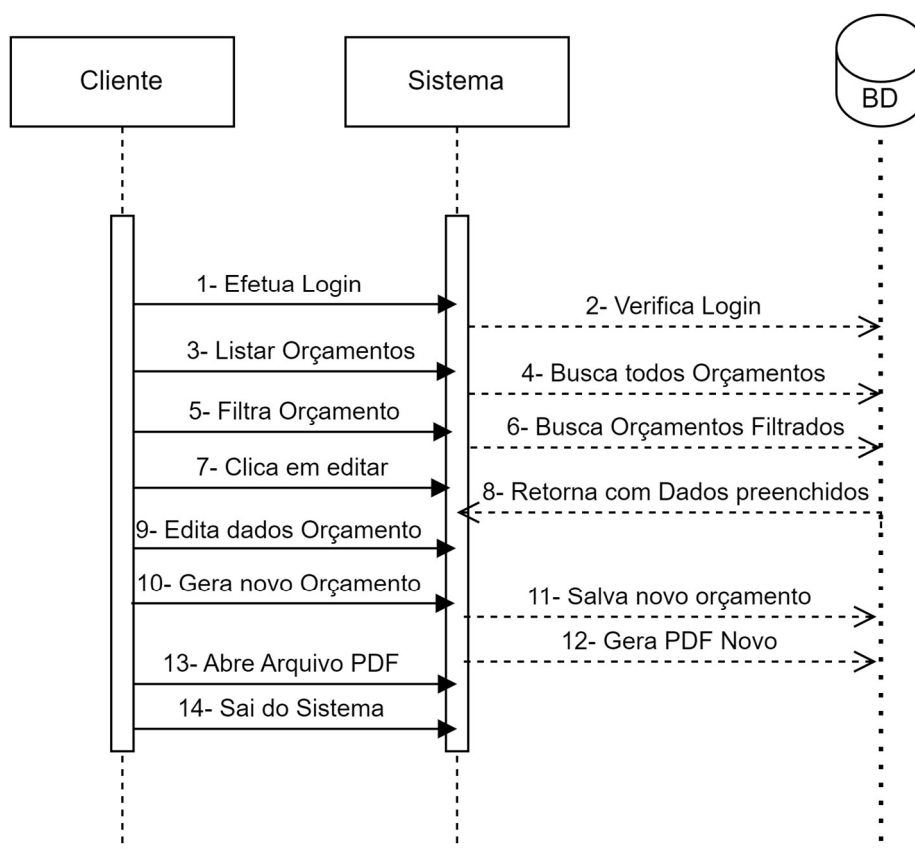
Figura 7 - Diagrama de Sequência Processo de geração de orçamento.



Fonte: o autor.

Após a geração do orçamento é possível utilizar o fluxo secundário do sistema, de maneira que, após o primeiro orçamento gerado, é possível utilizar o método de edição de orçamentos que já foram gerados (Figura 8). Essa ação está disponível após o login no sistema (ação 1), via ação de listagem de orçamentos (ação 3), na qual serão listados todos os orçamentos gerados até o momento. É possível aplicar um filtro (ação 5), fazendo com que a tela seja atualizada e exiba ao usuário apenas os orçamentos com o título referente a pesquisa descrita (ação 6). Em seguida, o usuário pode editar o orçamento que deseja (ação 7), levando o sistema a retornar à página de geração de orçamento com todos os dados do orçamento preenchidos. Isso torna possível a edição destes dados e possibilita a geração de um novo orçamento que será salvo no banco, sem a necessidade de digitar todos os campos novamente.

Figura 8 - Diagrama de Sequência Processo de Listar e Editar Orçamentos.



Fonte: o autor.

5 5 **MODELAGEM DE DADOS**

A modelagem de dados é o processo de representar as estruturas de dados de um sistema de forma organizada e ilustrativa. Ela envolve a criação de modelos que definem como os dados estão organizados, relacionados e armazenados em um sistema. A modelagem de dados auxilia na compreensão, comunicação e desenvolvimento de sistemas de informação, garantindo a integridade e a eficiência do armazenamento e manipulação dos dados.

No contexto de bancos de dados não relacionais, a modelagem de dados difere do modelo tradicional de bancos de dados relacionais. “Enquanto os bancos de dados relacionais utilizam tabelas e relacionamentos rígidos, os bancos de dados não relacionais empregam estruturas mais flexíveis para acomodar tipos variados de dados.” (BRADSHAW, 2020). No banco de dados utilizado no presente trabalho, empregam-se os conceitos de coleções, em vez de tabelas.

5.1 **COLEÇÃO DE DADOS**

“Uma coleção é um grupo de documentos JSON existentes em uma base de dados.” (FROZZA, 2018). A Coleção é uma representação visual de como os dados são organizados em uma coleção de um banco de dados não relacional, como o MongoDB. Ele fornece uma visão geral da estrutura de dados e dos atributos presentes em cada documento dentro da coleção. Embora seja menos formal do que um diagrama ER em bancos de dados relacionais, a coleção ajuda a compreender a disposição dos dados e a relação entre os atributos. Considerando o exemplo de uma estrutura que armazena informações sobre o login de um usuário, cada campo da coleção "login_usuarios" é descrito de maneira mais detalhada, com exemplos de valores a seguir (Figura 9):

- `_id`: Campo gerado automaticamente pelo MongoDB para representar a chave primária do documento.
- `senha`: Senha criptografada do usuário, armazenada como uma string. Utiliza a biblioteca `bcrypt` do Python para gerar a senha e um valor de salt único.
- `situacao`: Status da conta do usuário. Pode ser "OK" para ativa ou "NOK" para bloqueada após 3 tentativas de login malsucedidas.
- `login`: Nome de usuário usado para fazer login no sistema.
- `salt`: Valor único utilizado na geração da senha criptografada.
- `tentativas`: Número de tentativas de login malsucedidas. Quando atinge 3 tentativas, a situação da conta é alterada para "NOK".

Figura 9 - Diagrama de Coleção MongoDB.

```

{
  "collection": "login_usuarios",
  "_id": {
    "$oid": "64d3c5c2643e6d70f925e80b"
  },
  "senha": {
    "description": "Senha Criptografada do Usuario",
    "type": "String"
  },
  "situacao": {
    "description": "Status da conta, 'OK' e 'NOK' quando excede tentativas > 3 ",
    "type": "String"
  },
  "login": {
    "description": "Nome de usuario",
    "type": "String"
  },
  "salt": {
    "description": "valor unico para criptografia com a chave",
    "type": "String"
  },
  "tentativas": {
    "type": "Int"
  }
}

```

Fonte: o autor.

6 6 PROJETO DA INTERFACE

Ao apresentar um projeto de interface focado em botões de fácil acesso e usabilidade, prioriza-se a criação de uma experiência intuitiva para pessoas com dificuldades de acesso à tecnologia. Nesse contexto, a identificação das principais ações na interface é crucial, com uso de botões grandes e espaçados, que possuem cores de alto contraste e fontes legíveis (Figura 10). Essa abordagem foi realizada em tarefas essenciais, como "Gerar Orçamento" e "Listar Orçamentos Gerados", para que essas operações sejam rapidamente acessíveis, promovendo a inclusão e a acessibilidade digital.

Figura 10 - página gerar_orcamento.html “seções”

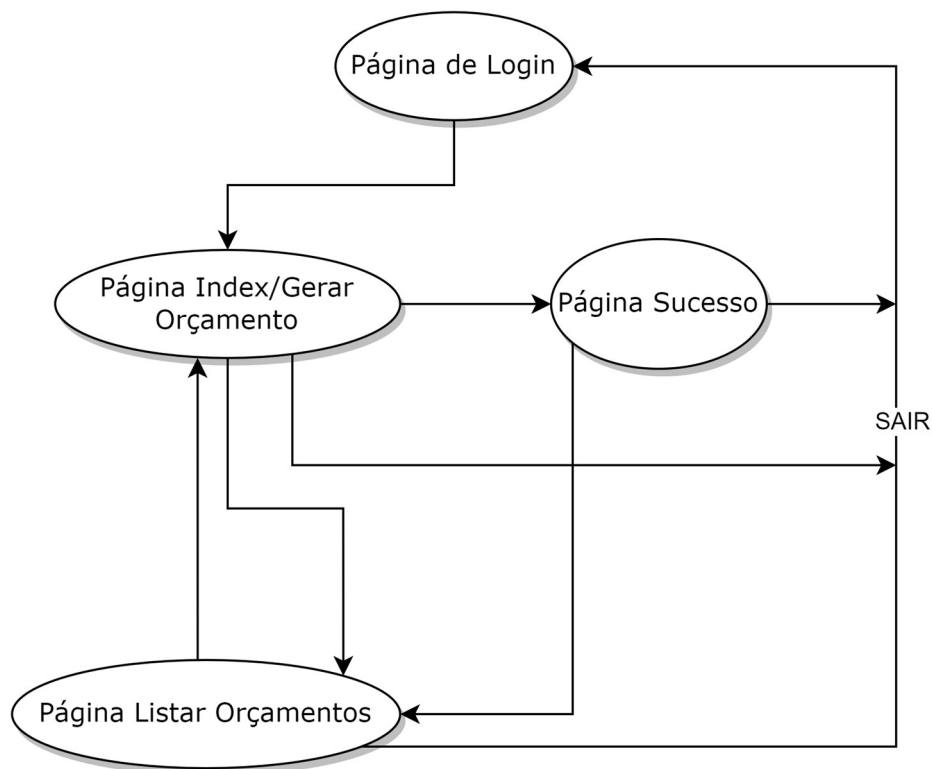
O diagrama mostra a interface de usuário para a página `gerar_orcamento.html`. O layout é dividido em seções com campos de entrada e botões de ação. As seguintes características de design são destacadas:

- Seções:** O título "Seções" está em uma caixa preta, com uma linha apontando para ele e o texto "Fonte Grande para leitura".
- Formulário:** O formulário contém quatro campos de entrada rotulados "Título da Seção:", "Nome do Item:", "Descrição do Item:" e "Valor do Item:". O campo "Valor do Item:" possui um exemplo de texto: "Ex.: 100 ou 100.50".
- Botões de Ação:**
 - O botão "Remover Item" é destacado por uma caixa preta e uma linha apontando para ele com o texto "Coloração intuitiva".
 - O botão "Adicionar Item" é amarelo.
 - O botão "Remover Seção" é marrom.
 - O botão "Adicionar Seção" é uma barra amarela larga.
 - O botão "Gerar Orçamento" é verde e está em uma caixa preta, com uma linha apontando para ele e o texto "Botões grandes, fácil leitura".

Fonte: o autor.

Um mapa de navegação claro e uma opção consistente de retorno ao login permitem a criação de uma experiência de usuário coesa, eficiente e confiável em um sistema. Uma boa navegação ajuda os usuários a se movimentarem pelo sistema de forma intuitiva, minimizando erros, promovendo a descoberta de recursos e facilitando a interação com as diferentes partes do sistema. Destaca-se a importância de um *design* funcional que atenda às necessidades práticas dos usuários, independentemente de suas habilidades tecnológicas (WILLIAMS, 2013). Sendo assim, um mapa de navegação foi elaborado para o sistema de geração de orçamentos (Figura 11). O sistema inicia pela abertura da página de login, a partir da qual tem-se acesso às páginas de criação e edição de orçamento e a página de listagem e consulta de orçamentos. Uma página específica é utilizada para mostrar mensagens de sucesso das operações.

Figura 11. Mapa de Navegação do sistema.



Fonte: Autor.

A página "listar_orçamentos.html" foi projetada com o intuito de oferecer uma visualização simplificada e amigável dos orçamentos previamente gerados (Figura 12). O foco está em facilitar a experiência do cliente ao proporcionar uma organização eficaz e uma busca intuitiva. Os orçamentos são exibidos de forma ordenada pela data de geração, permitindo ao cliente identificar rapidamente as informações mais recentes. Além disso, a funcionalidade de busca, implementada por meio de um campo de pesquisa, oferece ao cliente a capacidade de localizar orçamentos específicos pelo nome. Esse recurso agiliza a localização de informações relevantes em uma lista crescente de orçamentos.

Figura 12 - Página listar_orcamentos.html



Fonte: o autor.

Lembrando a abordagem de *design* centrada no usuário, os botões são mantidos grandes e de fácil localização, alinhados com o conceito de acessibilidade. Esses botões são projetados para tornar as ações como "Visualizar Detalhes", "Editar" ou "Imprimir" claramente identificáveis e de fácil alcance. Para manter o *design* limpo e sem sobrecarregar a tela, também foi incorporada à tela de listagem o mecanismo de paginação de registros (Figura 13). Com a paginação, uma quantidade gerenciável de orçamentos é exibida por vez, permitindo uma navegação mais fluida entre as diferentes páginas de resultados e evita o excesso de informações em tela, garantindo a clareza e a eficácia na apresentação dos dados.

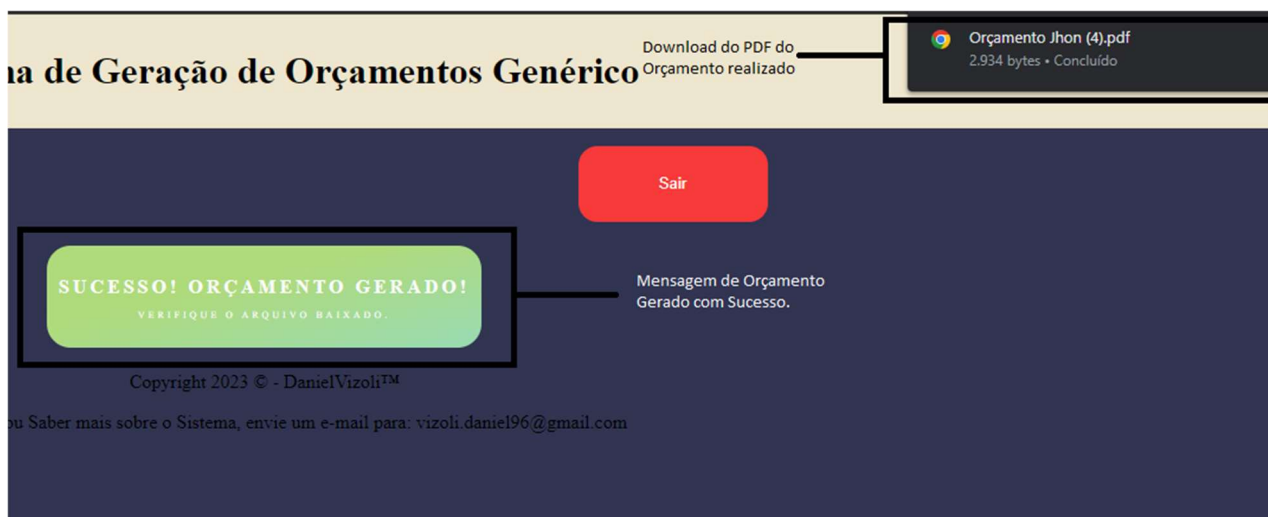
Figura 13 - Paginação listar_orçamentos.html



Fonte: o autor.

A página de sucesso é cuidadosamente projetada para fornecer uma experiência gratificante ao usuário após a geração do orçamento (Figura 14). Ao completar o processo, uma mensagem de sucesso é exibida, proporcionando um *feedback* positivo e reconhecimento da ação realizada. Além disso, para oferecer ainda mais conveniência, o PDF gerado é iniciado automaticamente para *download* no navegador do usuário. Essa abordagem visa proporcionar uma transição suave do processo de geração para o acesso ao documento, facilitando o uso e melhorando a satisfação do cliente.

Figura 14 - Orçamento Gerado com sucesso



Fonte: o autor.

Quanto ao formato do orçamento gerado, ele é estruturado de maneira clara e organizada para fornecer as informações necessárias de maneira compreensível (Figura 15). O orçamento é dividido em três seções distintas: cabeçalho, corpo e rodapé. No cabeçalho, são exibidos detalhes como o título do orçamento, informações do cliente e dados da empresa prestadora de serviços. O corpo do orçamento contém os itens que estão sendo orçados, organizados em seções, com cada item contendo sua descrição e valor correspondente.

Dentro do corpo do orçamento, procura-se manter a simplicidade e a clareza, permitindo que o usuário entenda facilmente o que está sendo orçado. O total do orçamento é apresentado no final, somando os valores de todos os itens. No rodapé do documento, campos de assinatura são disponibilizados para o cliente e a empresa, além da data de geração do orçamento.

Uma descrição também é incluída para informar a validade do orçamento, proporcionando informações relevantes para o cliente. Essa abordagem de *design* busca garantir que o orçamento gerado seja visualmente agradável, organizado e de fácil compreensão, proporcionando uma experiência satisfatória ao usuário e transmitindo profissionalismo na comunicação com o cliente.

Figura 15 – Orçamento gerado em PDF

Orçamento Jhon

Cliente

Nome:	João	Telefone:	11 984808370
Email:		CPF/CNPJ:	

Empresa

Nome:	lindomar marcenaria		
Telefone:	11 984808370	E-mail:	lindomarmarcenaria@hotmail.com

Itens Orçados

cozinha

Item	Descrição	Valor
gabinete inferior	Orçamento Orçamento Orçamento Orçamento Orçamento Orçamento Orçamento Orçamento Orçamento Orçamento	R\$ 100.00
movel geladeira	Orçamento Orçamento Orçamento Orçamento Orçamento	R\$ 300.00
móvel sumerior	Orçamento Orçamento Orçamento Orçamento	R\$ 500.00

sala

Item	Descrição	Valor
painel tv		R\$ 600.00
rak		R\$ 800.00

Total: R\$ 2300.00

Assinatura do Cliente: _____

Assinatura da Empresa: _____

Data do Orçamento: 28/08/2023
Orçamento válido por 7 dias a partir da data de geração.

Fonte: o autor.

7 7 IMPLEMENTAÇÃO

Para o desenvolvimento do sistema foi utilizada a linguagem Python. No contexto de sistemas web, foi utilizado o Flask, um framework web minimalista, eficaz para o desenvolvimento web em Python. O Flask se destaca pela simplicidade, utilizando a abordagem baseada em rotas para criar aplicativos web (GRINBERG, 2018). No sistema gerador de orçamentos, decidiu-se utilizar uma abordagem procedural devido à simplicidade do projeto, visando não inserir complexidade onde não é necessário. Uma análise de linhas de código

mostrou soluções procedurais que eram significativamente mais concisas do que as versões orientadas a objetos, usando apenas metade das linhas de código (BUGGE, 2019). Além disso,

"O estilo de programação Orientada a Objetos tem um impacto maior no tamanho do código resultante do que no número de instruções executadas. Isso é razoável, pois o uso de objetos aumenta significativamente o tamanho do código por meio da definição de classes, entretanto, o tempo de execução não aumenta drasticamente, principalmente devido ao uso de métodos *inline*". (CHATZIGEORGIOU, 2002)

Dessa maneira, ao optar por uma abordagem procedural, pode-se tirar proveito da simplicidade e eficiência ao trabalhar com protótipos, fazendo com que a entrega de prototipações de sistemas possa ser feita de melhor maneira por conta da baixa complexidade do código sequencial. Isso permite que o desenvolvedor se concentre diretamente na lógica de negócios sem a necessidade de abstrações complexas, o que é especialmente benéfico em projetos menores e de prazo limitado. Os programadores procedurais e aqueles que utilizam orientação a objetos parecem usar estratégias diferentes de compreensão. O desenvolvimento de uma aplicação precisa levar em conta essas diferenças, fornecendo abordagens específicas para programadores que usam paradigmas diferentes (CYNTHIA, 2001).

As ferramentas escolhidas para o desenvolvimento já foram abordadas anteriormente (Seção 1.5.1), mas é relevante destacar a interação fluida entre o MongoDB e a linguagem Python. A capacidade do MongoDB de armazenar dados de forma flexível e escalável enquadrou-se perfeitamente com as necessidades do sistema web. Um aspecto notável da implementação foi a abordagem prototipação iterativa (Seção 1.5), que permitiu ajustar e adaptar o sistema de acordo com o *feedback* do cliente e as mudanças nas demandas. Essa flexibilidade garantiu que o sistema atendesse de maneira precisa e eficaz as necessidades do usuário.

Outro aspecto importante no presente trabalho é a preservação da propriedade intelectual e a proteção dos recursos. Para esse fim, o código fonte do projeto foi armazenado em um repositório privado no GitHub. Essa medida visa garantir a confidencialidade e a segurança do sistema desenvolvido conforme o sistema evolui para potencialmente ser comercializado no futuro.

8 RESULTADOS OBTIDOS

A implementação do sistema web de geração de orçamentos pode gerar impactos significativos para as empresas que antes dependiam de processos manuais ou ferramentas pré-prontas. Ao oferecer mobilidade e acessibilidade, é possível simplificar o processo de criação de orçamentos, proporcionando uma experiência mais ágil e eficiente para nossos usuários. Empresas que costumam gerar orçamentos no papel ou em computadores fixos podem encontrar um novo nível de flexibilidade.

O Sistema foi utilizado durante duas semanas por duas pessoas diferentes o que gerou um feedback valioso dos usuários, refletindo a eficácia do sistema. Alguns comentaram: "Agora posso enviar orçamentos para os clientes muito mais rapidamente", destacando a agilidade ganha ao utilizar a plataforma. Outro comentário importante foi: "Achei que seria mais complicado usar o sistema, mas é realmente fácil." Essa reação positiva comprova que atingimos nosso objetivo de criar um sistema amigável e intuitivo, superando as expectativas dos usuários.

Ao priorizar a usabilidade e o fluxo intuitivo, é possível tornar a tecnologia acessível mesmo para aqueles que não têm familiaridade com sistemas ou uso de tecnologias. O feedback encorajador evidencia que o sistema é uma solução bem-sucedida para atender as necessidades de empresas e indivíduos que desejam simplificar o processo de geração de orçamentos.

9 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Ao final deste trabalho foi desenvolvido um sistema web voltado para a geração de orçamentos, com um enfoque direto nas necessidades de pequenas e médias empresas, e empreendedores com pouca familiaridade com a tecnologia. Foi criada uma solução que simplifica e otimiza o processo de criação de orçamentos, oferecendo uma alternativa amigável para um público que muitas vezes enfrenta barreiras tecnológicas.

Considerando trabalhos futuros, existe a possibilidade do sistema se tornar ainda mais flexível e personalizado. O sistema poderia ser adaptado para atender as necessidades específicas de cada cliente, agregando funcionalidades personalizadas que se alinhem com os processos individuais de cada empresa. Além disso, uma ideia interessante seria criar uma plataforma online para venda de pacotes de orçamentos gerados. Isso permitiria que pequenas

e médias empresas adquirissem uma quantidade limitada de orçamentos a um custo acessível, levando em consideração a demanda reduzida em comparação com empresas maiores.

10 AGRADECIMENTOS

Gostaria de expressar minha sincera gratidão à minha esposa Isadora Vizoli, cujo apoio constante e ajuda foram inestimáveis ao longo desta jornada. Também dedico este trabalho ao meu pai, Lindomar Vizoli, que não apenas inspirou este projeto, mas também se tornou o público-alvo, fornecendo *insights* valiosos e desafios reais que enfrenta em seu cotidiano. Por fim, agradeço profundamente ao meu orientador, Hylson Vescovi Netto, por sua orientação dedicada, inúmeras reuniões construtivas e revisões que contribuíram significativamente para a conclusão deste trabalho e a conclusão bem-sucedida do curso.

10 REFERÊNCIAS

- BODKER, Susanne. Through the interface: A human activity approach to user interface design. CRC Press, 2021.
- BRADSHAW, Shannon; BRAZIL, Eoin; CHODOROW, Kristina. MongoDB: the definitive guide: powerful and scalable data storage. O'Reilly Media, 2019.
- BUGGE, ERIK. The Best of Both Worlds? Reimplementing an Object-Oriented System with Functional Programming on the .NET Platform and Comparing the Two Paradigms. Master thesis, University of Oslo, 2019.
- CHATZIGEORGIOU, Alexander; STEPHANIDES, George. Evaluating performance and power of object-oriented vs. procedural programming in embedded processors. In: International Conference on Reliable Software Technologies. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 65-75.
- CORRITORE, Cynthia L.; WIEDENBECK, Susan. An exploratory study of program comprehension strategies of procedural and object-oriented programmers. International Journal of Human-Computer Studies, v. 54, n. 1, p. 1-23, 2001.
- CANVA. Site Canva. Disponível em: <https://www.canva.com/>. Acesso em: 29 ago. 2023.
- FOWLER, Martin. UML distilled: a brief guide to the standard object modeling language. Addison-Wesley Professional, 2018.

- FROZZA, Angelo Augusto; DOS SANTOS MELLO, Ronaldo; DA COSTA, Felipe de Souza. An approach for schema extraction of JSON and extended JSON document collections. In: 2018 IEEE International Conference on Information Reuse and Integration (IRI). IEEE, 2018. p. 356-363.
- GRINBERG, Miguel. Flask web development: developing web applications with python. O'Reilly Media, Inc.", 2018.
- GERADORDEORCAMENTO. Site. Disponível em: <https://geradordeorcamento.com/>. Acesso em: 08 set. 2023.
- JOBS, Steve.BrainyQuote.com, BrainyMedia Inc , 2023.
https://www.brainyquote.com/quotes/steve_jobs_416921, accessed October 25, 2023.
- LOGI. 3-Tier Architecture: A Complete Overview. 2021. Disponível em: <https://www.jinfonet.com/resources/bi-dened/3-tier-architecture-complete-overview>. Acesso em: 27 set. 2023.
- LARMAN, C. Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientados a Objetos. Bookman Editora. 2007.
- NORMAN, Don. The design of everyday things: Revised and expanded edition. Basic books, 2013.
- SOMMERVILLE, Ian. Engenharia de Software. Pearson, 10ª Edição. 2019.
- WILLIAMS, Robin. Design para quem não é designer. Callis. 2013.
- WIBUSHI. Site. Disponível em: <https://www.wibushi.com>. Acesso em: 08 set. 2023.