



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE LICENCIATURA EM COMPUTAÇÃO E INFORMÁTICA

CLICK STUDIUM: UM SISTEMA WEB PARA SELEÇÃO DE
FOTOGRAFIAS

JERÔNIMO GABRIEL DA SILVA

Cajazeiras - PB
2024

Jerônimo Gabriel da Silva

CLICK STUDIUM: UM SISTEMA WEB PARA SELEÇÃO DE FOTOGRAFIAS

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Licenciatura em Computação e Informática na modalidade de EaD do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Licenciado em Computação e Informática.

Orientador: Prof. Me. Janderson Ferreira Dutra.

Cajazeiras - PB

2024

IFPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

S586c Silva, Jerônimo Gabriel da.
Click studium : um sistema web para seleção de fotografias /
Jerônimo Gabriel da Silva. – 2024.

59f. : il.

Trabalho de Conclusão de Curso (Licenciatura em Computação e
Informática) - Instituto Federal de Educação, Ciência e Tecnologia da
Paraíba, Cajazeiras, 2024.

Orientador(a): Prof. Me. Janderson Ferreira Dutra.

1. Desenvolvimento de sistemas. 2. Sistema de fotografias. 3.
Estudio fotográfico. 4. Gerenciamento de imagens. I. Instituto
Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

IFPB/CZ

CDU: 004.4 (043.2)



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

JERÔNIMO GABRIEL DA SILVA

CLICK STUDIUM: UM SISTEMA WEB PARA SELEÇÃO DE FOTOGRAFIAS

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Licenciatura em Computação e Informática do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Licenciado em Computação e Informática.

Orientador:

Prof. Me. Janderson Ferreira Dutra

Aprovado em: **24 de Julho de 2024.**

Prof. Me. Janderson Ferreira Dutra - Orientador

Prof. Me. Fábio Abrantes Diniz - Avaliador

IFPB - Campus Cajazeiras

Prof. Antonio Ricart Jacinto de Oliveira Medeiros - Avaliador

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- Janderson Ferreira Dutra, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 04/09/2024 12:08:35.
- Antonio Ricart Jacinto de Oliveira Medeiros, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO, em 04/09/2024 12:26:49.
- Fabio Abrantes Diniz, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 04/09/2024 13:11:08.

Este documento foi emitido pelo SUAP em 04/09/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 601038
Verificador: 6f73e423c9
Código de Autenticação:



AGRADECIMENTOS

Deus esteve ao meu lado, dando-me força, ânimo e fé para não desistir e continuar lutando por este meu sonho e objetivo de vida. A Ele, eu devo minha gratidão. Chegou a hora de agradecer, e por isso começo por Deus, que esteve sempre presente em todos os momentos para garantir que minha meta fosse alcançada.

A esta instituição tão imponente, agradeço o ambiente propício à evolução e crescimento, bem como a todas as pessoas que a tornam tão especial. Pelo ambiente maravilhoso que encontrei todos os dias, sou grato, pois somente com um clima tão inspirador é possível atingir grandes objetivos.

Ao longo de todo o meu percurso, tive o privilégio de trabalhar de perto com os melhores professores, educadores e orientadores. Sem eles, não seria possível estar aqui hoje, de coração repleto de orgulho. A todos os professores e orientadores que me acompanharam ao longo dos últimos anos, deixo uma palavra sincera de gratidão. Sem sua paciência e sabedoria, eu jamais teria me tornado essa pessoa tão realizada.

À minha família e a todos os meus amigos verdadeiros, quero que saibam que reconheço tudo o que fizeram por mim, a força que inculcaram em meu pensamento para não desistir, e o conforto de saber que nunca estarei só e sempre serei capaz de enfrentar qualquer dificuldade.

A todos aqueles que não mencionei, mas que estiveram ao meu lado, prometo reconhecer essa proximidade, ajuda e incentivo todos os dias da minha vida.

RESUMO

A pandemia de COVID-19 levou à aceleração de transformações digitais em diversas áreas, ressaltando a importância de prover soluções que permitam interações remotas e reduzam o contato físico entre as pessoas. Este trabalho explora uma solução de *software* para fotógrafos e estúdios de fotografia, destacando seu papel na modernização do setor. O estudo apresenta sistema Click Studium cujo objetivo é contribuir para a melhoria da experiência do cliente, ao permitir a seleção remota de fotos, e para a promoção da segurança, ao reduzir a necessidade de visitas presenciais ao estúdio. Além disso, o trabalho examina a eficiência operacional, resultante da diminuição da ocupação dos computadores dos editores de fotografia, proporcionando segurança, produtividade e o bem-estar de fotógrafos e clientes. Os resultados são apresentados por meio de visualizações de modelos e artefatos elaborados durante o desenvolvimento do sistema. A avaliação do sistema pelos usuários revelou uma redução significativa no tempo necessário para conclusão dos projetos fotográficos, ao passo que contribuiu para a fidelização dos clientes, que perceberam uma maior rapidez na entrega final de seus pedidos.

Palavras-chave: Fotografias; Estúdios fotográficos; Desenvolvimento de software; Gerenciamento.

ABSTRACT

The COVID-19 pandemic has accelerated digital transformations in several areas, highlighting the importance of providing solutions that enable remote interactions and reduce physical contact between people. This paper explores a software solution for photographers and photography studios, highlighting its role in modernizing the sector. The study presents the Click Studium system, which aims to contribute to improving the customer experience by enabling remote selection of photos and promoting safety by reducing the need for in-person visits to the studio. In addition, the paper examines operational efficiency, resulting from the reduction in the occupancy of photo editors' computers, providing safety, productivity, and the well-being of photographers and clients. The results are presented through visualizations of models and artifacts created during the development of the system. The evaluation of the system by users revealed a significant reduction in the time required to complete photography projects, while contributing to customer loyalty, who perceived greater speed in the final delivery of their orders.

Keywords: Photographs; Photographic studios; Software development; Management.

LISTA DE FIGURAS

Figura 1: Processo de comunicação na <i>Web</i> por meio de um navegador.....	14
Figura 2: Exemplo da estrutura básica do HTML5.....	15
Figura 3: Estrutura do CSS.....	17
Figura 4: Exemplo de Código <i>JavaScript</i>	19
Figura 5: Arquitetura do NGINX.....	25
Figura 6: Etapas de desenvolvimento.....	28
Figura 7: Diagrama de caso de uso geral.....	36
Figura 8: Caso de uso do cliente.....	37
Figura 9: Diagrama de classes.....	38
Figura 10: Diagrama de sequência - cliente e seleção de fotografias.....	39
Figura 11: Diagrama de atividade - fluxo de tarefas.....	40
Figura 12: Arquitetura do sistema.....	41
Figura 13: <i>Landing page</i>	43
Figura 14: Tela de <i>login</i>	43
Figura 15: Lista de clientes.....	44
Figura 16: Lista e cadastro de pastas.....	45
Figura 17: Tela upload de fotos.....	45
Figura 18: Página inicial do cliente.....	46
Figura 19: Tela seleção de fotos.....	47
Figura 20: Tela de notificações do estúdio.....	47

LISTA DE QUADROS

Quadro 1: Comparativo de aplicações.....	27
Quadro 2: Requisitos funcionais do usuário cliente	33
Quadro 3: Requisitos Funcionais do usuário colaborador.....	34
Quadro 4: Requisitos Não-Funcionais.....	35

LISTA DE ABREVIATURAS E SIGLAS

HTTP	→ HyperText Transfer Protocol.
HTML	→ HyperText Markup Language.
CSS	→ Cascading Style Sheets.
TCP	→ Transmission Control Protocol.
IP	→ Internet Protocol.
DRY	→ Don't Repeat Yourself.
ORM	→ Object-Relational Mapping.
SQL	→ Structured Query Language.
CSRF	→ Cross-Site Request Forgery.
XSS	→ Cross-Site Scripting.
MVCC	→ Multiversion Concurrency Control.
SSL	→ Secure Sockets Layer.
TLS	→ Transport Layer Security.
B-tree	→ Balanced Tree.
Hash	→ Hashing.
GiST	→ Generalized Search Tree.
SP-GiST	→ Space-Partitioned Generalized Search Tree.
BRIN	→ Block Range INdex.
JSON	→ JavaScript Object Notation.
CI/CD	→ Continuous Integration/Continuous Deployment (ou Continuous Delivery).
SGBDs	→ Sistemas de Gerenciamento de Bancos de Dados.
CPU	→ Central Processing Unit.
UML	→ Unified Modeling Language.
VPS	→ Virtual Private Server.
OVH	→ On Vous Héberge.
API	→ Application Programming Interface.
AWS	→ Amazon Web Services.
DARPA	→ Defense Advanced Research Projects Agency.

SUMÁRIO

1. INTRODUÇÃO	9
1.1. MOTIVAÇÃO E DEFINIÇÃO DO PROBLEMA	10
1.2. OBJETIVO GERAL	11
1.3. OBJETIVOS ESPECÍFICOS	11
1.4. ORGANIZAÇÃO DO DOCUMENTO	12
2. FUNDAMENTAÇÃO TEÓRICA	13
2.1. INTERNET E WORLD WIDE WEB	13
2.2. HTML	14
2.3. CSS	16
2.4. JAVASCRIPT	17
2.5. PYTHON	19
2.6. DJANGO	20
2.7. POSTGRESQL	21
2.8. DOCKER	23
2.9. NGINX	24
2.10. A Dinâmica Operacional de um Estúdio Fotográfico	26
2.11. Trabalho relacionados	26
3. METODOLOGIA	28
3.1. LEVANTAMENTO DE REQUISITOS	29
3.2. PROJETO DO SISTEMA	29
3.3. IMPLEMENTAÇÃO	29
3.4. IMPLANTAÇÃO	30
3.5. REVISÃO E FEEDBACK COM O USUÁRIO	30
4. RESULTADOS E DISCUSSÕES	32
4.1. ANÁLISE DAS NECESSIDADES E FUNCIONALIDADES	32
4.2. MODELAGEM	36
4.3. ARQUITETURA DO SISTEMA	41
4.4. VISUALIZAÇÃO DO SISTEMA	42
5. CONSIDERAÇÕES FINAIS	49
5.1. TRABALHOS FUTUROS	50
REFERÊNCIAS	52

1. INTRODUÇÃO

No mercado dinâmico e multifacetado dos estúdios fotográficos, a busca por inovação e eficiência é uma constante. Tradicionalmente reconhecidos por seus serviços de alta qualidade, como ensaios fotográficos e cobertura de eventos, os estúdios enfrentam um desafio crescente em adaptar-se às demandas modernas e às mudanças de comportamento do consumidor (VIEIRA, 2020; PEIXOTO, 2020).

A sociedade foi profundamente e de várias formas impactada pela pandemia de COVID-19, provocando uma aceleração na transformação de hábitos e necessidades. O aumento da demanda por serviços digitais se destacou como uma característica importante deste período. O processo de transformação digital, que já estava em curso antes da pandemia, é destacado como uma das mudanças significativas para as empresas e foi acelerado significativamente por ela (REIS e REIS, 2021). É nesse contexto que emerge o software, uma plataforma digital projetada para atender às demandas emergentes dos fotógrafos e estúdios de fotografia.

Segundo Vial (2019, p. 121), a transformação digital é descrita como "um processo que busca aprimorar uma entidade por meio da combinação de tecnologias de informação, computação, comunicação e conectividade". Para otimizar suas operações e fornecer valor adicional aos clientes, é necessário não só adotar novas tecnologias, mas também realizar uma mudança cultural e estrutural dentro das organizações.

Neste contexto, este trabalho visa desenvolver uma sistema web que auxilie fotógrafos e estúdios de fotografia, destacando seu papel na modernização do setor. O estudo foca em como o sistema Click Studium contribui para a melhoria da experiência do cliente, ao permitir a seleção remota de fotos, e para a promoção da segurança, ao reduzir a necessidade de visitas presenciais ao estúdio. Para alcançar esses objetivos, foram utilizadas interações frequentes e feedback contínuo com os *stakeholders*.

Os principais instrumentos utilizados no desenvolvimento incluem o *framework Django*, para a criação do backend do sistema, e para o desenvolvimento do frontend, foram utilizados *HTML* e *CSS*, garantindo uma interface responsiva. Além disso, o trabalho examina a eficiência operacional resultante da diminuição da ocupação dos computadores dos editores de fotografia.

A implementação foi realizada em uma VPS (*Virtual Private Server*) na OVH, utilizando Docker para a gestão de containers do PostgreSQL, para o armazenamento de dados, assegurando escalabilidade e robustez na solução proposta.

1.1. MOTIVAÇÃO E DEFINIÇÃO DO PROBLEMA

A indústria fotográfica tem sido historicamente caracterizada por sua adaptabilidade e evolução constante, moldada pelas demandas do mercado e pelas tendências tecnológicas emergentes (CUNHA et al. 2020). Essa experiência com a pandemia impulsionou a sociedade a buscar e adotar soluções tecnológicas em diversos setores. Por exemplo, na educação, houve uma rápida expansão do ensino à distância; no setor de saúde, a telemedicina passou a ser amplamente utilizada; e no comércio, o e-commerce cresceu significativamente, exigindo respostas ágeis e inovadoras para garantir sua sobrevivência e crescimento contínuo (Agência CNI, 2020).

A pandemia trouxe consigo uma série de restrições e medidas de distanciamento social que impactaram significativamente os estúdios fotográficos e os profissionais da área (MALTA et al., 2020). Tradicionalmente focados na interação pessoal e na oferta de serviços presenciais, esses estúdios tiveram que repensar rapidamente seus modelos de negócios e adotar abordagens mais flexíveis e digitais.

Nesse contexto, diante das restrições impostas pela pandemia, os estúdios fotográficos enfrentam o desafio de manter a qualidade de seus serviços, garantir a satisfação do cliente e, ao mesmo tempo, priorizar a segurança e o bem-estar de todos os envolvidos (MALTA et al., 2020). Essa dualidade entre a necessidade de adaptação às novas realidades e a manutenção dos padrões de excelência estabelecidos delinea o cerne do problema a ser abordado neste estudo.

O *software* surge como uma solução proposta a essas questões, oferecendo uma plataforma digital abrangente que visa modernizar e simplificar os processos operacionais dos estúdios fotográficos. No entanto, a eficácia e a viabilidade dessa solução em atender às necessidades específicas dos estúdios fotográficos, especialmente em tempos de crise, são questões que exigem uma análise mais aprofundada.

Portanto, este trabalho desenvolveu um sistema web a fim de otimizar a operação e a eficiência dos estúdios fotográficos, considerando seu potencial para promover a segurança dos clientes e dos profissionais, ao mesmo tempo em que aprimora a experiência do usuário e mantém os padrões de qualidade esperados pela indústria fotográfica.

Este sistema *web* visa otimizar a operação e a eficiência do estúdio, reduzindo a necessidade de visitas presenciais dos clientes para a escolha de fotos, e assim, minimizar a ocupação dos computadores dos editores de fotografia. Além disso, o sistema busca atender às necessidades de clientes que residem longe do estúdio, permitindo que realizem a seleção de fotos no conforto de suas casas ou de qualquer outro lugar com acesso à internet.

1.2. OBJETIVO GERAL

Desenvolver um sistema *web* que permita aos clientes de estúdios fotográficos selecionarem suas fotografias de forma remota, utilizando dispositivos móveis ou computadores pessoais.

1.3. OBJETIVOS ESPECÍFICOS

- Compreender detalhadamente os processos operacionais utilizados no estúdio fotográfico, através de análises e entrevistas, para assegurar que a ferramenta atenda às necessidades reais e otimize o fluxo de trabalho.
- Avaliar tecnologias e metodologias de desenvolvimento mais adequadas para a solução proposta, garantindo eficiência, segurança e escalabilidade.
- Prover uma ferramenta com interface amigável e responsiva, compatível com dispositivos móveis e computadores pessoais, que permita aos clientes acessarem e selecionarem suas fotografias de forma intuitiva.
- Analisar como o sistema impacta nos processos operacionais dos estúdios fotográficos, identificando os benefícios e desafios da adoção dessa solução tecnológica.
- Discutir a satisfação dos clientes em relação ao uso do sistema Click Studium, investigando sua percepção sobre a experiência de seleção de fotos e interação remota com os serviços do estúdio fotográfico.

1.4. ORGANIZAÇÃO DO DOCUMENTO

Além dessa abordagem introdutória, este documento está organizado de acordo com os seguintes capítulos: no capítulo 2, apresenta-se a fundamentação teórica, abordando conceitos relacionados ao desenvolvimento de sistemas *web* e tecnologias utilizadas no Click Studium. No capítulo 3, detalha-se a metodologia, desde o levantamento de requisitos até a implantação do sistema. O capítulo 4 traz os resultados e discussões, analisando o impacto na operação e eficiência dos estúdios fotográficos, destacando melhorias e *feedbacks* dos usuários. Por fim, no capítulo 5, são discutidas as considerações finais, principais dificuldades e propostas de trabalhos futuros para a evolução contínua do sistema.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo introduz os conceitos fundamentais necessários para facilitar a compreensão do leitor sobre os temas correlatos abordados neste trabalho. A seguir, são detalhados os principais tópicos que sustentam o desenvolvimento e a implementação do *software*. Alguns dos tópicos abordados são: *Internet, World Wide Web, HTML, CSS, Javascript, Python, Django, Postgresql, Docker, Nginx*.

2.1. INTERNET E WORLD WIDE WEB

A Internet, como a conhecemos hoje, evoluiu a partir da ARPANET, desenvolvida na década de 1960 pela Agência de Projetos de Pesquisa Avançada do Departamento de Defesa dos Estados Unidos (DARPA). A ARPANET foi pioneira ao utilizar a comutação de pacotes, permitindo a comunicação mesmo em caso de falhas na rede. Para possibilitar essa comunicação eficiente, Vinton Cerf e Robert Kahn desenvolveram o conjunto de protocolos TCP/IP, que estabeleceu as regras e formatos para a troca de dados na rede, formando a base da Internet moderna (TANENBAUM e WETHERALL, 2011; KUROSE e ROSS, 2013).

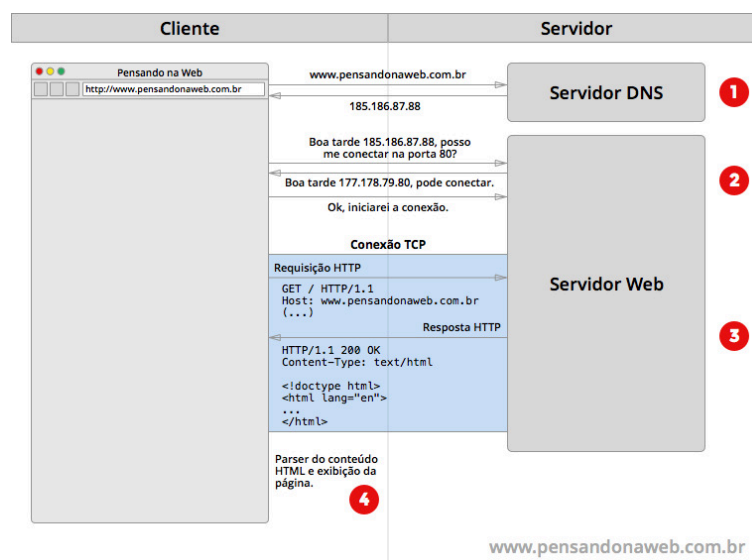
Em 1989, Tim Berners-Lee, cientista do CERN, criou a *World Wide Web* (WWW) para organizar e compartilhar informações na Internet. A WWW, um sistema de documentos interligados acessíveis pela Internet, revolucionou a maneira como a informação é compartilhada globalmente. Utilizando protocolos como *HTTP* (*Hypertext Transfer Protocol*), *URLs* e *HTML*, a *Web* permitiu a navegação e visualização de páginas web, tornando-se uma ferramenta essencial para a comunicação e o acesso à informação (BERNERS-LEE, 1994; FOROUZAN e OLIVEIRA, 2008).

Com o objetivo de facilitar a comunicação entre pesquisadores, Berners-Lee definiu três elementos centrais:

- Hipertexto: Permite que os usuários naveguem facilmente entre documentos interligados por meio de links.
- Protocolo *HTTP* (*HyperText Transfer Protocol*): Define as regras para a troca de dados entre navegadores e servidores web.
- Linguagem *HTML* (*HyperText Markup Language*): Define a estrutura e o conteúdo das páginas da web.

O *browser* (navegador) é uma ferramenta essencial para a navegação na *web*, permitindo aos usuários solicitarem diversos tipos de informações. A Figura 1 mostra o fluxo da requisição de um cliente a um recurso disponível em um servidor.

Figura 1: Processo de comunicação na *Web* por meio de um navegador



Fonte: DOS SANTOS (2014)

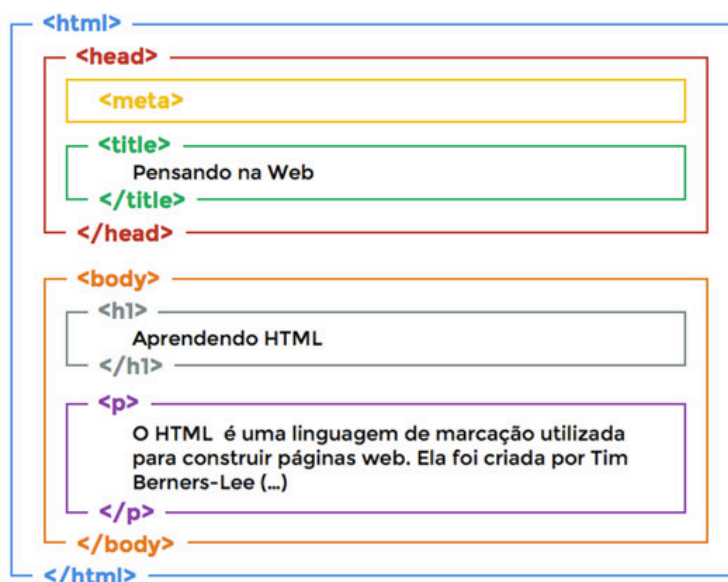
Neste processo, o navegador envia uma requisição HTTP ao servidor, que por sua vez é responsável por processar a solicitação e enviar uma resposta adequada de volta ao cliente. Este mecanismo de requisição e resposta é fundamental para o funcionamento da *World Wide Web* (FOROUZAN e OLIVEIRA, 2008; RHODES e GOERZEN, 2015).

2.2. HTML

O *HTML* é a linguagem padrão utilizada para criar e estruturar páginas na *World Wide Web*. Ele permite que os desenvolvedores descrevam a estrutura de documentos *web* utilizando elementos que representam diferentes tipos de conteúdo, como texto, imagens, *links* e formulários. Desde sua criação, o *HTML* tem evoluído para suportar uma ampla gama de funcionalidades que são essenciais para a construção de *websites* modernos (TORRES, 2018; SCHROEDER, 2012; DUARTE, 2023; SOARES, 2022).

A estrutura básica de um documento *HTML* é composta por uma série de elementos aninhados, cada um com uma função específica. A maioria dos documentos *HTML* começa com a declaração `<!DOCTYPE html>`, que informa ao navegador que o documento está utilizando a versão HTML5. Em seguida, o documento é estruturado em duas partes principais: o `<head>`, que contém metadados, *links* para folhas de estilo e *scripts*; e o `<body>`, que contém o conteúdo visível da página, incluindo parágrafos, cabeçalhos, listas, links e outros elementos (FLATSCHART, 2011; SCHROEDER, 2012). A figura 2 representa a estrutura básica do HTML.

Figura 2: Exemplo da estrutura básica do HTML5



Fonte: DOS SANTOS (2014)

Os elementos *HTML* são os blocos de construção da *web*. Cada elemento é definido por uma *tag* de abertura, um conteúdo e uma *tag* de fechamento. Alguns dos elementos mais comuns incluem:

- `<p>` para parágrafos;
- `<h1>` a `<h6>` para cabeçalhos;
- `<a>` para hyperlinks;
- `` para imagens;
- `` e `` para listas não ordenadas e ordenadas, respectivamente.

Os atributos fornecem informações adicionais sobre os elementos e são incluídos dentro das *tags* de abertura. Por exemplo, o atributo `href` em uma tag `<a>` define o destino de um *hyperlink*, e o atributo `src` em uma tag `` especifica o caminho da imagem a ser exibida (Torres, 2018).

O *HTML* é fundamental para a criação de *websites* acessíveis e responsivos. Ele trabalha em conjunto com *CSS* (*Cascading Style Sheets*) e *JavaScript* para proporcionar uma experiência de usuário rica e interativa. A semântica adequada dos elementos *HTML* melhora a acessibilidade, permitindo que dispositivos de leitura de tela interpretem o conteúdo de maneira mais eficaz para usuários com deficiências visuais. Além disso, a correta utilização do *HTML* é essencial para a otimização dos mecanismos de busca (SEO), ajudando os *websites* a alcançarem melhor visibilidade nos resultados de busca (DUARTE, 2023; SOARES, 2022).

2.3. CSS

CSS (*Cascading Style Sheets*) pode ser descrito como uma “folha de estilo” composta por camadas que é utilizada para definir a aparência de uma aplicação web. Ele especifica como os elementos contidos no código *HTML* serão exibidos, permitindo a separação entre a estrutura do documento e a sua apresentação visual (SILVA, 2020; DUARTE, 2023; SOARES, 2022). Os recursos de programação das páginas da internet evoluíram significativamente, adotando uma variedade cada vez maior de estilos que proporcionam mais elegância e atração aos usuários. Essa evolução tem sido fundamental para melhorar a experiência do usuário, tornando as páginas mais agradáveis e intuitivas (SILVESTRE e BATISTELA, 2019).

O *CSS3* é a terceira e mais recente versão da linguagem *CSS*, oferecendo uma ampla gama de novos recursos que permitem definir estilos de maneira mais sofisticada e eficiente. Entre os novos recursos estão as transições, animações, gradientes, e a capacidade de manipular o *background* e as bordas dos elementos de forma mais avançada (SOARES, 2022). Por exemplo, é possível criar bordas arredondadas, sombras e aplicar múltiplas imagens de plano de fundo sem a necessidade de usar imagens externas, o que otimiza o carregamento e a performance das páginas (RABETTI, 2019).

A principal função do *CSS3* é permitir a criação de *layouts* mais complexos e visualmente atraentes, eliminando a necessidade de várias imagens de fundo e bordas arredondadas que anteriormente eram feitas com gráficos externos. Além disso, o *CSS3* introduz efeitos de transição e animação que possibilitam a criação de interfaces mais interativas e dinâmicas (SILVA, 2020).

Conforme destacado por Rabetti (2019), o CSS é uma ferramenta essencial para o desenvolvimento *web* moderno, proporcionando uma maneira eficiente de controlar a apresentação de documentos *HTML* e permitindo que os desenvolvedores criem experiências visuais ricas sem sobrecarregar a estrutura do documento com código de estilo (SOARES, 2022).

A Figura 3 representa um exemplo de código CSS onde o seletor `h1` é usado para estilizar todos os elementos `<h1>` em uma página *web*.

Figura 3: Estrutura do CSS

```
h1 {  
  color: blue;  
  font-size: 20px;  
}  
  
@media (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

Fonte: o autor (2024)

A cor do texto dos elementos `<h1>` é definida como azul (`color: blue;`) e o tamanho da fonte é ajustado para 20 *pixels* (`font-size: 20px;`). Além disso, o código inclui uma consulta de mídia (`@media`) que aplica estilos diferentes com base no tamanho da tela do dispositivo. Especificamente, quando a largura da tela é de no máximo 600 *pixels* (`max-width: 600px`), a cor de fundo do elemento `<body>` é alterada para azul claro (`background-color: lightblue;`). Essa técnica é usada para criar designs responsivos que se adaptam a diferentes tamanhos de dispositivos, como *tablets* e *smartphones*, proporcionando uma experiência de usuário consistente e agradável.

2.4. JAVASCRIPT

A linguagem de *script JavaScript*, criada em 1995 pela *Netscape*, rapidamente ganhou força ao implementar dinamicidade ao ambiente hospedeiro,

caracterizando-se como uma linguagem leve e versátil (DA SILVA GUNTZEL et al., 2019). Sendo multiparadigma, interpretada e baseada em objetos com funções de primeira classe, *JavaScript* se popularizou como a linguagem de *script* para páginas *web*, mas também encontrou aplicação em ambientes fora dos navegadores, como *node.js*, *Apache CouchDB* e *Adobe Acrobat* (DA SILVA GUNTZEL et al., 2019).

JavaScript é fundamental na criação de páginas *web*, e suas funções podem ser inseridas diretamente nos documentos *HTML*, permitindo o aprimoramento destes. Através de JavaScript, é possível responder facilmente a eventos iniciados pelo usuário e incluir efeitos que tornam a página dinâmica. Isso permite o desenvolvimento de páginas sofisticadas e interativas (Silva, 2020). A linguagem compõe a tríade de tecnologias essenciais no desenvolvimento *web*, juntamente com *HTML* e *CSS*. Enquanto o *HTML* é utilizado para determinar o conteúdo das páginas e o *CSS* para refinar a apresentação, *JavaScript* especifica o comportamento dessas páginas, possibilitando a criação de interfaces ricas e interativas (SILVESTRE e BATISTELA, 2019). *JavaScript* é a linguagem de programação da *web*, e a ampla maioria dos sites modernos a utiliza, tornando-a a linguagem de programação mais onipresente da história (SOARES, 2022).

JavaScript também se destaca pela vasta quantidade de bibliotecas e *frameworks* disponíveis, que facilitam o desenvolvimento *web*. Ferramentas como *jQuery*, *Angular*, *React* e *Vue.js* fornecem abstrações e funcionalidades adicionais que ajudam a gerenciar o estado da aplicação, manipular o DOM de forma eficiente e implementar componentes reutilizáveis. Por exemplo, a biblioteca *SAPUI5* é amplamente utilizada no ambiente corporativo para construir interfaces de usuário sofisticadas e responsivas (ALVES e PETRUCCELLI, 2019).

Além disso, a educação em *JavaScript* é crucial para preparar novos desenvolvedores *web*. Programas como o *JavaScript Academy* oferecem um arcabouço estruturado para ensinar *JavaScript*, com lições práticas e exemplos que ajudam os alunos a entender e aplicar os conceitos fundamentais da linguagem (CORREIA, 2019). Esse tipo de iniciativa é vital para a formação de profissionais competentes na área de desenvolvimento *web*.

A Figura 4 representa um exemplo de código *JavaScript* onde utilizou-se o método *getElementById* para selecionar um elemento *HTML* com o ID "*myButton*". Em seguida, adicionou-se um ouvinte de evento (*addEventListener*) para detectar

cliques nesse botão. Quando o botão é clicado, a função anônima fornecida como segundo argumento é executada.

Figura 4: Exemplo de Código *JavaScript*

```
document.getElementById("myButton").addEventListener("click", function() {
    console.log("Hello, World!");
    alert("Button clicked!");
});
```

Fonte: o autor (2024)

Dentro dessa função, duas ações ocorrem: primeiro, a mensagem "*Hello, World!*" é registrada no console do navegador (`console.log("Hello, World!");`) e, em seguida, um alerta aparece na tela com a mensagem "*Button clicked!*" (`alert("Button clicked!");`). Isso demonstra como interagir com elementos *HTML* e responder a eventos do usuário utilizando *JavaScript*.

2.5. PYTHON

Python é uma linguagem de programação de alto nível, interpretada e de propósito geral, que vem ganhando destaque no desenvolvimento de *software*. Criada por Guido van Rossum e lançada em 1991, *Python* foi projetada com a filosofia de enfatizar a legibilidade do código, permitindo que os desenvolvedores escrevam menos código para realizar tarefas mais complexas (PAIVA et al., 2019).

Python é amplamente utilizada em diversas áreas do desenvolvimento de *software*, incluindo automação, análise de dados, desenvolvimento web e aprendizado de máquina. Sua sintaxe clara e intuitiva, junto com uma vasta coleção de bibliotecas e *frameworks*, facilita a implementação de soluções eficientes e escaláveis (SILVA e SILVA, 2019). A linguagem é reconhecida pela sua facilidade de aprendizagem e uso, o que a torna uma escolha popular tanto entre iniciantes quanto entre programadores experientes.

No contexto de desenvolvimento *web*, *Python* se destaca especialmente com o uso de *frameworks* como *Django* e *Flask*, que permitem a criação de aplicações *web* robustas e escaláveis de maneira rápida e eficiente (SANTIAGO et al., 2020). O *Django*, em particular, é conhecido por seguir o princípio de "baterias incluídas",

oferecendo uma ampla gama de funcionalidades prontas para uso, o que acelera o desenvolvimento e reduz a necessidade de reinventar a roda.

Além disso, *Python* é amplamente utilizado em aplicações de automação industrial e supervisão de sistemas. Por exemplo, no desenvolvimento de sistemas supervisórios para dispositivos Modbus *TCP/IP*, *Python* demonstra sua flexibilidade e poder ao ser utilizada para monitoramento e controle de dispositivos em ambientes industriais, proporcionando uma interface *web* intuitiva para gerenciamento e visualização de dados (COSTA, 2022).

A linguagem também desempenha um papel crucial na análise de dados e ciência de dados. A facilidade com que *Python* integra bibliotecas como *Pandas*, *NumPy* e *Matplotlib* torna a análise exploratória de dados mais acessível e eficiente. Estas bibliotecas fornecem ferramentas poderosas para manipulação, análise e visualização de grandes volumes de dados, facilitando a extração de *insights* valiosos (LOPES et al., 2019).

Outro aspecto importante é a utilização de *Python* em sistemas de medição para faturamento, onde a linguagem é empregada para desenvolver sistemas *web* que gerenciam e monitoram pontos de medição, garantindo a precisão e eficiência no processo de faturamento (EUROPEU, 2022).

2.6. DJANGO

Django é um *framework* de alto nível para o desenvolvimento de aplicações *web*, escrito em *Python*, que promove um desenvolvimento rápido e um *design* limpo e pragmático. Criado inicialmente para gerenciar o conteúdo de sites jornalísticos, *Django* foi lançado publicamente em 2005 e desde então tem se destacado por sua capacidade de facilitar a criação de aplicações *web* complexas com uma base de código minimalista (SANTIAGO et al., 2020).

O principal objetivo do *Django* é permitir que os desenvolvedores se concentrem em escrever componentes novos sem precisar reinventar a roda. Ele é conhecido por seguir o princípio *DRY* (*Don't Repeat Yourself*), que incentiva a reutilização de código e a manutenção de uma base de código eficiente. *Django* vem com uma série de funcionalidades embutidas que facilitam o desenvolvimento de aplicações *web* seguras, escaláveis e de fácil manutenção (SOARES, 2022).

Uma das principais características do *Django* é o seu poderoso sistema de *ORM (Object-Relational Mapping)*, que permite aos desenvolvedores interagir com bancos de dados utilizando objetos *Python* em vez de escrever *SQL (Structured Query Language)* manualmente. Isso não apenas acelera o processo de desenvolvimento, mas também reduz a possibilidade de erros e torna o código mais legível e mantido (DUARTE, 2023).

Além disso, *Django* inclui um sistema de administração automática, que gera uma interface administrativa a partir dos modelos de dados definidos pelos desenvolvedores. Este sistema permite a gestão eficiente de dados, oferecendo uma interface amigável e robusta para os administradores do site, sem a necessidade de codificação adicional (SANTIAGO et al., 2020).

Django também promove práticas de desenvolvimento seguro, incorporando proteções contra ataques comuns como *CSRF (Cross-Site Request Forgery)*, *XSS (Cross-Site Scripting)* e *SQL Injection*. Isso ajuda a garantir que as aplicações desenvolvidas com *Django* sejam resilientes contra vulnerabilidades conhecidas e estejam alinhadas com as melhores práticas de segurança na *web* (SOARES, 2022).

A flexibilidade do *Django* o torna ideal para uma variedade de aplicações, desde *blogs* simples até sistemas de gerenciamento de conteúdo complexos e plataformas de *e-commerce*. Sua arquitetura modular permite que os desenvolvedores integrem facilmente bibliotecas e ferramentas externas, expandindo ainda mais suas capacidades e permitindo a criação de soluções personalizadas para necessidades específicas (DUARTE, 2023).

2.7. POSTGRESQL

PostgreSQL é um sistema de gerenciamento de banco de dados *objeto-relacional* de código aberto, conhecido por sua robustez, escalabilidade e conformidade com o padrão *SQL*. Inicialmente lançado em 1989 como "*Postgres*" no departamento de ciência da computação da Universidade da *Califórnia* em *Berkeley*, o projeto evoluiu significativamente, tornando-se *PostgreSQL* em 1996 após a adição do suporte completo a *SQL* (SOARES, 2022).

Um dos principais diferenciais do *PostgreSQL* é a sua aderência aos padrões *SQL* e a ampla gama de recursos avançados que oferece. Estes incluem suporte a transações *ACID* (Atomicidade, Consistência, Isolamento e Durabilidade), índices

avançados, tipos de dados personalizados, controle de concorrência *multiversion* (MVCC), e a capacidade de armazenar e manipular grandes volumes de dados com alta eficiência. O MVCC, em particular, permite que o *PostgreSQL* execute operações de leitura e escrita de maneira concorrente sem conflitos, aumentando a performance e a escalabilidade do sistema (MIYAGUSKU, 2008).

PostgreSQL também é reconhecido por sua extensibilidade. Os usuários podem definir novos tipos de dados, operadores, funções agregadas, linguagens de procedimento, e até mesmo métodos de indexação. Essa extensibilidade torna o *PostgreSQL* uma escolha popular para aplicações que necessitam de funcionalidades específicas ou personalizadas que não estão disponíveis em outros sistemas de gerenciamento de banco de dados (CARVALHO, 2017).

A segurança é outro aspecto crucial do *PostgreSQL*. Ele incorpora uma variedade de mecanismos de segurança para proteger os dados armazenados, incluindo autenticação baseada em senha, criptografia de dados em trânsito usando *SSL*, e controles de acesso refinados que permitem definir permissões específicas para diferentes usuários e funções dentro do banco de dados. Adicionalmente, *PostgreSQL* pode ser configurado para conformidade com padrões de segurança rigorosos, tornando-o uma opção viável para aplicações empresariais que lidam com dados sensíveis (BERNIER, 2009).

Em termos de uso prático, *PostgreSQL* é amplamente adotado em aplicações web modernas devido à sua capacidade de gerenciar grandes volumes de dados com alta eficiência e confiabilidade. Ele suporta diversos tipos de indexação, como *B-tree*, *Hash*, *GiST*, *SP-GiST*, *GIN*, e *BRIN*, que podem ser utilizados para otimizar o desempenho das consultas. Além disso, o *PostgreSQL* oferece suporte nativo para dados *JSON*, permitindo a integração fluida com aplicações baseadas em documentos e APIs *RESTful* (SOARES, 2022).

A comunidade ativa de desenvolvimento e a vasta documentação disponível também contribuem para a popularidade do *PostgreSQL*. A comunidade não só mantém e aprimora continuamente o sistema, mas também fornece suporte através de fóruns, listas de discussão, e conferências dedicadas. Esta rede de suporte ajuda a garantir que o *PostgreSQL* permaneça atualizado com as últimas tecnologias e práticas de desenvolvimento, oferecendo uma solução confiável e moderna para o gerenciamento de bancos de dados (CARVALHO, 2017).

2.8. DOCKER

Docker é uma plataforma de código aberto que automatiza a implantação de aplicações dentro de contêineres de software, fornecendo uma camada adicional de abstração e automação de virtualização a nível de sistema operacional. Introduzido em 2013 pela *Docker Inc.*, o *Docker* facilita a criação, o envio e a execução de aplicações de forma consistente em diferentes ambientes (SANTOS et al., 2019).

Os contêineres *Docker* são leves e portáteis, encapsulando tudo o que uma aplicação necessita para ser executada: código, bibliotecas, variáveis de ambiente e arquivos de configuração. Essa abordagem resolve o problema da incompatibilidade entre ambientes de desenvolvimento e produção, garantindo que o software funcione de maneira idêntica em qualquer lugar (NETTO, 2019).

Uma das principais vantagens do *Docker* é a eficiência no uso de recursos do sistema. Ao contrário das máquinas virtuais tradicionais, que requerem uma cópia completa do sistema operacional e uma camada de hipervisor, os contêineres *Docker* compartilham o kernel do sistema operacional *host*, o que reduz significativamente a sobrecarga e melhora o desempenho. Essa eficiência faz do *Docker* uma escolha ideal para ambientes de desenvolvimento ágil e integração contínua (CI/CD) (SANTOS et al., 2019).

Além disso, o *Docker* facilita a escalabilidade e a orquestração de serviços complexos. Ferramentas como *Docker Compose* permitem a definição de configurações multi-contêiner, enquanto plataformas de orquestração como *Kubernetes* e *Docker Swarm* gerenciam a distribuição e a escalabilidade de contêineres em clusters de servidores. Essas capacidades são cruciais para aplicações modernas baseadas em microserviços, onde componentes independentes são desenvolvidos, implantados e escalados de forma isolada (Netto, 2019).

No contexto de bancos de dados, o *Docker* provou ser uma ferramenta valiosa. A possibilidade de criar ambientes de banco de dados isolados e consistentes para desenvolvimento, testes e produção agiliza o ciclo de vida do desenvolvimento de *software*. Estudos de desempenho, como o realizado por Santos et al. (2019), mostram que contêineres *Docker* podem executar diferentes Sistemas de Gerenciamento de Banco de Dados (SGBDs) relacionais com

desempenho comparável ao de instalações nativas, destacando sua viabilidade para aplicações críticas.

A segurança dos contêineres também é um aspecto importante. *Docker* implementa várias camadas de segurança, incluindo isolamento de namespace, controle de recursos via *cgroups*, e a capacidade de definir políticas de segurança detalhadas. No entanto, a segurança em ambientes de contêineres exige atenção constante, especialmente em cenários de produção onde vulnerabilidades podem ser exploradas se não forem gerenciadas adequadamente (NETTO, 2019).

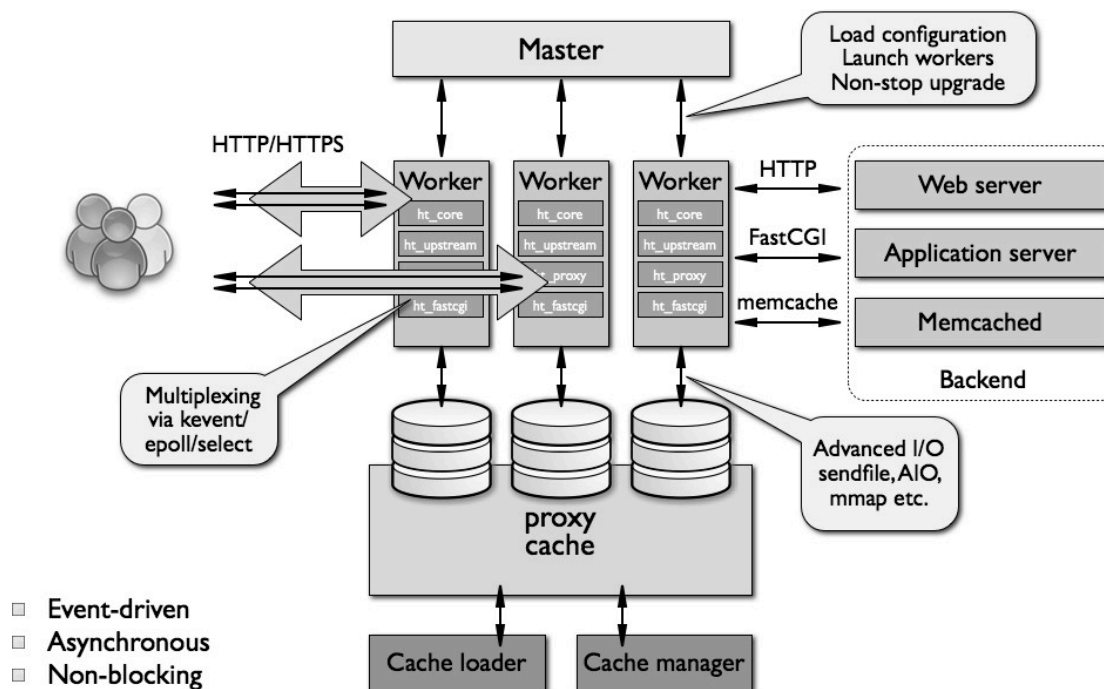
Por fim, *Docker* representa um avanço significativo na maneira como as aplicações são desenvolvidas, distribuídas e executadas. Sua abordagem de contêineres fornece uma solução eficiente e escalável para os desafios de compatibilidade e isolamento em ambientes de TI, tornando-se uma tecnologia essencial no ecossistema de desenvolvimento de software moderno.

2.9. NGINX

NGINX é um servidor web de código aberto que também funciona como um proxy reverso, balanceador de carga e cache HTTP. Lançado inicialmente em 2004 por Igor Sysoev, o *NGINX* foi desenvolvido para resolver problemas de desempenho do Apache, especialmente em relação à alta concorrência de conexões (SILVA; HUSEMANN; MALHEIROS, 2017).

A arquitetura do *NGINX* é baseada em eventos e altamente escalável, o que lhe permite lidar com milhares de conexões simultâneas com o uso eficiente dos recursos do sistema. Isso contrasta com a abordagem de processos ou threads do *Apache*, tornando o *NGINX* mais eficiente em cenários de alta carga. Essa eficiência é particularmente evidente quando comparado em ambientes com diferentes arquiteturas de CPU, como ARM *Cortex A57* e *Intel x86*, onde o *NGINX* demonstra superioridade em termos de desempenho e consumo de recursos (SILVA; HUSEMANN; MALHEIROS, 2017). A Figura 5 mostra a arquitetura do *NGINX*.

Figura 5: Arquitetura do NGINX



Fonte: ALEXEEV (2012)

Além de servir conteúdo estático de maneira eficiente, o *NGINX* também é amplamente utilizado como *proxy* reverso e balanceador de carga. Ele distribui o tráfego de entrada entre vários servidores backend, melhorando a disponibilidade e a confiabilidade dos serviços *web*. Os algoritmos de balanceamento de carga no *NGINX*, como *round-robin*, *least connections* e *IP hash*, permitem uma distribuição eficiente do tráfego, aumentando a escalabilidade das aplicações *web* (Tamanho, Domingues e Bittencourt, 2021).

A flexibilidade do *NGINX* também se estende à sua capacidade de cache, o que pode melhorar significativamente o desempenho de aplicações *web* ao armazenar em *cache* respostas de *backend* para solicitações futuras. Esta funcionalidade é essencial para a criação de ambientes *web* de alto desempenho, como demonstrado na implementação do portal CEULP/ULBRA, onde o *NGINX* foi utilizado para otimizar a entrega de conteúdo e garantir alta disponibilidade do serviço (NEVES JUNIOR e SOUSA, 2012).

Em termos de configuração e administração, o *NGINX* é reconhecido por sua simplicidade e eficiência. Seus arquivos de configuração são concisos e fáceis de ler, o que facilita a implementação de ajustes e melhorias sem a necessidade de reiniciar o servidor. Esta característica, aliada ao suporte ativo da comunidade e à

documentação abrangente, faz do *NGINX* uma escolha popular entre desenvolvedores e administradores de sistemas (SILVA; HUSEMANN; MALHEIROS, 2017).

Em suma, o *NGINX* representa uma solução robusta e eficiente para o gerenciamento de serviços *web*. Sua arquitetura baseada em eventos, capacidade de balanceamento de carga, funcionalidade de *proxy* reverso e recursos de *cache* fazem dele uma ferramenta indispensável para a criação de ambientes *web* modernos e de alto desempenho.

2.10. A Dinâmica Operacional de um Estúdio Fotográfico

A dinâmica de um estúdio fotográfico envolve uma série de processos desde o primeiro contato do cliente até a entrega final do produto. Inicialmente, o cliente entra em contato com o estúdio, seja por telefone, WhatsApp, Instagram, ou pessoalmente, para discutir detalhes como a data do evento e os pacotes disponíveis. Após a realização do evento, as fotos são descarregadas no computador do estúdio, e o cliente é convidado a selecionar as imagens que deseja incluir no álbum. Este processo pode exigir várias visitas ao estúdio, utilizando um dos computadores dos editores para escolher as fotos que serão editadas.

Uma vez que as fotos são selecionadas e editadas, elas são enviadas para uma empresa especializada que se encarrega da revelação e montagem do álbum físico. O estúdio, então, recebe o álbum pronto e notifica o cliente para que ele possa retirar o produto final. Esse processo envolve a interação direta com o cliente em várias etapas, garantindo que o resultado final atenda às suas expectativas, e também demanda a colaboração com empresas terceirizadas para a produção dos álbuns, o que ressalta a importância de uma gestão eficiente do fluxo de trabalho dentro do estúdio.

2.11. Trabalho relacionados

O Click Studium se destaca significativamente em comparação com os outros sistemas analisados, como Epics, Selpics e Picsize. Enquanto os concorrentes exigem múltiplos passos burocráticos e a adesão a planos limitados, o Click Studium simplifica todo o processo. Com apenas alguns cliques e entradas mínimas de

dados, o sistema realiza muito mais, oferecendo uma experiência rápida e intuitiva para o usuário. Essa simplicidade é complementada pela ausência de burocracia e pela flexibilidade na escolha de planos, tornando-o uma opção muito mais acessível e eficiente. Um comparativo entre os sistemas supracitados e o Click Studium é mostrado no Quadro 1.

Quadro 1: Comparativo de aplicações

Sistemas	Seleção de fotos	Simplicidade	Burocracia	Planos pagos	Limite de fotos
Click studium	Sim	Sim	Não	Não	Não
Epics	Sim	Não	Sim	Sim	Sim
Selpics	Sim	Não	Sim	Sim	Sim
Picsize	Sim	Não	Sim	Sim	Sim

Fonte: o autor (2024)

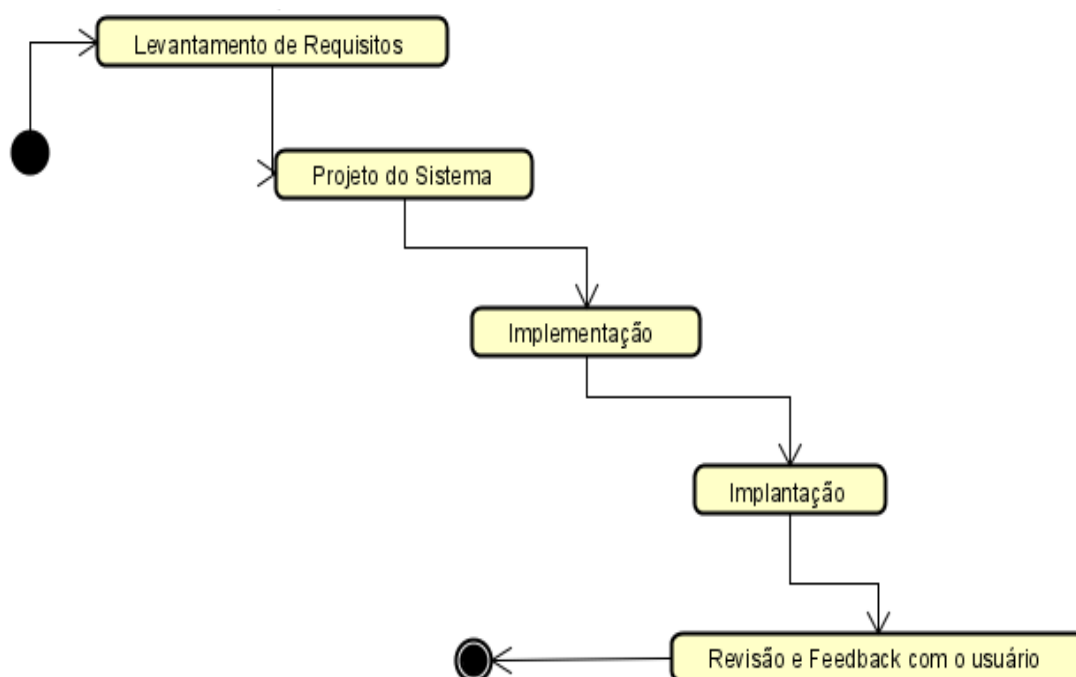
Além disso, o Click Studium oferece uma interface exclusiva para cada cliente, com uma página inicial própria, fácil de usar. Essa abordagem permite que os clientes gerenciem suas seleções de fotos de maneira independente e com maior praticidade. O sistema também é otimizado para um upload de fotos mais eficiente, garantindo que as imagens sejam processadas e disponibilizadas rapidamente, sem complicações. Esses diferenciais fazem do Click Studium uma solução superior e preferida entre as opções disponíveis no mercado.

3. METODOLOGIA

Nesta seção são destacados os procedimentos metodológicos propostos para o desenvolvimento do trabalho. Foi realizada uma pesquisa de natureza aplicada e do tipo exploratória, tendo em vista que trata-se de uma investigação direcionada à construção prática da solução para o objeto de investigação. Quanto à finalidade trata-se de uma pesquisa experimental (KÖCHE, 2016). A apresentação dos resultados é realizada por meio de visualizações de modelos elaborados durante o desenvolvimento do sistema.

A metodologia deste trabalho foi estruturada em etapas que integraram as abordagens *Scrum* e *Cascata*, permitindo uma elaboração e implementação eficiente do sistema, bem como a avaliação de seus resultados, conforme mostra a Figura 6. A combinação destas metodologias proporcionou uma estrutura flexível e iterativa, característica do *Scrum*, juntamente com a abordagem sequencial e linear da *Cascata*, garantindo um desenvolvimento organizado e adaptável às necessidades do projeto.

Figura 6: Etapas de desenvolvimento



Fonte: o autor (2024)

As próximas subseções detalham cada uma das etapas seguidas. Os resultados de cada etapa são apresentados e discutidos no Capítulo 4.

3.1. LEVANTAMENTO DE REQUISITOS

Inicialmente, realizou-se um levantamento detalhado das necessidades do estúdio de fotografia Roberto *Photo Studio* em Guarabira-PB. Esse processo envolveu entrevistas e reuniões com os proprietários e funcionários do estúdio para identificar os principais problemas enfrentados na gestão do fluxo de clientes e na escolha das fotos.

Nesta etapa foram realizadas reuniões detalhadas com os *stakeholders* do estúdio Roberto *Photo Studio* para entender suas necessidades e expectativas em relação ao *software*. Foram utilizadas algumas técnicas de elicitação de requisitos como, por exemplo, entrevistas e conversas informais, para coletar informações sobre as funcionalidades desejadas, problemas atuais e expectativas de melhoria.

A partir dos requisitos funcionais (seção 4.1), foi criado um *backlog* que serviu de base para o desenvolvimento do sistema. Essas informações permitiram mapear de forma precisa as demandas do estúdio, visando otimizar o fluxo de trabalho e melhorar a experiência dos clientes.

3.2. PROJETO DO SISTEMA

Com os requisitos definidos, iniciou-se a fase de projeto do sistema. Alguns diagramas UML (*Unified Modeling Language*) foram utilizados para representar a estrutura e o comportamento do sistema. Um diagrama de classes foi elaborado para mostrar as entidades e suas relações. Diagramas de casos de uso foram usados para ilustrar a interação dos usuários com as funcionalidades específicas do sistema.

A arquitetura do sistema foi modelada, onde foram definidos os componentes principais e interações com as tecnologias utilizadas como suporte ao desenvolvimento. A estrutura projetada garantiu uma base sólida para o desenvolvimento, permitindo a eficiência do funcionamento do Click Studium.

3.3. IMPLEMENTAÇÃO

A implementação do software foi conduzida seguindo práticas ágeis de desenvolvimento, permitindo interações frequentes e *feedback* contínuo com os

stakeholders do estúdio. Utilizou-se as linguagens de programação *Python* e o *framework Django* para o desenvolvimento do sistema *web*, com segurança e usabilidade. Durante esta fase, foram desenvolvidas as funcionalidades principais, como cadastro de clientes, upload de fotos, criação de pastas e mecanismos para que os clientes pudessem selecionar e comentar fotos.

Na criação da interface do usuário foi usado o *HTML* para estruturar as páginas *web*. Em seguida, *CSS* para definir a aparência do sistema, dando uma separação clara entre a estrutura do documento e a sua apresentação visual.

3.4. IMPLANTAÇÃO

Após a fase de desenvolvimento, a implantação do software foi realizada em uma *VPS (Virtual Private Server)* alugada na *OVH*, utilizando o sistema operacional *Ubuntu*. Optou-se pelo *Ubuntu* devido à sua estabilidade, segurança e compatibilidade com uma ampla gama de ferramentas e tecnologias de desenvolvimento. Essa escolha nos permitiu criar um ambiente de servidor robusto e confiável.

Inicialmente, foi configurado o servidor *web Nginx*, reconhecido por sua eficiência em servir conteúdo estático e atuar como um *proxy* reverso.

Para que os *stakeholders* façam uma transição eficaz para o sistema, “treinamos” os funcionários do Roberto *Photo Studio* no uso do *software*. Realizamos sessões de demonstração para familiarizar os funcionários com as novas funcionalidades, fluxos de trabalho e práticas recomendadas para utilizar o sistema de maneira eficiente. Dessa forma, a implantação do *software* na *VPS* da *OVH*, utilizando *Ubuntu*, *Docker*, *Nginx* e *PostgreSQL*, proporcionou uma solução robusta e escalável, melhorando a satisfação dos clientes e a eficiência operacional do estúdio fotográfico.

3.5. REVISÃO E FEEDBACK COM O USUÁRIO

A última etapa envolve a validação do sistema com os usuários finais. O *feedback* dos clientes foi realizado pelos funcionários do estúdio por meio de conversas informais para avaliar a eficácia do *software* em melhorar a operação e eficiência do estúdio fotográfico. Verificou-se que o sistema atende aos requisitos e

expectativas definidos na fase de levantamento. Foram identificadas áreas de melhoria e ajustes necessários para gerar uma boa satisfação dos usuários e qualidade do serviço.

Essa metodologia sistemática e iterativa garantiu que o desenvolvimento e implementação do sistema sejam realizados de maneira eficiente, atendendo às necessidades do Roberto *Photo Studio* e proporcionando uma experiência positiva para seus clientes.

4. RESULTADOS E DISCUSSÕES

A primeira etapa do desenvolvimento do sistema consistiu em um estudo detalhado das necessidades do Roberto *Photo Studio*, identificando os principais problemas enfrentados na gestão do fluxo de clientes. A análise revelou que a principal dificuldade estava na ocupação dos computadores pelos clientes que visitavam o estúdio para escolher fotos, resultando em atrasos na finalização dos trabalhos e na entrega das fotos.

4.1. ANÁLISE DAS NECESSIDADES E FUNCIONALIDADES

O sistema proposto buscou atender às principais necessidades identificadas, para isso, foram definidas as seguintes funcionalidades mais gerais do sistema na visão do cliente:

- Criação de Pastas Específicas por Cliente: esta funcionalidade facilita a organização e o acesso às fotos, permitindo uma gestão mais eficiente e personalizada dos arquivos de cada cliente. Cada cliente terá uma pasta exclusiva onde suas fotos estarão organizadas, melhorando a experiência de navegação e seleção de imagens.
- A autenticação deve ser robusta, utilizando padrões de segurança modernos para proteger os dados dos usuários e prevenir acessos não autorizados. O sistema deve suportar diferentes tipos de usuários, como fotógrafos, clientes e administradores, cada um com permissões específicas, garantindo que cada usuário acesse apenas as funcionalidades relevantes para suas necessidades.
- Cadastro dos Clientes no Sistema: O cadastro detalhado e organizado dos clientes permite um melhor gerenciamento e acesso rápido às informações necessárias para a prestação de serviços. Esta funcionalidade assegura que os dados dos clientes estejam sempre atualizados e facilmente acessíveis pelos funcionários do estúdio, otimizando o atendimento.
- *Upload* das Fotos para a Pasta do Cliente: Com esta funcionalidade, os clientes podem escolher suas fotos remotamente, a qualquer momento e de qualquer lugar, utilizando dispositivos móveis ou computadores pessoais. Isso reduz a necessidade de visitas presenciais ao estúdio, evitando

congestionamentos e a ocupação dos computadores dos editores de fotografia. Além disso, permite que clientes que residem longe do estúdio realizem a seleção de fotos no conforto de suas casas ou de qualquer outro lugar com acesso à internet.

Essas funcionalidades foram projetadas para otimizar a operação e a eficiência do estúdio, atendendo às necessidades específicas do Roberto *Photo Studio* e proporcionando uma experiência melhorada para seus clientes.

A análise dos processos operacionais dos estúdios fotográficos após a implementação do sistema revelou significativas melhorias na eficiência e na organização. A integração do sistema facilitou a criação e gestão de pastas específicas para cada cliente, permitindo um acesso mais rápido e organizado às fotografias. Isso reduziu o tempo de espera dos clientes no estúdio, contribuindo para uma experiência mais ágil e satisfatória.

A possibilidade de seleção de fotos de forma remota minimizou a necessidade de visitas presenciais, alinhando-se com as exigências sanitárias impostas pela pandemia e oferecendo uma solução segura e moderna.

O Quadro 2 contém a lista de requisitos do sistema do usuário cliente. A primeira coluna contém a identificação do requisito, a segunda coluna descreve o requisito, a terceira coluna indica a prioridade de desenvolvimento do requisito e a última coluna mostra a fonte cuja funcionalidade foi obtida.

Quadro 2: Requisitos funcionais do usuário cliente

Requisito	Descrição	Prioridade	Fonte
RF_1	O sistema deve permitir que o usuário visualize a lista de pastas e as fotografias contidas nelas.	Importante	Fotógrafo
RF_2	O sistema deve permitir que o cliente selecione a fotografia desejada para revelação e insira uma observação.	Importante	Fotógrafo
RF_3	O sistema deve permitir que os usuários visualizem seus perfis para alterar o email e o telefone.	Desejável	Fotógrafo
RF_4	O sistema deve permitir que o cliente solicite os seguintes serviços: casamento, aniversário, batizado, formatura, ensaio no estúdio, ensaio externo e outros.	Importante	Fotógrafo

RF_5	O sistema deve direcionar o cliente para o WhatsApp se a opção "outros" for escolhida entre as disponíveis no serviço.	Desejável	Fotógrafo
RF_6	O sistema deve direcionar o cliente para o agente do fotógrafo configurado para o tipo de serviço escolhido.	Importante	Fotógrafo
RF_7	O sistema deve permitir que o cliente escolha um dia e horário disponíveis na agenda do fotógrafo.	Importante	Estúdio fotográfico

Fonte: o autor (2024)

O Quadro 3 lista os requisitos do sistema do usuário colaborador. O usuário colaborador é um funcionário do estúdio fotográfico, responsável pelas interações com os clientes e realizar tarefas dentro do sistema. Esse tipo de usuário possui permissões para acessar e criar pastas de fotos, fazer upload de novas imagens, gerenciar cadastros de clientes e visualizar notificações sobre as fotos selecionadas para edição e revelação.

Quadro 3: Requisitos Funcionais do usuário colaborador

Requisito	Descrição	Prioridade	Fonte
RF_8	O sistema deve permitir cadastrar, listar, atualizar, visualizar e deletar clientes (usuários).	Importante	Estúdio fotográfico
RF_9	O sistema deve permitir cadastrar, listar, atualizar, visualizar e deletar colaboradores (usuários).	Desejável	Estúdio fotográfico
RF_10	O sistema deve permitir criar, listar, visualizar, atualizar e deletar pastas.	Importante	Estúdio fotográfico
RF_11	O sistema deve permitir fazer upload de fotografias, visualizar e deletá-las.	Importante	Estúdio Fotográfico
RF_12	O sistema deve notificar o fotógrafo quando o cliente finalizar a escolha das fotografias para revelação.	Importante	Estúdio fotográfico
RF_13	O sistema deve mostrar a lista de fotos selecionadas para revelação, com seus respectivos números e observações.	Importante	Estúdio fotográfico
RF_14	O sistema deve fornecer uma	Essencial	Estúdio

	funcionalidade para imprimir a lista de fotos selecionadas, juntamente com suas observações, nos formatos PDF e CSV.		fotográfico
RF_15	O sistema deve permitir a cópia dos nomes das fotos selecionadas para facilitar a busca no <i>Lightroom</i> .	Desejável	Estúdio fotográfico
RF_16	O sistema deve incluir uma funcionalidade para bloquear e desbloquear usuários de clientes.	Importante	Estúdio fotográfico
RF_17	O sistema deve permitir o bloqueio e desbloqueio das pastas dos clientes.	Importante	Estúdio fotográfico
RF_18	O sistema deve oferecer a opção de definir um período específico para o bloqueio das pastas dos clientes.	Essencial	Estúdio fotográfico
RF_19	O sistema deve incluir uma funcionalidade para configurar a agenda do fotógrafo, permitindo a mudança de status do dia e a indicação do tipo de serviço disponível.	Importante	Estúdio fotográfico
RF_20	O sistema deve permitir que o fotógrafo reagende um serviço.	Importante	Estúdio fotográfico
RF_21	O sistema deve enviar notificações ao fotógrafo quando um cliente solicitar um agendamento.	Essencial	Estúdio fotográfico

Fonte: o autor (2024)

O Quadro 4 contém a lista de requisitos não funcionais do sistema.

Quadro 4: Requisitos Não-Funcionais

Requisitos	Descrição	Prioridade	Fonte
RNF_22	Usabilidade: As interfaces do sistema devem ser intuitivas e de fácil compreensão para todos os usuários.	Importante	Estúdio fotográfico
RNF_23	Disponibilidade: A agenda do sistema deve permanecer disponível durante todos os dias e indicar o status dos dias como disponíveis, indisponíveis ou fechados.	Essencial	Estúdio fotográfico
RNF_24	Documentação de Uso: o sistema deve fornecer descrições detalhadas e claras sobre seu uso para orientar os usuários quanto ao cumprimento de regulamentos.	Importante	Estúdio fotográfico

RNF_25	Responsividade: as páginas web do sistema devem ser responsivas, para adaptação a diferentes tamanhos de tela e dispositivos.	Importante	Estúdio fotográfico
--------	---	------------	---------------------

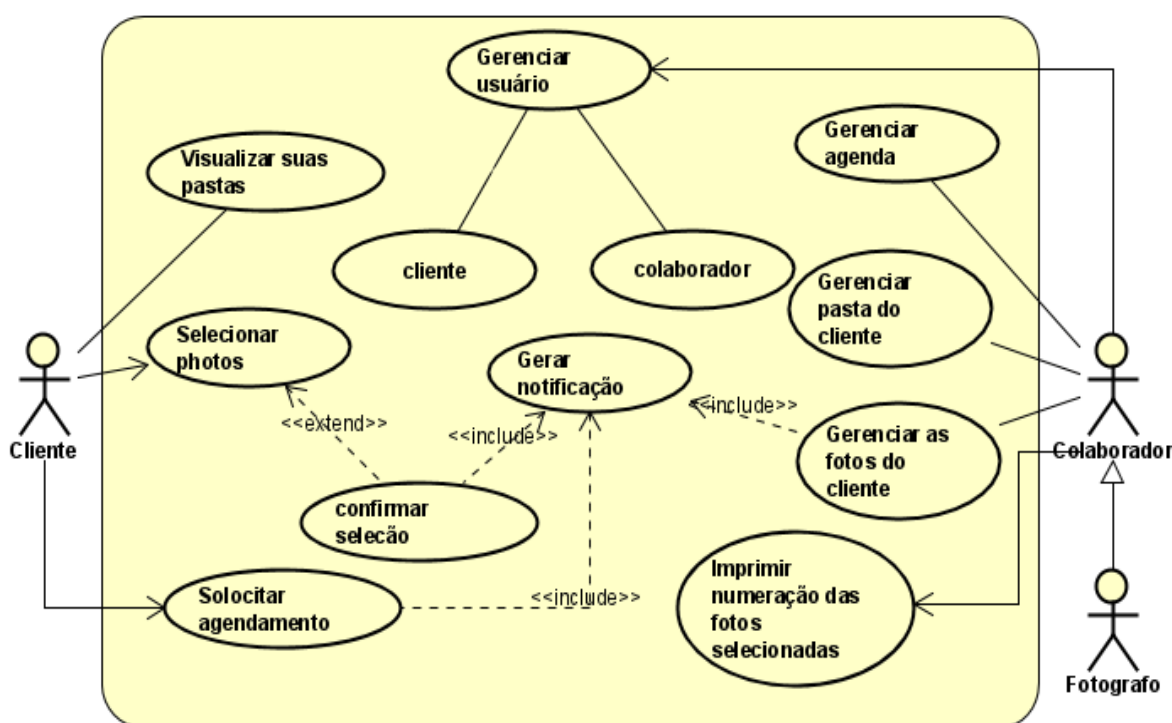
Fonte: o autor (2024)

4.2. MODELAGEM

Princípios de engenharia de software e a UML foram considerados na modelagem do sistema.

O diagrama de caso de uso do sistema representa graficamente as interações entre os usuários (clientes e administradores do estúdio) e as funcionalidades do sistema. O diagrama de caso de uso é uma parte fundamental da metodologia, pois permite a visualização clara das interações do sistema com seus usuários, ajudando a garantir que todas as necessidades do cliente sejam atendidas de maneira eficiente. A Figura 7 mostra o diagrama de casos de uso geral do sistema.

Figura 7: Diagrama de caso de uso geral

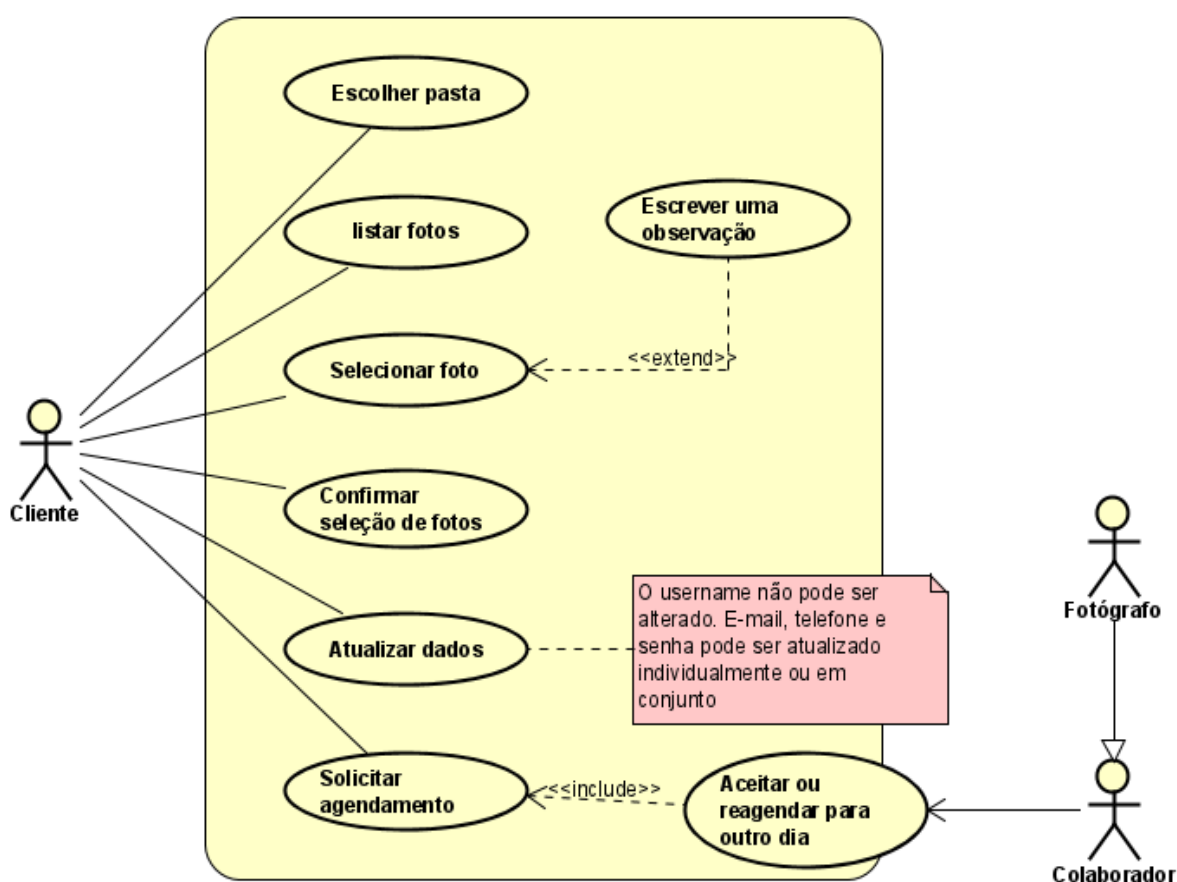


Fonte: o autor (2024)

Por meio do diagrama de casos de uso geral é possível visualizar como cada funcionalidade do sistema atende às necessidades identificadas e como os usuários interagem com ele para realizar suas tarefas de maneira eficiente e segura

A Figura 8, que detalha os casos de uso do cliente, oferece uma visão mais específica das interações e funcionalidades voltadas para os clientes, proporcionando uma compreensão aprofundada de como os usuários finais utilizam o sistema e acessam suas funcionalidades.

Figura 8: Caso de uso do cliente

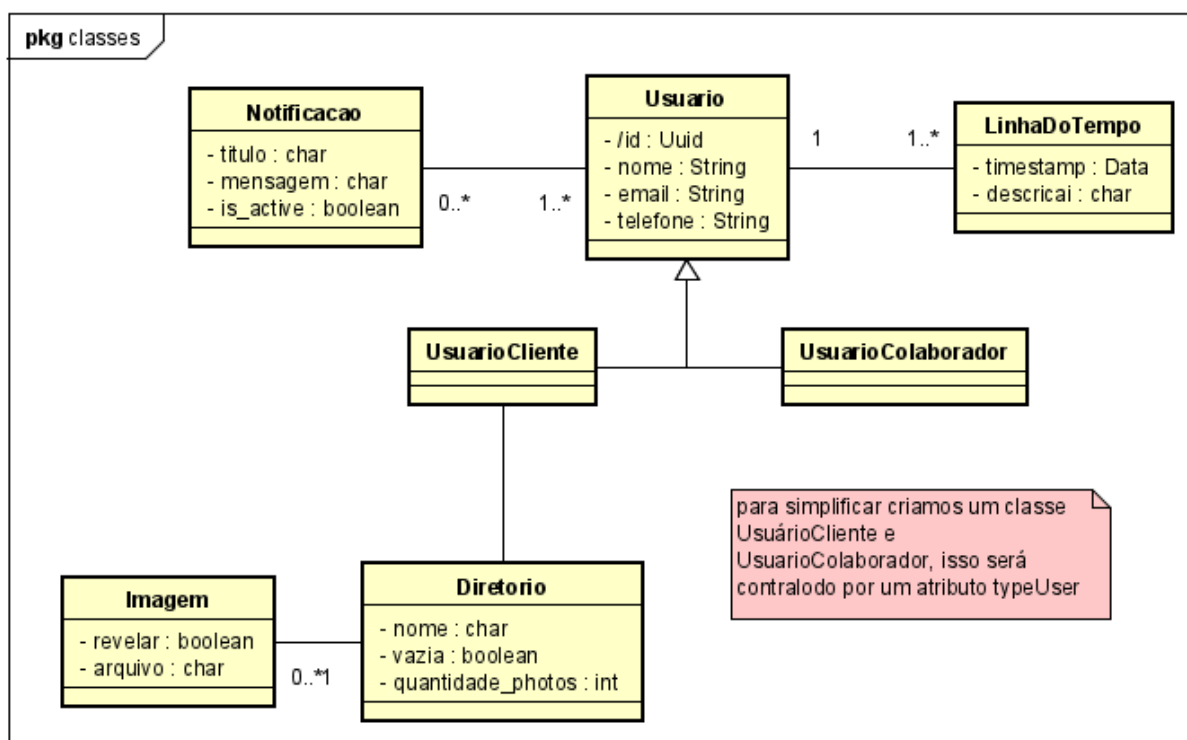


Fonte: o autor (2024)

A implementação dessas funcionalidades não apenas melhora a gestão do fluxo de clientes, mas também se adapta às exigências sanitárias impostas pela pandemia, oferecendo uma solução segura e moderna para o estúdio.

O diagrama de classes ilustrado na Figura 9 foi desenvolvido para representar as entidades e suas relações de maneira clara e precisa.

Figura 9: Diagrama de classes



Fonte: o autor (2024)

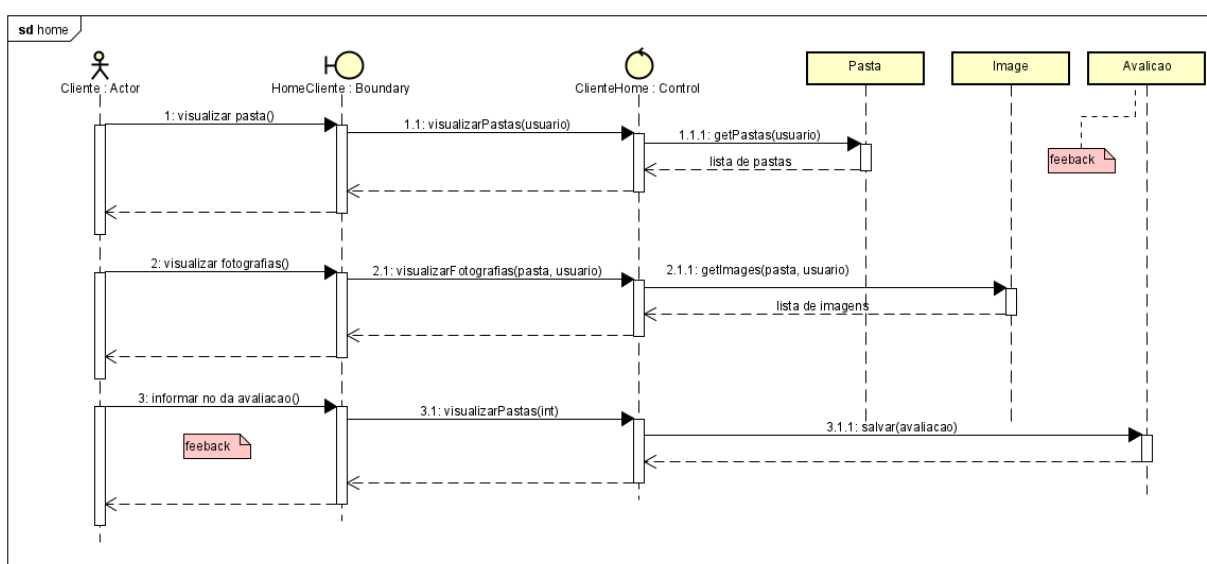
Foram realizadas reuniões com usuários do estúdio para compreender detalhadamente os problemas e necessidades. A partir dessas análises, foram criadas as entidades que serão representadas no sistema, garantindo que todas as funcionalidades essenciais sejam corretamente modeladas e implementadas. As principais classes identificadas são:

- **UsuarioCliente:** Representa os clientes do estúdio, contendo atributos como nome, e-mail e telefone. Esta classe possui uma associação com a classe LinhaDoTempo, que registra as atividades e interações do cliente com o sistema.
- **Pasta/Diretorio:** Representa as pastas específicas criadas para cada cliente. Esta classe está associada à classe UsuarioCliente e contém métodos para criar, editar e organizar as fotografias nas pastas, facilitando o gerenciamento e a busca de imagens.
- **Foto/Imagem:** Contém atributos relacionados às fotografias, incluindo o atributo "revelar", que indica se o cliente deseja ou não revelar a imagem. Esta classe também pode incluir atributos adicionais, como data de upload e comentários.

- Notificacao: Armazena dados para que os editores de fotografia saibam quais fotografias o cliente escolheu para edição e revelação. Esta classe ajuda a automatizar a comunicação entre os clientes e os editores, garantindo que os pedidos de edição sejam processados de maneira eficiente.

O diagrama de sequência do software ilustrado na Figura 10 mostra como os diferentes componentes do sistema interagem em diversos cenários, assegurando que todas as funcionalidades necessárias sejam implementadas corretamente. Também é ilustrado o principal cenário do cliente sendo representados no diagrama de sequência fluxo cliente seleção de fotografias.

Figura 10: Diagrama de sequência - cliente e seleção de fotografias

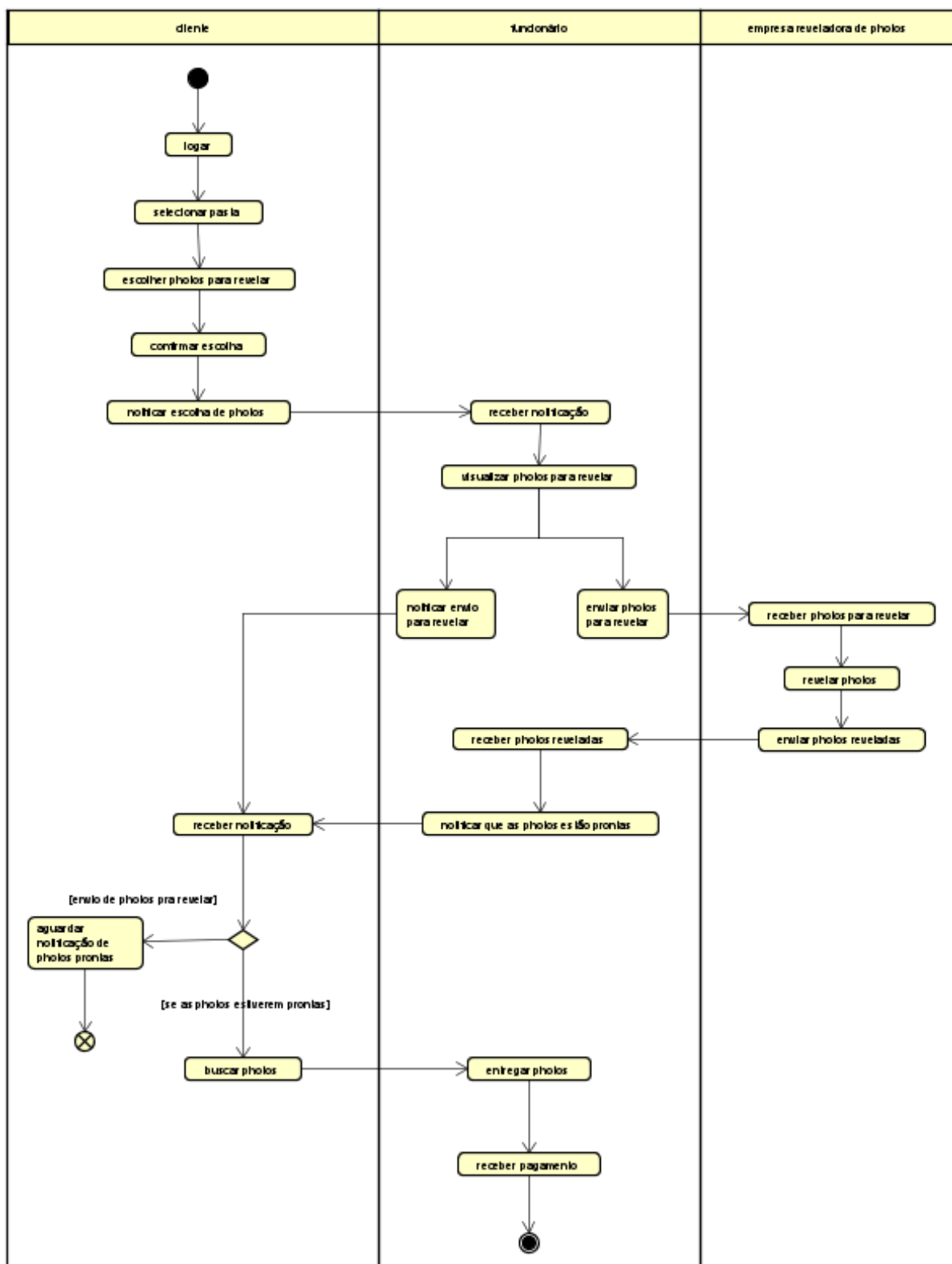


Fonte: o autor (2024)

O diagrama de sequência (Figura 10) inicia com o cliente na página inicial, onde ele pode visualizar a lista de pastas de cada evento fotografado pelo estúdio. Ao clicar em uma dessas pastas, o cliente acessa a lista de fotografias disponíveis para seleção. Durante o processo de escolha, o cliente tem a opção de adicionar comentários sobre cada imagem, especificando quaisquer edições desejadas, como a remoção de pessoas ou objetos.

O diagrama de atividade na Figura 11 ilustra o principal processo representado no diagrama de atividades, representando o fluxo de trabalho do cliente do editor de fotografia, desde o início até a conclusão das tarefas, incluindo a revelação que é terceirizada.

Figura 11: Diagrama de atividade - fluxo de tarefas



Fonte: o autor (2024)

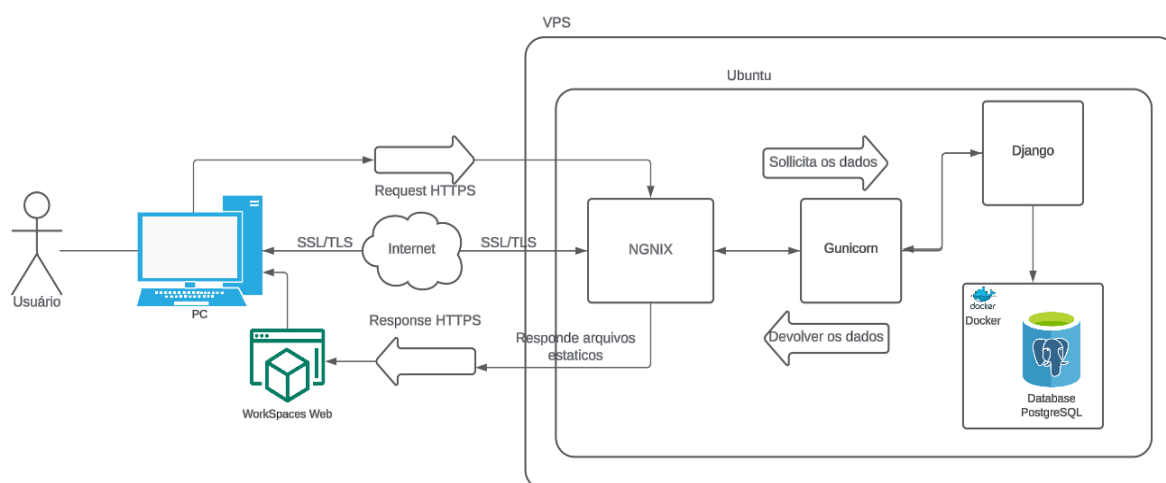
Esses diagramas UML, combinados com uma análise detalhada das necessidades do estúdio e a implementação eficiente das funcionalidades, auxiliam

para que o *software* atenda aos requisitos do *Roberto Photo Studio* e ofereça uma experiência positiva e eficiente para seus clientes.

4.3. ARQUITETURA DO SISTEMA

A Figura 12 ilustra a arquitetura do sistema Click Studium, que foi implementado em um servidor *VPS* (*Virtual Private Server*) alugado na *OVH*, utilizando o sistema operacional *Ubuntu*, utilizando *Nginx* para gerenciar as solicitações *HTTPS*, e *Gunicorn* para servir as aplicações *Django*. O banco de dados *PostgreSQL* foi escolhido para o armazenamento das informações dos usuários. O *PostgreSQL*, foi encapsulado em contêineres *Docker*, assegurando consistência entre os ambientes de desenvolvimento e produção e facilitando a escalabilidade e a orquestração do serviço.

Figura 12: Arquitetura do sistema



Fonte: o autor (2024)

O *CSS3* foi utilizado para criar telas responsivas, permitindo que o sistema se adapte a diferentes dispositivos e resoluções de tela. Isso garante que os clientes que não possuem computadores possam acessar o sistema de forma eficiente através de seus smartphones e tablets.

Para implementar a funcionalidade dinâmica do sistema, foi usado *JavaScript*, que permite responder a eventos iniciados pelo usuário e incluir efeitos que torna a página da seleção de fotografias mais interativa. O *JavaScript* foi crucial para

implementar o envio de imagens para o servidor, possibilitando que o estúdio carregue as fotos de forma rápida e segura diretamente através do navegador.

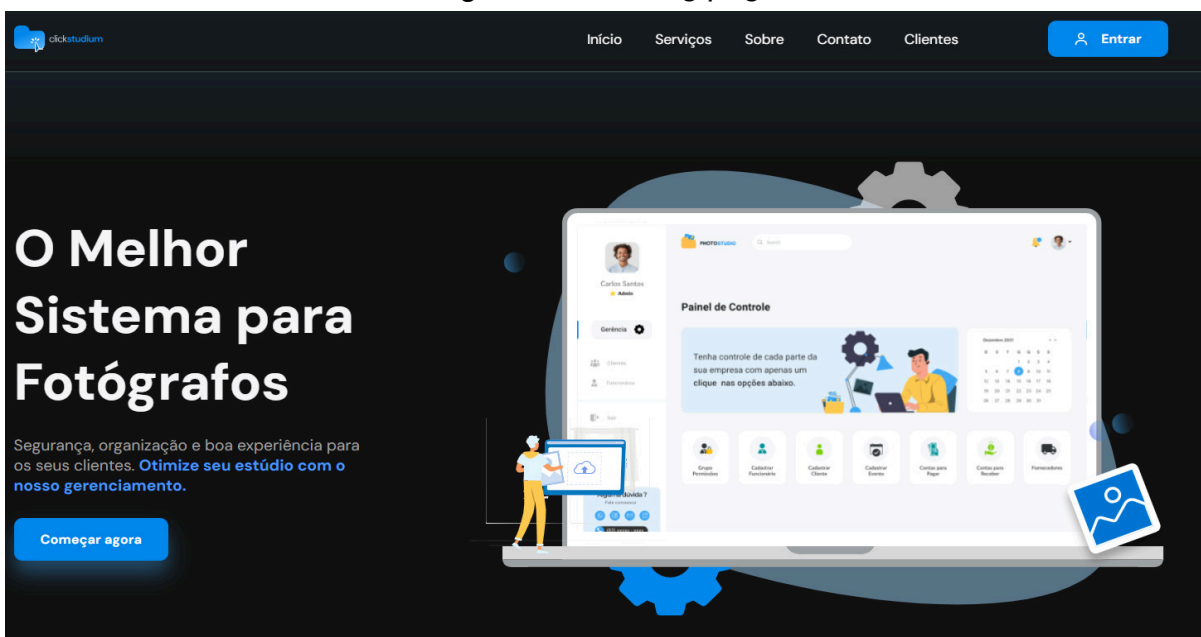
No *backend*, foi usada a linguagem *Python* devido à sua simplicidade e versatilidade. O *framework Django* foi usado para o desenvolvimento do sistema, aproveitando suas funcionalidades embutidas que facilitam a criação de aplicações *web* seguras e escaláveis. Para o gerenciamento de dados foi usado o *PostgreSQL*, conhecido por sua robustez e escalabilidade. A fim de garantir a portabilidade e a consistência do ambiente de desenvolvimento, utilizamos *Docker*, encapsulando o *software* em contêineres que incluem todas as dependências necessárias.

Para garantir a segurança das comunicações, configuramos o *Nginx* para suportar *HTTPS* com um certificado digital *SSL/TLS*. Isso assegurou que todas as comunicações entre os clientes e o servidor fossem criptografadas, protegendo assim os dados sensíveis. O *Nginx* para responder às páginas *web* e servir os arquivos estáticos, enquanto o *Gunicorn* foi configurado para trabalhar em conjunto com o *Django*, lidando com a geração e entrega dos dados dinâmicos. Essa arquitetura permitiu que o *Nginx* servisse eficientemente os arquivos estáticos, como imagens, folhas de estilo e *scripts*, enquanto delegava as requisições dinâmicas ao *Gunicorn*, que então processava as requisições do *Django*. Essa divisão de responsabilidades entre *Nginx* e *Gunicorn* melhorou significativamente o desempenho e a escalabilidade do sistema.

Para garantir a consistência do *software* em diferentes ambientes, desde o desenvolvimento até a produção, foi utilizado o *Docker* para encapsular todo o banco de dados. Essa abordagem nos permitiu criar imagens *Docker* do banco de dados *PostgreSQL*. A utilização de *Docker* também facilitou os processos de construção e implantação, permitindo que cada alteração no código fosse testada e simulando o ambiente de produção, reduzindo o risco de erros e acelerando o ciclo de desenvolvimento.

4.4. VISUALIZAÇÃO DO SISTEMA

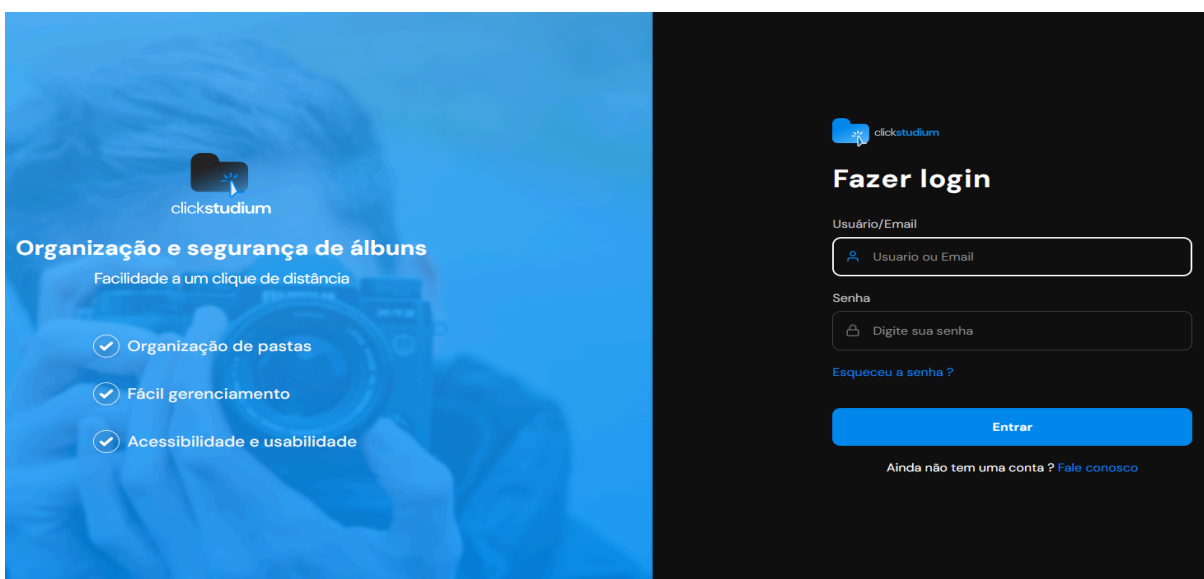
A *landing page* do *software* (Figura 13) foi desenvolvida para oferecer uma experiência intuitiva e envolvente para os visitantes. Com um design moderno e responsivo, a página se adapta perfeitamente a qualquer dispositivo, garantindo uma navegação fluida tanto em computadores quanto em smartphones.

Figura 13: *Landing page*

Fonte: o autor (2024)

O *layout* limpo e organizado destaca os principais benefícios e funcionalidades do sistema. A *landing page* também conta com seções informativas. A página se adapta perfeitamente a qualquer dispositivo, garantindo uma navegação fluida tanto em computadores quanto em *smartphones*.

A tela de *login* do *software* (Figura 14), permite que os usuários acessem a plataforma de forma rápida e segura. Nessa tela, os usuários têm a flexibilidade de utilizar tanto o seu endereço de email quanto o nome de usuário para efetuar o *login*.

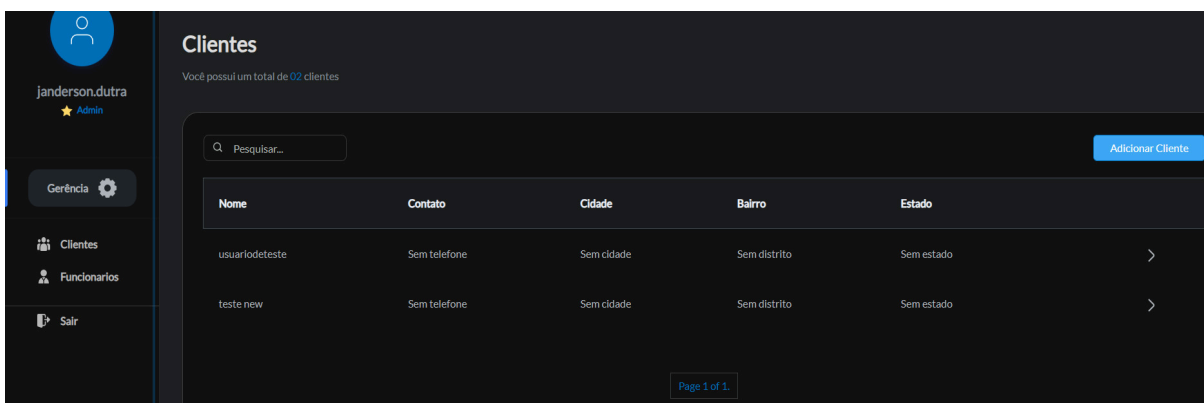
Figura 14: Tela de *login*

Fonte: o autor (2024)

Essa funcionalidade facilita o acesso, atendendo às diversas preferências dos usuários e garantindo que possam iniciar sessão sem complicações. A implementação da funcionalidade de autenticação foi desenvolvida de modo que houvesse um sistema seguro de *login*, que permitisse aos clientes autenticarem-se utilizando tanto seu endereço de email, quanto seu nome de usuário.

A lista de clientes (Figura 15) é uma funcionalidade essencial do sistema Click Studium, projetada para otimizar a gestão de informações dos clientes do estúdio fotográfico. Esta funcionalidade permite que os administradores do estúdio visualizem, cadastrem, editem e gerenciem os dados dos clientes de maneira eficiente e organizada.

Figura 15: Lista de clientes

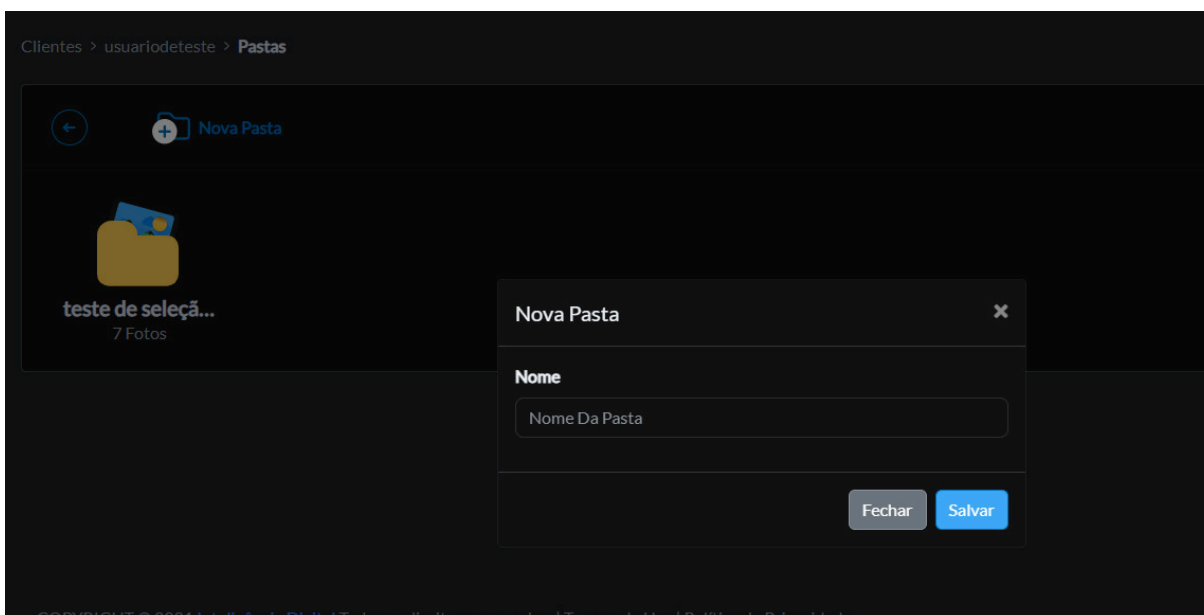


Nome	Contato	Cidade	Bairro	Estado
usuariodeteste	Sem telefone	Sem cidade	Sem distrito	Sem estado
teste new	Sem telefone	Sem cidade	Sem distrito	Sem estado

Fonte: o autor (2024)

A tela de gerenciamento de pastas (Figura 16) do sistema Click Studium foi projetada para oferecer uma interface intuitiva e eficiente. Nesta tela, os usuários podem ver a lista de pastas já cadastradas, cada uma exibindo a informação da quantidade de imagens contidas em cada pasta, além disso, na mesma tela oferece a funcionalidade de cadastro de novas pastas. Os usuários podem facilmente adicionar novas pastas sem a necessidade de navegar para uma página diferente, otimizando o fluxo de trabalho.

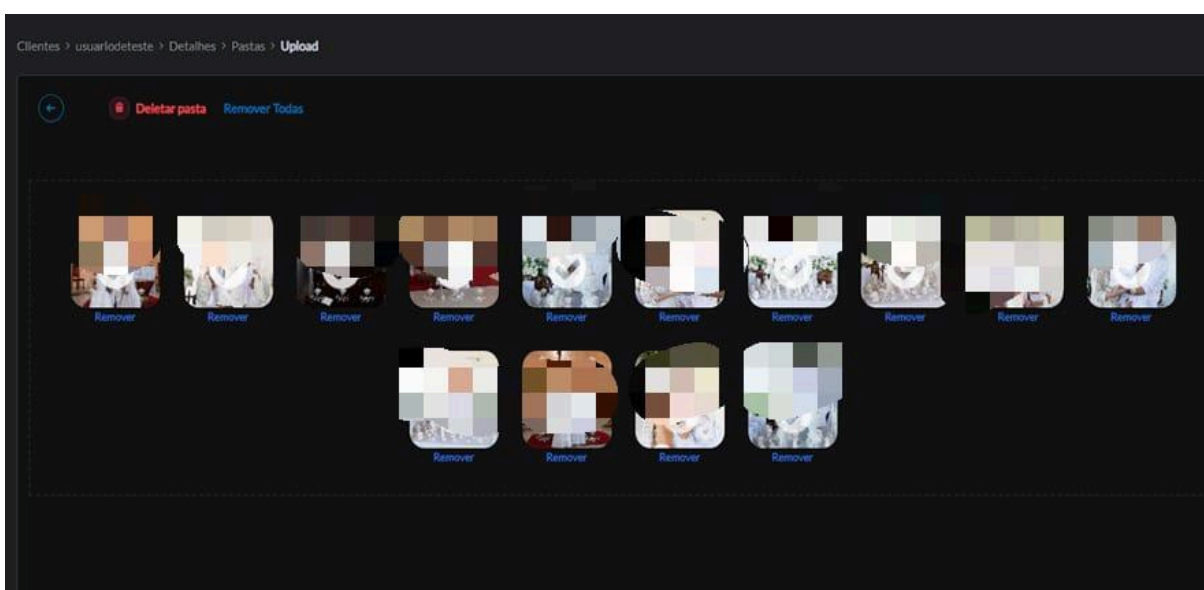
Figura 16: Lista e cadastro de pastas



Fonte: o autor (2024)

A tela de *upload* de fotos (Figura 17) do sistema Click Studium permite aos usuários carregar imagens de até 1 MB, aceitando apenas formatos de arquivo .jpg e .jpeg. Os usuários podem selecionar múltiplas fotos através de um sistema de arrastar e soltar ou por um botão de *upload*.

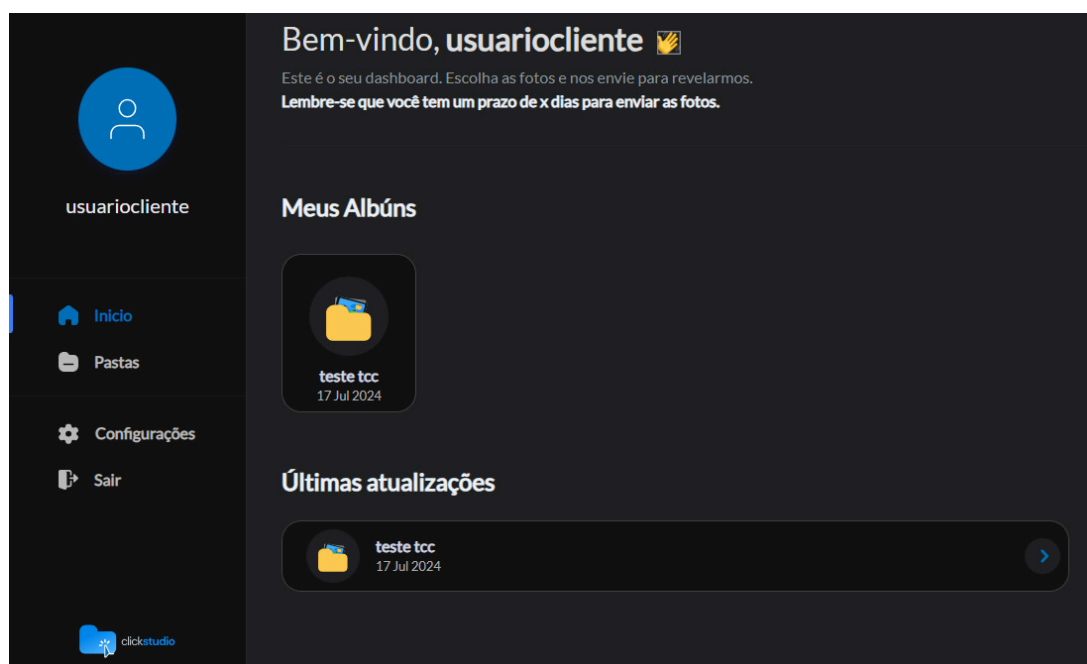
Figura 17: Tela upload de fotos



Fonte: o autor (2024)

Na página inicial (Figura 18) do cliente no sistema Click Studium, ele tem acesso à sua lista de pastas, cada uma representando um evento fotográfico específico. Ao clicar em uma dessas pastas, o cliente pode visualizar todas as fotos associadas a esse evento. Isso permite que ele navegue facilmente entre as pastas e selecione as imagens desejadas para edição, revelação ou simplesmente para visualização.

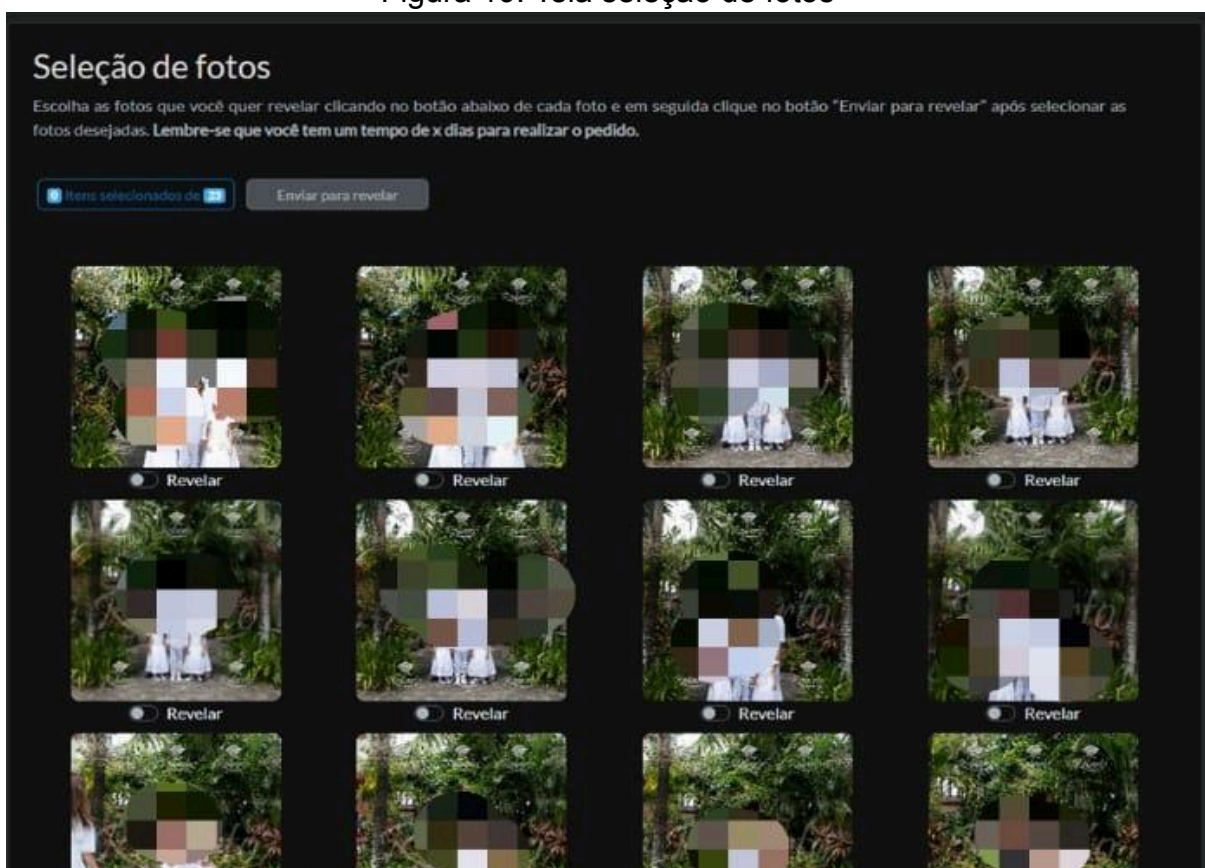
Figura 18: Página inicial do cliente



Fonte: o autor (2024)

Na tela de seleção de fotos (Figura 19) no sistema Click Studium, os clientes podem visualizar todas as fotos de um evento específico. Eles têm a capacidade de selecionar quais fotos desejam, marcando as opções correspondentes para revelação. Além disso, ao clicar em uma imagem, os clientes podem visualizá-la em tamanho grande, facilitando a visualização detalhada. Para uma experiência mais dinâmica, há também a opção de reproduzir as fotos em formato de slide, permitindo que o cliente percorra todas as imagens de maneira automática.

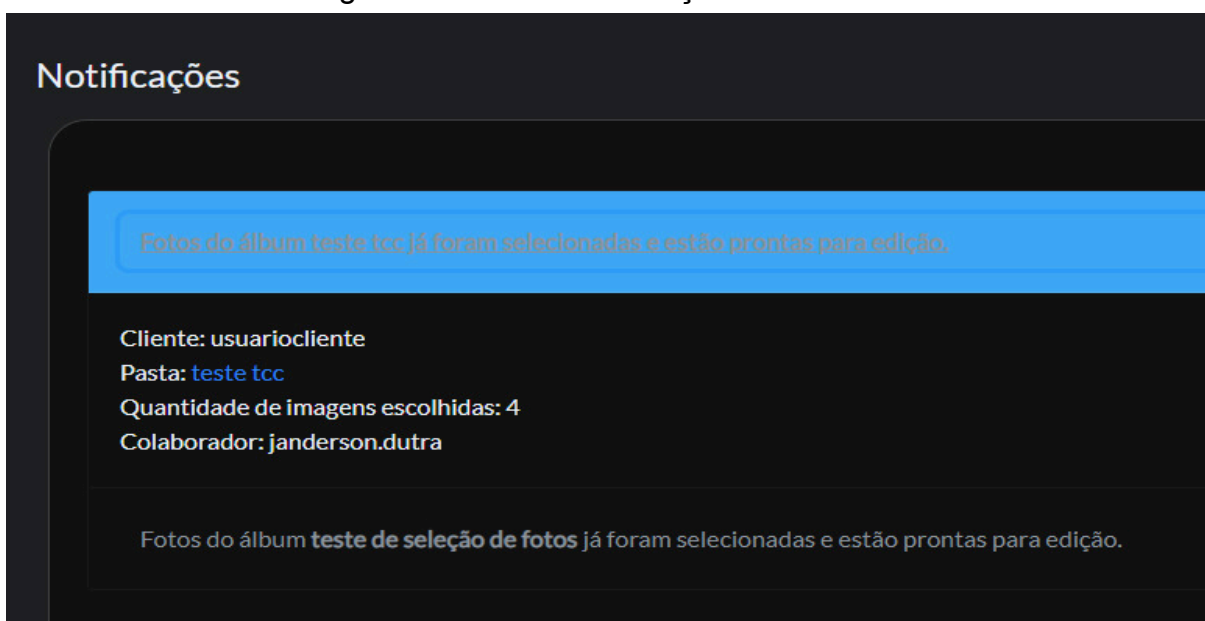
Figura 19: Tela seleção de fotos



Fonte: o autor (2024)

Na tela de notificações (Figura 20) do estúdio fotográfico no sistema Click Studium, os editores podem visualizar quais clientes já selecionaram suas fotos para edição ou revelação.

Figura 20: Tela de notificações do estúdio



Fonte: o autor (2024)

Cada notificação indica o nome do cliente, a pasta do evento fotográfico em que as fotos foram selecionadas. Isso permite uma gestão eficiente do fluxo de trabalho, garantindo que todas as solicitações dos clientes sejam atendidas de maneira organizada e oportuna.

5. CONSIDERAÇÕES FINAIS

Os resultados obtidos neste trabalho demonstram que a implementação do *software* proporcionou benefícios significativos para o estúdio fotográfico, melhorando não apenas a eficiência operacional e a qualidade dos serviços, mas também elevando a satisfação e a experiência geral dos clientes. A adoção de soluções tecnológicas como o Click Studium não apenas se mostrou crucial para a adaptação do estúdio às demandas modernas, como também abriu novas oportunidades para inovação e crescimento dentro do setor fotográfico.

Além disso, a capacidade de os clientes selecionarem e comentarem sobre as fotografias remotamente através do sistema também otimizou os fluxos de trabalho internos. Os editores de fotografia puderam priorizar suas tarefas com base nas preferências e necessidades dos clientes, resultando em um processo de pós-produção mais eficiente e alinhado com as expectativas dos usuários finais.

A adoção do *software* teve um impacto direto na qualidade dos serviços oferecidos pelos estúdios fotográficos. A organização aprimorada das imagens e a facilidade de acesso proporcionaram aos clientes uma experiência mais personalizada e satisfatória. A capacidade de os clientes indicarem preferências específicas, como remoção de pessoas ou objetos de imagens, foi um diferencial positivo, permitindo ajustes precisos conforme as solicitações dos clientes.

A análise entre o antes e o depois da implementação foi realizada em conversas informais com os stakeholders do estúdio, que mostraram uma redução significativa no tempo necessário para conclusão dos projetos fotográficos. Isso não apenas melhorou a eficiência operacional do estúdio, mas também contribuiu para a fidelização dos clientes, que perceberam uma maior rapidez na entrega final de seus pedidos.

A avaliação da satisfação dos clientes em relação ao uso do *software* revelou uma resposta positiva geral. A possibilidade de interagir remotamente com o sistema para selecionar e comentar sobre as fotografias foi bem recebida pelos clientes, que destacaram a conveniência e a acessibilidade do processo. A maioria dos clientes expressou maior satisfação com a experiência de escolha de suas fotos, indicando uma melhoria significativa na interação com os serviços oferecidos pelo estúdio fotográfico.

As conversas informais dos clientes do estúdio com os *stakeholders* do estúdio, colaboraram para essas percepções, fornecendo *insights* valiosos sobre as áreas de melhoria contínua do sistema. Permitiu identificar pontos fortes e áreas de oportunidade, orientando ajustes futuros para maximizar ainda mais a eficiência e a satisfação do cliente com o *software*.

5.1. TRABALHOS FUTUROS

Para ampliar a funcionalidade do sistema Click Studium e aumentar ainda mais a eficiência dos estúdios fotográficos, pretende-se desenvolver um sistema de agendamento. Esse sistema permitirá que os clientes agendem sessões fotográficas diretamente pelo portal, visualizando a disponibilidade em tempo real e selecionando horários que melhor se adequem às suas agendas. A introdução do agendamento online não apenas reduzirá a carga administrativa do estúdio, mas também poderá melhorar a experiência do cliente, proporcionando conveniência e flexibilidade.

Para garantir a escalabilidade e a modularidade do novo sistema de agendamento, a arquitetura pode ser baseada em *microservices*. Essa abordagem permitirá que diferentes componentes do sistema funcionem de maneira independente, facilitando a manutenção e a atualização de cada serviço sem afetar os outros. Os *microservices* podem incluir serviços para gerenciamento de usuários, gerenciamento de horários, notificações e pagamentos. Essa separação de preocupações simplificará o desenvolvimento e a implantação contínua de novas funcionalidades, além de melhorar a resiliência do sistema como um todo.

Um aspecto crucial da arquitetura de *microservices* é a implementação de uma API (*Application Programming Interface*) robusta. A API atuará como um intermediário entre os *microservices* e os clientes, permitindo que as aplicações *frontend* e outros serviços interajam com o sistema de maneira eficiente e segura. A API fornecerá endpoints para operações como criação, atualização e cancelamento de agendamentos, autenticação de usuários e consulta de disponibilidade. A segurança será uma prioridade, utilizando práticas como autenticação baseada em *tokens* e criptografia de dados para proteger as informações dos usuários.

O desenvolvimento do frontend do sistema será realizado utilizando *Angular*, um *framework* de desenvolvimento *web* altamente eficiente e escalável. *Angular* permitirá a criação de uma interface de usuário interativa e responsiva, melhorando

significativamente a experiência do usuário. Para hospedar e gerenciar a infraestrutura do sistema, será utilizada a nuvem da *AWS (Amazon Web Services)*. A *AWS* proporcionará escalabilidade, segurança e confiabilidade, garantindo que o sistema possa crescer conforme a demanda e que os dados dos clientes sejam protegidos com os mais altos padrões de segurança.

REFERÊNCIAS

AGÊNCIA CNI DE NOTÍCIAS. Crise do novo coronavírus promove desafios para a indústria e para o Brasil. Portal da Indústria, 23 mar. 2020. Disponível em: <https://noticias.portaldaindustria.com.br/noticias/economia/crise-do-novo-coronavirus-promove-desafios-para-a-industria-e-para-o-brasil/>. Acesso em: 13 maio de 2024.

ALEXEEV, Andrew. The Architecture of Open Source Applications (Volume 2) nginx. 2012. Disponível em: <https://aosabook.org/en/v2/nginx.html>. Acesso em 17 de julho de 2024.

ALVES, G. H. F.; PETRUCCELLI, E. E. VISÃO GERAL E FUNCIONALIDADES DA BIBLIOTECA JAVASCRIPT SAPUI5 E SUA UTILIZAÇÃO NO AMBIENTE CORPORATIVO. Revista Interface Tecnológica, [S. l.], v. 16, n. 1, p. 31–42, 2019. Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/view/544>. Acesso em: 30 maio de 2024.

BERNERS-LEE, Tim. The world-wide web. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/179606.179671>. Acesso em: 20 maio de 2024.

BERNIER, Robert. Segurança Total em um Banco de Dados PostgreSQL: Disponibilidade para o primeiro ataque. IBM, PostgreSQL Business Intelligence Analyst, Medio Systems, 16 dez. 2009. Disponível em: <https://pdfslide.tips/documents/seguranca-total-em-um-banco-de-dados-postgresql.html>. Acesso em: 08 junho de 2024.

CARDOSO, Natanael Silva; BISPO, Thamirys Martha da Silva. Um estudo comparativo entre os principais frameworks de desenvolvimento web em linguagem python. Orientador: João Ferreira de Santanna Filho. 2019. 70 f. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) – Universidade Federal Rural da Amazônia, Belém, 2019. Disponível em: <https://bdta.ufra.edu.br/jspui/bitstream/123456789/541/1/Um%20Estudo%20Comparativo%20Entre%20os%20Principais%20Frameworks%20de%20Desenvolvimento%20Web%20em%20Linguagem%20Python.pdf>. Acesso em: 03 junho de 2024.

CARVALHO, Vinícius. PostgreSQL: Banco de dados para aplicações web modernas. São Paulo: Casa do Código, 2017. Disponível em: ASIN B06XKLDLL5.

CORREIA, Arthur Sampaio Perico. JavaScript Academy – Um Arcabouço para Lições em JavaScript. 2019. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/xmlui/bitstream/handle/riufcg/20695/ARTHUR%20SAMPAIO%20PERICO%20CORREIA%20-%20TCC%20CI%3%8ANCIA%20DA%20COMPUTA%3%87%3%83O%202019.pdf?sequence=3&isAllowed=y>. Acesso em: 30 maio de 2024.

COSTA, C. C. R. (2022). A Tecnologia Na Experiência Do Cliente (CX): Uma Revisão da Literatura. Desafio Online, 10(3). Disponível em:

<https://periodicos.ufms.br/index.php/SIMSAD/article/view/13362/9219>. Acesso em: 10 maio de 2024.

COSTA, C. C. R. (2022). Experiência do Consumidor e Inteligência Artificial: Uma Revisão da Literatura. *Desafio Online*, 10(3). Disponível em: <https://pdfs.semanticscholar.org/73c2/1f8a860c3789b500596a72c9f767f7a256.pdf>. Acesso em: 10 maio de 2024.

COSTA, Eduardo Lima da. Sistema supervisorio para dispositivos modbus tcp/ip via aplicação web. 2022. 62 p. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Recife, 2022. Orientador: Douglas Contente Pimentel Barbosa.

CUNHA, Aline Cristina Olímpio Ventura da et al. A fotografia como extensão da memória: um estudo de caso sobre álbuns de casamento. *Revista H-Tec Humanidades e Tecnologia*, v. 4, n. 1, p. 6-187, jan./jun., 2020. Disponível em: <https://www.fateccruzeiro.edu.br/revista/index.php/htec/article/view/129>. Acesso em: 13 maio de 2024.

DA SILVA GUNTZEL, P. H., Leal Nunes Da Silva, L., & Padilha Moreira, J. (2019). JAVASCRIPT UM DOS 3 PILARES DA WEB. *SEMINÁRIO DE TECNOLOGIA GESTÃO E EDUCAÇÃO*, 1(2), 6. Recuperado de <https://raam.alcidesmaya.edu.br/index.php/SGTE/article/view/16>. Acesso em: 30 maio de 24.

DIAS, Luís Antonio Queiroz. Um ambiente seguro para aplicação com Framework web Django. 2020. 54 f. Monografia (Curso Superior de Tecnologia em Segurança da Informação) – Faculdade de Tecnologia de Americana, Centro Estadual de Educação Tecnológica Paula Souza, Americana, 2020. Orientador: Prof. Esp. Marcus Vinícius Lahr Giraldi. Disponível em: http://ric.cps.sp.gov.br/bitstream/123456789/8292/1/1S2020_Lu%C3%ADs%20Antonio%20Queiroz%20Dias_OD0984.pdf. Acesso em: 03 junho de 2024.

DOS SANTOS, M. Como funciona a Internet e a World Wide Web. Disponível em: <https://pensandonaweb.com.br/como-funciona-a-internet-e-a-world-wide-web/>. Acesso em: 5 jul. 2024.

DOS SANTOS, M. Aprendendo HTML, a linguagem da web: Disponível em: <https://pensandonaweb.com.br/content/images/2014/Aug/html-hierarchy.png>. Acesso em: 5 jul. 2024b.

DUARTE, Tiago Daltro. Sistema integrado de monitoramento de consumo de água residencial. 2023. 82 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Instituto Federal de Educação da Paraíba, Unidade Acadêmica de Processos Industriais, João Pessoa, 2023. Orientador: Profº D.r Lincoln Machado de Araújo.

EUROPEU, Pedro Fillype de Souza. Desenvolvimento de um sistema para web para a gestão de pontos do sistema de medição para faturamento. 2022. 54 f. Trabalho

de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Instituto Federal de Educação da Paraíba, Unidade Acadêmica de Processos Industriais, João Pessoa, 2022. Orientador: Profº DSc. Álvaro de Medeiros Maciel.

FLATSCHART, Fábio. HTML 5: Embarque Imediato. 1ª ed. Rio de Janeiro: Brasport, 2011. 256 p. ISBN 978-8574524771.

FOROUZAN, Behrouz A.; OLIVEIRA, Jonas Santiago de. Comunicação de Dados e Redes de Computadores. 4ª ed. Tradução de: Ariovaldo Griesi. Porto Alegre: AMGH, 2008. 1134 p. ISBN 978-85-63308-47-4. ISBN-13: 978-8586804885.

KÖCHE, J. C. Fundamentos de metodologia científica. Petrópolis: Editora Vozes, 2016. Disponível em:
http://www.adm.ufrpe.br/sites/ww4.deinfo.ufrpe.br/files/Fundamentos_de_Metodologia_Cienti%CC%81fica.pdf. Acesso em: 21 fev. 2023.

KUROSE, Jim; ROSS, Keith. Redes de Computadores e a Internet: Uma Abordagem Top-Down. 6ª ed. Tradução de: Daniel Vieira. São Paulo: Pearson Universidades, 2013. 656 p. ISBN: 978-85-430-1443-2. ISBN-13: 978-8581436777.

LUCAS DA SILVA PEREIRA; JOÃO PADILHA MOREIRA. SISTEMA WEB PARA AGENDAMENTOS DE UM SALÃO DE BELEZA. SEMINÁRIO DE TECNOLOGIA GESTÃO E EDUCAÇÃO, v. 2, n. 2, 22 out. 2020. Disponível em:
<https://raam.alcidesmaya.edu.br/index.php/SGTE/article/view/247>. Acesso em: 03 junho de 2024.

Malta, D. C., Szwarcwald, C. L., Barros, M. B. de A., Gomes, C. S., Machado, Í. E., Souza Júnior, P. R. B. de, ... & Azevedo, L. O. (2020). A pandemia da COVID-19 e as mudanças no estilo de vida dos brasileiros adultos: um estudo transversal, 2020. *Epidemiologia e Serviços de Saúde*, 29(4), e2020407. Disponível em:
<https://www.scielo.br/j/ress/a/VkvxmKYhw9djmrNBzHsvrx/>. Acesso em: 13 mai. 2024.

MIYAGUSKU, Renata. Curso Prático de SQL: guia de referência completo para usar a linguagem SQL nos bancos de dados: MS SQL Server, Oracle, PostgreSQL, MySQL. São Paulo: Digerati Books, 2008. 96 p. ISBN 978-85-60480-85-2.

NETTO, H. V. VIRTUALIZAÇÃO EM NÍVEL DE SISTEMA USANDO O DOCKER. Anais da Mostra de Ensino, Pesquisa, Extensão e Cidadania (MEPEC) - ISSN 2596-0954, [S. l.], v. 3, p. 120, 2019. Disponível em:
<https://publicacoes.ifc.edu.br/index.php/MEPEC/article/view/425>. Acesso em: 15 junho de 2024.

NEVES JUNIOR, V. C.; SOUSA, J. G.. Criação de um ambiente web de alto desempenho para o Portal CEULP/ULBRA. In: ENCOINFO - Congresso de Computação e Tecnologias da Informação, 14., 2012, Palmas - TO. Anais [...]. Palmas - TO: CEULP/ULBRA, 2012. p. 53 - 62. ISSN e-ISSN: 2447-0767 versão online. Disponível em:
<https://ulbra-to.br/encoinfo/edicoes/2012/artigos/criacao-de-um-ambiente-web-de-alto-desempenho-para-o-portal-ceulp-ulbra/>. Acesso em: 16 jun. 2024

PAIVA, Fábio Augusto Procópio de; NASCIMENTO, João Maria Araújo do; MARTINS, Rodrigo Siqueira; SOUZA, Givanildo Rocha de. Introdução a Python com Aplicações de Sistemas Operacionais. Projeto gráfico, diagramação e capa: Charles Bamam Medeiros de Souza; revisão textual: Rodrigo Luiz Silva Pessoa. Natal: IFRN, 2019. 240 p. ISBN 978-65-86293-38-8. Disponível em: <https://memoria.ifrn.edu.br/handle/1044/2090>. Acesso em: 04 junho 2024.

PEIXOTO, J. G. M. Intercom, Rev. Bras. Ciênc. Comun. (2020). Experiências inovadoras no fotojornalismo contemporânea: o caso Innovative Storytelling do World Press Photo Digital Storytelling Contest. SciELO Brasil. Disponível em: <https://www.scielo.br/j/intercom/a/BjL76F9pvJyYDnDbRpmPnZg/#>. Acesso em: 10 maio de 2024.

RABETTI, Rafael dos Santos. Comparativo entre as metodologias atômica e tradicional no desenvolvimento de CSS. 2019. Disponível em: https://bdm.unb.br/bitstream/10483/23679/1/2019_RafaelDosSantosRabetti_tcc.pdf. Acesso em: 26 maio de 2024.

RHODES, Brandon; GOERZEN, John. Programação de redes com Python: Guia abrangente de programação e gerenciamento de redes com Python 3. Tradução de: Aldir José Coelho Corrêa da Silva. Rio de Janeiro: Novatec Editora, 2015. 552 p. ISBN: 978-85-7522-437-3.

ROMERO, Daniel. Containers com Docker: Do desenvolvimento à produção. São Paulo: Casa do Código, 2015. Disponível em: ASIN B019NJB50C.

SANTIAGO, Cynthia Pinheiro; VERAS, Nécio Lima; ARAGÃO, Anderson Passos de; CARVALHO, Daniel Albuquerque; AMARAL, Luciana Alves. Desenvolvimento de sistemas Web orientado a reuso com Python, Django e Bootstrap. Sociedade Brasileira de Computação, 10 set. 2020. Disponível em: https://scholar.google.com.br/citations?view_op=view_citation&hl=pt-BR&user=5QOXuK8AAAAAJ&citation_for_view=5QOXuK8AAAAAJ:9yKSN-GCB0IC. Acesso em: 03 junho de 2024.

SANTOS, Brena; ENDO, Patricia Takako; SILVA, Francisco Airton. Uma Avaliação de Desempenho de Contêineres Docker Executando Diferentes SGBDs Relacionais. In: WORKSHOP EM DESEMPENHO DE SISTEMAS COMPUTACIONAIS E DE COMUNICAÇÃO (WPERFORMANCE), 2019. , 2019, Belém. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019. ISSN 2595-6167. DOI:. Disponível em <https://sol.sbc.org.br/index.php/wperformance/article/view/6467/6363>. Acesso em: 15 junho de 2024.

SILVA, Draenne Micarla dos Santos. Interface Web de um Sistema de Gerenciamento de Plantéis da Espécie Bovina. 2020. Disponível em: <https://memoria.ifrn.edu.br/bitstream/handle/1044/1902/INTERFACE%20WEB%20DE%20UM%20SISTEMA%20DE%20GERENCIAMENTO%20DE%20PLANTEIS%20DA%20ESP%20C3%89CIE%20BOVINA.pdf?sequence=1&isAllowed=y>. Acesso em: 26 maio de 2024.

SILVA, Rogério Oliveira da. Linguagem de programação Python. Revista Tecnologias em Projeção, v. 10, n. 1, 2019. Disponível em: <https://revista.projecao.br/index.php/Projecao4/article/view/1359>. Acesso em: 03 junho de 2024.

SILVA, Mateus da; HUSEMANN, Ronaldo; MALHEIROS, Marcelo de Gomensoro. COMPARAÇÃO DE DESEMPENHO DAS ARQUITETURAS SOC DE CPU ARM CORTEX A57 E INTEL X86 COMO SERVIDORES DOS SOFTWARES APACHE E NGINX. Revista Destaques Acadêmicos, [S. l.], v. 9, n. 4, 2017. DOI: 10.22410/issn.2176-3070.v9i4a2017.1670. Disponível em: <https://univates.br/revistas/index.php/destaques/article/view/1670>. Acesso em: 16 jun. 2024.

SILVESTRE, Isabela Mariane; BATISTELA, Karoline Kirner. SisCan: Sistema web para o processo de adoção e doação de animais. 2019. Disponível em: https://ric.cps.sp.gov.br/bitstream/123456789/3928/1/20192S_SILVESTREIsabelaMarianeBATISTELAKarolineKirner_OD0711.pdf. Acesso em: 26 maio 2024.

SOARES, Edilva Carvalho. Projeto de intervenção pedagógica: construindo uma aplicação web com o framework Django. 2022. 19 f. Trabalho de Conclusão de Curso (Especialização em Docência para Educação Profissional e Tecnológica) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cabedelo, 2022. Disponível em: <https://repositorio.ifpb.edu.br/bitstream/177683/2540/1/TCC%20Edilva.pdf>. Acesso em: 03 junho de 2024.

SOARES, Aden. Sistema WEB para gerenciamento de projetos baseado na metodologia OPTIMUS. 2022. 70f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) - Centro de Tecnologia, Universidade Federal do Rio Grande do Norte, Natal, 2022. Orientador: Francisco José Targino Vidal. Coorientador: Flaviano Costa Dantas. Disponível em: <https://repositorio.ufrn.br/handle/123456789/48812>. Acesso em: 03 junho de 2024.

SCHROEDER, Ricardo. HTML5, um novo desenvolvimento para a web. Revista Caminhos, Rio do Sul, a. 3, n. 5, p. 25-39, jul./set. 2012. Dossiê Gestão. Disponível em: <https://siteunidavi.s3.amazonaws.com/revistaCaminhos/ano3.pdf#page=25>. Acesso em: 20 maio de 2024.

TANENBAUM, Andrew S.; WETHERALL, David. Redes de Computadores. 5ª ed. Tradução de: Daniel Vieira. São Paulo: Pearson Universidades, 2011. 600 p. ISBN: 978-8576059240.

TAMANHA, Lucas Koiti Geminiani; DOMINGUES, Guilherme Luis; BITTENCOURT, Luiz Fernando. Algoritmos para Balanceamento de Carga no NGINX. Relatório Técnico - IC-PFG-21-49. Projeto Final de Graduação. 2021 - Dezembro. Universidade Estadual de Campinas, Instituto de Computação. Disponível em: <https://www.ic.unicamp.br/~reltech/PFG/2021/PFG-21-49.pdf>. Acesso em: 16 jun. 2024.


TORRES, V. M. (2018). HTML E SEUS COMPONENTES. Revista Ada Lovelace, 2, 99–101. Recuperado de <https://anais.unievangelica.edu.br/index.php/adalovelace/article/view/4652>. Acesso em: 20 maio de 2024.

VIAL, G. Understanding digital transformation: A review and a research agenda Journal of Strategic Information Systems. Elsevier B.V. 2019

Reis, R. & Reis, D. (2021). A pandemia de covid-19 e o processo de transformação digital. Revista Processando o Saber. Fatec Praia Grande. Disponível em: <https://scholar.archive.org/work/wzsgs4rmendaliahbv3pnofvxa/access/wayback/https://www.fatecpg.edu.br/revista/index.php/ps/article/download/213/151>. Acesso em: 10 maio de 2024.

VIEIRA, N. (2020). Do nascimento às selfies: como a tecnologia transformou a fotografia. Canaltech. Disponível em:

<https://canaltech.com.br/entretenimento/do-nascimento-as-selfies-como-a-tecnologia-transformou-a-fotografia-170108/>. Acesso em: 10 maio de 2024.

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Cajazeiras - Código INEP: 25008978
	Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
	CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

Documento Digitalizado Ostensivo (Público)

TCC

Assunto:	TCC
Assinado por:	Jeronimo Gabriel
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Jerônimo Gabriel da Silva, **ALUNO (201712320058) DE LICENCIATURA EM COMPUTAÇÃO E INFORMÁTICA - EAD - CAJAZEIRAS**, em 15/09/2024 13:15:06.

Este documento foi armazenado no SUAP em 15/09/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1249122

Código de Autenticação: 1e8bef8ac6

