

**MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA FARROUPILHA
CAMPUS SANTO ÂNGELO**

**QUEBRA GALHO
FACILITANDO APPs**

TRABALHO DE CONCLUSÃO DE CURSO

Argel Guimarães de Almeida

**Santo Ângelo, RS, Brasil.
2023**

**QUEBRA GALHO
FACILITANDO APPs**

por

Argel Guimarães de Almeida

**Monografia apresentada ao Instituto Federal de Educação Ciência e Tecnologia
Farroupilha, como requisito parcial para obtenção do título de Tecnólogo em Sistemas
para Internet.**

Orientadora: Marta Breunig Loose

Santo Ângelo, RS, Brasil

2023

**Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação Ciência e Tecnologia Farroupilha**

A Comissão Examinadora, abaixo assinada,
aprova a Monografia

QUEBRA GALHO

FACILITANDO APPs

elaborada por
Argel Guimarães de Almeida

como requisito parcial para obtenção do título de
Tecnólogo em Sistemas para Internet

COMISSÃO EXAMINADORA

Marta Breunig Loose, Me.
(Orientadora)

John Soldera, Dr. (IFFar)

Marcos Vinicius Pizzuti Nascimento, Esp. (IFFar)

Conceito Final: _____

Santo Ângelo, 14 de dezembro de 2023.

RESUMO

QUEBRA GALHO

FACILITANDO APPs

Autor: Argel Guimarães de Almeida

Orientadora: Marta Breunig Loose

Data e Local da Defesa: Santo Ângelo, 14 de dezembro de 2023.

Os aplicativos se tornaram uma parte essencial do nosso cotidiano, oferecendo uma gama de funcionalidades que simplificam tarefas, nos mantêm conectados e proporcionam entretenimento. Foi observado que o mesmo aplicativo pode variar sua estrutura, funções conforme troca-se de plataforma, sejam elas web, desktop ou mobile. Pensando nisso, este trabalho apresenta um fórum colaborativo, batizado com o nome de “quebra galho”, que tem como objetivo ajudar a solucionar problemas e dúvidas relacionadas a aplicativos, oferecendo um espaço para a comunidade se ajudar mutuamente. O trabalho foi desenvolvido em arquitetura MVC, sendo utilizado o *framework* Laravel, que é derivado do PHP, como linguagem *back-end*, e para a *front-end* o bootstrap e a codificação do banco de dados foi desenvolvida em MySQL juntamente com as funções do Laravel. A plataforma é de fácil uso e conta com uma interface intuitiva, o que facilita a navegação e a busca por informações. Com isso, o fórum se torna uma ferramenta útil para quem busca aprender mais sobre tecnologia e aplicativos, além de proporcionar uma interação social entre usuários com interesses em comum. O projeto foi desenvolvido com o intuito de fornecer uma solução prática e colaborativa para a comunidade de tecnologia, tendo em vista a importância cada vez maior dos aplicativos em nossas vidas.

Palavras-chave: aplicativos; sistema web; php; laravel; mvc;

ABSTRACT

Apps have become an essential part of our daily lives, offering a range of features that simplify tasks, keep us connected and provide entertainment. It was observed that the same application can vary its structure and functions depending on the platform being changed, whether web, desktop or mobile. With this in mind, this work presents a collaborative forum, called “quebra galho”, which aims to help resolve problems and doubts related to applications, offering a space for the community to help each other. The work was developed in MVC architecture, using the Laravel framework, which is derived from PHP, as the back-end language, and for the front-end the bootstrap and database coding was developed in MySQL together with the functions of the Laravel. The platform is easy to use and has an intuitive interface, which makes navigation and searching for information easier. As a result, the forum becomes a useful tool for those looking to learn more about technology and applications, as well as providing social interaction between users with common interests. The project was developed with the aim of providing a practical and collaborative solution for the technology community, given the increasing importance of applications in our lives.

Keywords: applications; web system; php; laravel; mvc;

AGRADECIMENTOS

Passando aqui pra agradecer essa galera que arreventou a boca do balão nessa jornada do meu TCC! Foi um caminho cheio de aprendizados, mas graças a vocês eu consegui ultrapassar cada obstáculo.

Agradeço à minha orientadora, por sua dedicação, orientação e seu mar de paciência nessa missão. Sem ela, não teria chegado a esses resultados com tanta confiança.

Também quero agradecer a minha família e amigos que me aguentaram nessa empreitada, me dando apoio, compreensão e torcida.

Agradecer a música, que sempre foi a calmaria após uma tempestade.

Quero mandar um obrigado para todos que de alguma forma contribuíram para esse projeto. Sem o apoio de cada um de vocês, essa conquista não teria sido possível.

Que esse TCC seja só o começo de muitas outras vitórias!

Success is my only option, failure's not. - Eminem

LISTA DE ABREVIATURAS

ARPAnet - Advanced Research Projects Network

API - Application Programming Interface

APPs - Aplicativos

CGI - Common Gateway Interface

CSS - Cascading Style Sheets

HTML - HyperText Markup Language

HTTP - Hypertext Transfer Protocol

MVC - Model-View-Controller

ODM - Object Document Mapping

ORM - Object Relational Mapping

PHP - PHP: Hypertext Preprocessor

SGBD - Sistema de gerenciamento de banco de dados

TCP/IP - Transmission Control Protocol/Internet Protocol

UC - Use Case

UML - Unified Modeling Language

URL - Uniform Resource Locator

XML - eXtensible Markup Language

LISTA DE TABELAS

Tabela 1: Especificação do caso de uso manter usuário.....	22
Tabela 2: Especificação do caso de uso manter publicação.....	23
Tabela 3: Especificação do caso de uso gerenciar imagem.....	24
Tabela 4: Especificação do caso de uso manter comentário.....	25
Tabela 5: Especificação do caso de uso manter rascunho.....	26

LISTA DE FIGURAS

Figura 1 - Reddit: criação de postagem.....	13
Figura 2 - Reddit: configurações.....	13
Figura 3 - AlternativeTo.....	14
Figura 4 - Alternativa Para.....	15
Figura 5 - Diagrama de Casos de Uso Geral do Sistema.....	23
Figura 6 - Diagrama de seqüência do fluxo principal do caso de uso manter usuário.....	28
Figura 7 - Diagrama de seqüência do fluxo alternativo I do caso de uso manter usuário.....	29
Figura 8 - Diagrama de seqüência do fluxo alternativo II do caso de uso manter usuário.....	29
Figura 9 - Diagrama de seqüência do fluxo principal do caso de uso manter publicação.....	30
Figura 10 - Diagrama de seqüência do fluxo alternativo I do caso de uso manter publicação.....	30
Figura 11 - Diagrama de seqüência do fluxo alternativo II do caso de uso manter publicação.....	31
Figura 12 - Diagrama de seqüência do fluxo alternativo III e IV do caso de uso manter publicação.....	31
Figura 13 - Diagrama de seqüência do fluxo principal do caso de uso gerenciar imagem.....	32
Figura 14 - Diagrama de seqüência do fluxo alternativo I do caso de uso gerenciar imagem.....	32
Figura 15 - Diagrama de seqüência do fluxo alternativo II do caso de uso gerenciar imagem.....	33
Figura 16 - Diagrama de seqüência do fluxo principal do caso de uso manter comentário.....	33
Figura 17 - Diagrama de seqüência do fluxo alternativo II do caso de uso manter comentário.....	34
Figura 18 - Diagrama de seqüência do fluxo principal do caso de uso manter rascunho.....	35
Figura 19 - Diagrama de seqüência do fluxo alternativo I do caso de uso manter rascunho.....	35
Figura 20 - Diagrama de seqüência do fluxo alternativo II do caso de uso manter rascunho.....	36
Figura 21 - Modelo entidade relacionamento.....	37
Figura 22 - Modelo lógico.....	38
Figura 23 - Diagrama de Classes.....	39
Figura 24 - Tela inicial do sistema.....	40
Figura 25 - Tela de login do sistema.....	41
Figura 26 - Tela de registro de usuário.....	41
Figura 27 - Tela de informações do usuário.....	42
Figura 28 - Tela das publicações.....	42
Figura 29 - Tela de Usuários.....	43
Figura 30 - Tela da publicação.....	43
Figura 31 - Como você avaliaria a facilidade de uso do sistema em uma escala de 1 a 5?.....	45
Figura 32 - Sobre as funcionalidades.....	45
Figura 33 - O design e layout do sistema são intuitivos para você? (Fáceis de usar).....	46
Figura 33 - Qual a probabilidade de você recomendar o sistema?.....	46

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 Objetivos.....	13
1.1.1 Objetivos Gerais.....	13
1.1.2 Objetivos Específicos.....	13
1.2 Trabalhos Correlatos.....	13
1.2.1 Reddit.....	13
1.2.2 AlternativeTo.....	15
1.2.3 Alternativa Para.....	15
1.2.4 Diferencial.....	16
1.3 Fundamentação Teórica.....	17
1.3.1 Modelagem MVC.....	17
1.3.2 Programação Orientada a Objetos.....	17
1.3.3 PHP.....	18
1.3.4 Laravel.....	18
1.3.5 Composer.....	19
1.3.6 XAMPP.....	20
1.3.6 Banco de dados relacional.....	20
1.3.7 MySQL Workbench.....	21
1.3.8 Draw.io.....	21
1.3.9 HTML e CSS.....	21
1.3.10 Bootstrap.....	22
2 DESENVOLVIMENTO.....	23
2.1 Caso de Uso.....	23
2.1.1 Diagrama de Casos de Uso.....	23
2.1.2 Especificações de Casos de Uso.....	24
2.2 Diagrama de Sequência.....	29
2.3 Banco de Dados.....	37
2.3.1 Diagrama Entidade Relacionamento.....	37
2.3.2 Modelo Lógico.....	38
2.4 Diagrama de Classes.....	40
2.5 Telas do Sistema.....	40
3 RESULTADOS E DISCUSSÕES.....	45
3.1 Questionário de avaliação e testes dos Sistemas.....	45
3.1.1 Resultados do Sistema.....	45
4 CONSIDERAÇÕES FINAIS.....	49
REFERÊNCIAS.....	50
APÊNDICE A : MODELO FÍSICO DO SISTEMA.....	52

1 INTRODUÇÃO

A partir de pesquisas militares para facilitar a troca de informações, surge durante o auge da Guerra Fria a ARPAnet, visando a comunicação dos computadores do sistema de defesa americano, que temiam um ataque iminente por parte da União Soviética. Essa rede de computadores evoluiu, e hoje se tornou a internet, que nada mais é que um número massivo de computadores conectados entre si, via protocolos TCP/IP, que trocam informações dentro da rede.

Nesse sentido, hoje em dia é difícil imaginar um mundo onde não exista a internet. Com a sua evolução constante, surgem novos aplicativos (APPs) para atender as necessidades do usuário para um aplicativo de *streaming*, como o *Spotify* ou para comunicações, como o *WhatsApp*.

Assim, um ambiente onde uma loja de aplicativos como a *Play Store* tem mais de 2 milhões de aplicativos, fica difícil saber qual aplicativo usar, e muitas vezes os usuários acabam sendo influenciados pelo que a maioria está usando, optando por algo que pode nem ter sido uma vontade genuína de quem o utiliza.

Pensando nisso, foi desenvolvido o Quebra Galho - Facilitando APPs, que auxiliará o usuário a ter uma noção maior em aplicativos que ele tem um interesse prévio, mostrando desde quais sistemas operacionais e dispositivos o aplicativo funciona, até o que muda entre as versões presentes para aquele app em diferentes plataformas.

Para o Quebra Galho, foi desenvolvida uma plataforma web no padrão de modelagem MVC tendo a linguagem de programação PHP para a *back-end*, HTML 5, CSS e *Bootstrap 5* para o design da *front-end* e o SGBD *MySQL* para gerenciamento do banco de dados.

Dessa maneira, o sistema funcionará de forma colaborativa entre os usuários por meio de uma espécie de fórum, onde os usuários poderão trocar experiências sobre as aplicações e debater sobre as funcionalidades. O site também conta com um módulo onde usuários com maiores privilégios poderão atualizar as informações dos aplicativos, passando por uma espécie de aprovação dos outros usuários, onde as informações serão verificadas antes de serem postadas.

1.1 Objetivos

1.1.2 Objetivos Gerais

Desenvolver uma plataforma web autêntica, confiável, intuitiva e acessível para os diferentes níveis de usuários, onde os mesmos possam gerenciar informações sobre os aplicativos de maneira colaborativa.

1.1.3 Objetivos Específicos

- Permitir o registro, *login* e exclusão de usuários no sistema;
- Ter níveis de usuário: padrão, cadastrado e administrador;
- Possibilitar o cadastro, alteração, exclusão e consulta de publicações sobre os aplicativos, de acordo com o nível do usuário;
- Possibilitar a adição de fotos juntamente às publicações;
- Permitir salvar rascunhos ou arquivar as publicações;
- Trabalhar com interações via e-mail;
- Ter uma seção de comentários para interação dos usuários.

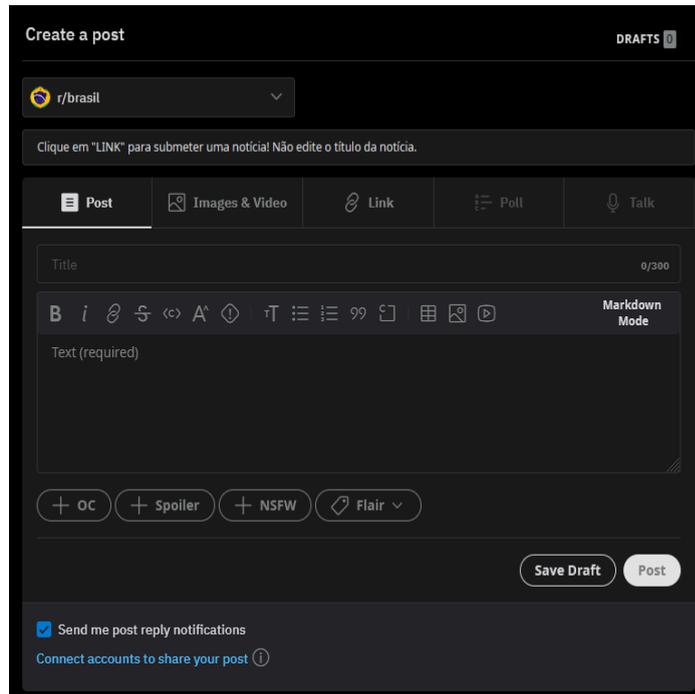
1.2 Trabalhos Correlatos

Nesta seção serão apresentados os trabalhos correlatos, que serviram de inspiração para partes do trabalho desenvolvido.

1.2.1 Reddit

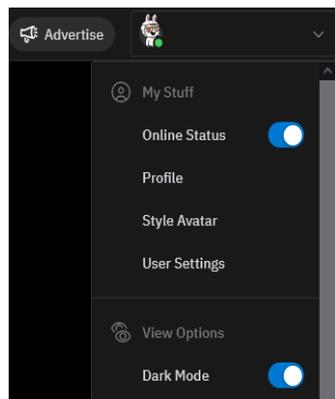
O Reddit é uma popular rede social formada por fóruns, onde usuários podem criar e debater discussões sobre os assuntos mais variados. A figura 1 apresenta a seção de criação de postagens do Reddit, enquanto a figura 2, uma parte da seção de configurações do usuário.

Figura 1 - Reddit: criação de postagem



Fonte: [Reddit](#)

Figura 2 - Reddit: configurações



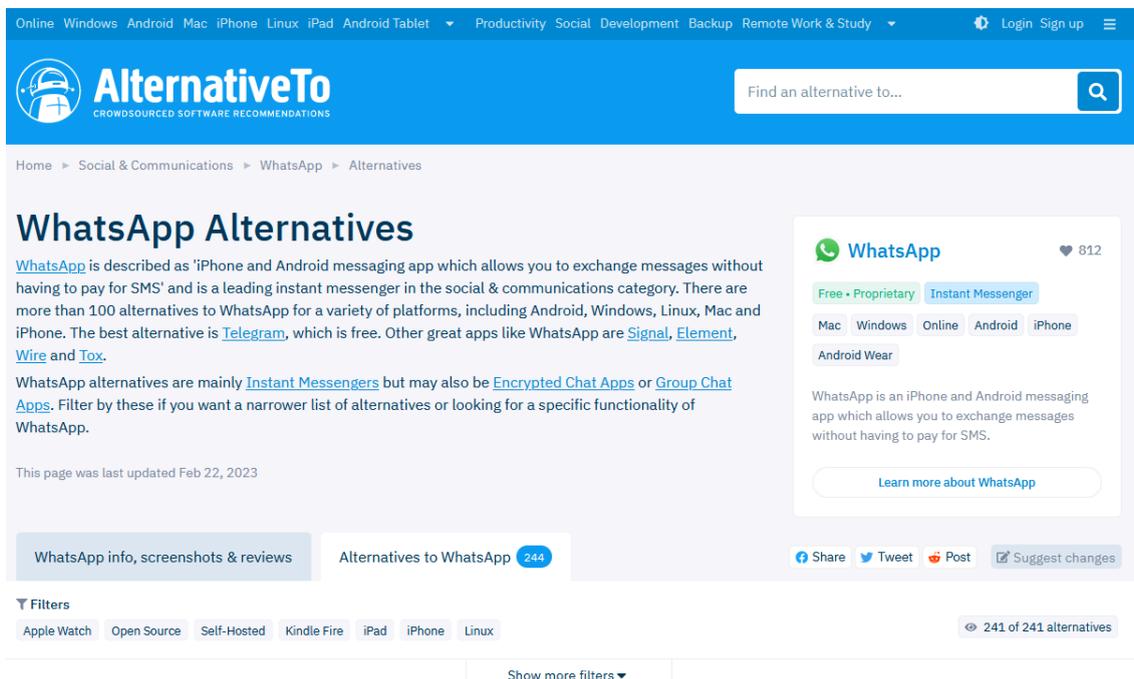
Fonte: [Reddit](#)

O site conta com diversas opções de estilizar suas publicações, formatando o texto, podendo adicionar fotos e vídeos juntamente ao texto. Na seção do usuário podemos observar a praticidade em acessar os links vinculados ao perfil do usuário (em destaque na Figura 2).

1.2.2 AlternativeTo

Neste site o usuário busca por um aplicativo que ele não deseja utilizar, e o buscador retornará a página deste aplicativo, onde há aplicativos parecidos com este, informando descrição e *tags* sobre as plataformas onde este APP pode ser utilizado, sobre a licença de uso do programa e uma seção de revisões dos usuários para os aplicativos. A figura 3 mostra a página de um aplicativo de mensagens no AlternativeTo.

Figura 3 - AlternativeTo



The screenshot shows the AlternativeTo website interface. At the top, there is a navigation bar with categories like 'Online', 'Windows', 'Android', etc., and a search bar. The main content area is titled 'WhatsApp Alternatives' and includes a detailed description of WhatsApp and its alternatives. The page also features a list of alternative platforms and filters for 'Free', 'Proprietary', and 'Instant Messenger'. The page is updated as of Feb 22, 2023.

Fonte: [AlternativeTo](https://www.alternativeto.com)

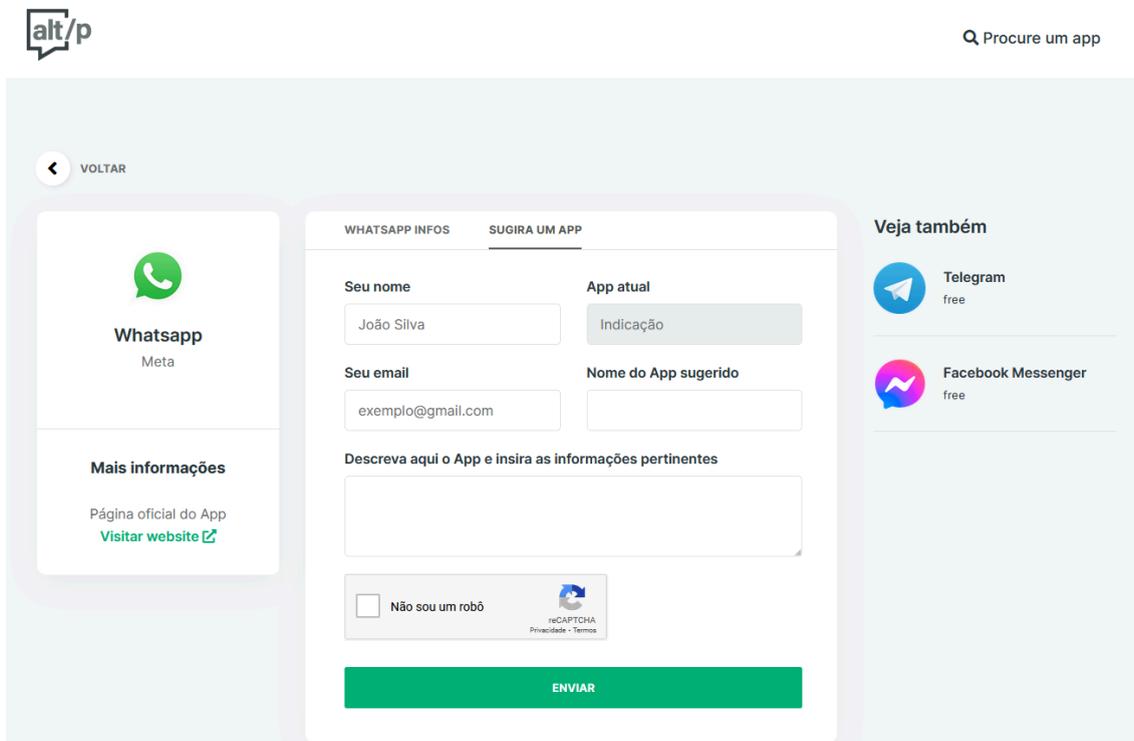
Este site é muito utilizado por usuários que procuram alternativas para aplicativos.

1.2.3 Alternativa Para

O Alternativa Para tem um objetivo parecido com o citado anteriormente AlternativeTo, com seu diferencial de ser totalmente em português, não contando com uma

seção de comentários, porém há uma seção onde o usuário pode sugerir um aplicativo. A figura 4 mostra a página de sugestões de aplicativos do site Alternativa Para.

Figura 4 - Alternativa Para



The image shows a web form for suggesting an app. At the top left is the 'alt/p' logo and a search bar with the text 'Procure um app'. Below the logo is a 'VOLTAR' button. The form is titled 'SUGIRA UM APP' and is divided into two main sections: 'WHATSAPP INFOS' and 'SUGIRA UM APP'. The 'WHATSAPP INFOS' section features the WhatsApp logo, the name 'Whatsapp', the developer 'Meta', and a link to the 'Página oficial do App' with the text 'Mais informações' and 'Visitar website'. The 'SUGIRA UM APP' section contains several input fields: 'Seu nome' (filled with 'João Silva'), 'App atual' (filled with 'Indicação'), 'Seu email' (filled with 'exemplo@gmail.com'), and 'Nome do App sugerido'. Below these is a large text area for 'Descreva aqui o App e insira as informações pertinentes'. At the bottom of the form is a checkbox for 'Não sou um robô' next to a reCAPTCHA logo and a green 'ENVIAR' button. To the right of the form is a 'Veja também' section with recommendations for 'Telegram free' and 'Facebook Messenger free'.

Fonte: [Alternativa Para](#)

1.2.4 Diferencial

O Quebra Galho - Facilitando APPs foi desenvolvido com inspiração em algumas funcionalidades nestes trabalhos correlatos que foram citados anteriormente, tendo também o seu diferencial no sistema.

Esse diferencial se refere pelo fato do sistema ser focado em descrever o aplicativo, as diferentes funcionalidades dependendo da plataforma, e não somente nas alternativas para ele. Também as postagens, que serão uma forma de sistema colaborativo entre os usuários,

passarão pelo aval de um administrador, para averiguar a veracidade das informações e essas publicações contarão com uma seção de comentários, para discussão de quem a vê.

1.3 Fundamentação Teórica

Neste capítulo serão tratadas as fundamentações teóricas das ferramentas utilizadas para o desenvolvimento do projeto.

1.3.1 Modelagem MVC

A arquitetura Model-View-Controller (MVC) é uma arquitetura de software utilizada para separar as responsabilidades em um aplicativo. A MVC é um dos padrões de projeto mais populares e é comumente usado em aplicativos da web.

Martin Fowler, um estudioso importante na área de desenvolvimento de software, define MVC como:

MVC é uma arquitetura de software que separa os elementos de apresentação de uma aplicação em três componentes principais: o modelo, que contém os dados e a lógica de negócios; a visão, que representa a interface do usuário; e o controlador, que coordena a interação entre o modelo e a visão (FOWLER, 2003, p. 252).

A MVC é benéfica porque separa as responsabilidades em um aplicativo e torna mais fácil gerenciar e manter o código. Ele também permite que diferentes equipes trabalhem em diversos aspectos do aplicativo sem afetar o trabalho dos outros.

Além disso, a MVC pode facilitar a reutilização de código. Uma vez que o modelo, a visão e o controlador são separados, eles podem ser usados em diferentes aplicativos sem a necessidade de reescrever o código.

1.3.2 Programação Orientada a Objetos

A Programação Orientada a Objetos (POO) é um paradigma de programação que se baseia na criação de objetos, que são instâncias de classes, como forma de organizar e estruturar o código.

Segundo Cay Horstmann, autor do livro "Big Java: Late Objects" (2013), a Programação Orientada a Objetos (POO) é um paradigma de programação que se concentra em modelar o mundo real como uma coleção de objetos. Cada objeto tem um estado interno, um conjunto de operações que pode executar uma maneira de se comunicar com outros objetos.

Os objetos possuem atributos, que representam suas características, e métodos, que representam seus comportamentos. Além disso, os objetos podem se comunicar entre si por meio de mensagens, que são chamadas de métodos.

A POO apresenta diversas vantagens em relação a outros paradigmas de programação, como a modularidade, o encapsulamento, a herança e o polimorfismo. A modularidade se refere à capacidade de dividir o código em módulos independentes, facilitando a manutenção e o reuso do código. O encapsulamento, permite ocultar informações importantes e restringir o acesso a elas, aumentando a segurança do código. A herança permite a criação de novas classes a partir de outras já existentes, aproveitando suas características e comportamentos. Já o polimorfismo permite que objetos de diferentes classes sejam tratados de forma semelhante, aumentando a flexibilidade do código.

1.3.3 PHP

PHP é uma linguagem de programação de código aberto, desenvolvida em 1994 como um conjunto de binários escritos em CGI em linguagem C. Combinando código com HTML, o PHP permite criar páginas web dinâmicas que se adaptam às interações do usuário e se conectam a bancos de dados.

Atualmente na versão 8.2, o PHP vem ganhando cada vez mais adeptos devido à sua facilidade de compreensão, gerando uma curva de aprendizado linear e também as suas constantes atualizações, que acabam levando-o a ter uma evolução contínua, onde a cada nova versão, são adicionando recursos, melhorando o desempenho e abordando questões de segurança. Que vem para manter a linguagem relevante e atraente para desenvolvedores e empresas (PHP, 2023).

1.3.4 Laravel

O Laravel é um *framework* de desenvolvimento web PHP que foi lançado pela primeira vez em 2011 por Taylor Otwell. Desde então, ele se tornou um dos *frameworks* mais populares e amplamente adotados na comunidade de desenvolvedores PHP. Sendo projetado com foco na elegância, simplicidade e produtividade, oferecendo um conjunto abrangente de ferramentas e recursos para facilitar o desenvolvimento de aplicativos web modernos e escaláveis. Com lançamentos regulares de novas versões com recursos inovadores e uma

comunidade ativa que contribui para seu crescimento contínuo, o Laravel conquistou uma base de usuários fiel e continua a ser uma escolha popular para desenvolvedores que buscam criar aplicativos web de alta qualidade de maneira eficiente.

Segundo Taylor Otwell: “Laravel é o competidor mais forte no ecossistema PHP simplesmente porque ele inclui as funcionalidades necessárias para construir aplicativos web modernos”.

Dentre as principais funcionalidades do Laravel, destaca-se o Eloquent ORM que facilita a interação entre o código da aplicação e o banco de dados, seu sistema de rotas para gerenciamento das URLs das páginas, sua tecnologia de *seeders* e *factories* que são úteis para criar registros no banco de dados, o mecanismo Blade de templates que auxilia para criar interfaces de usuários simplificadas e o *Artisan Console* que é uma ferramenta de linha de comandos.

1.3.5 Composer

O Composer é uma ferramenta de gerenciamento de dependências para PHP que foi lançada pela primeira vez em 2011. Ele foi desenvolvido por Nils Adermann e Jordi Boggiano e rapidamente se tornou uma parte fundamental do ecossistema de desenvolvimento PHP. O Composer simplifica o processo de instalação e atualização de bibliotecas de terceiros, tornando o desenvolvimento PHP mais eficiente e organizado.

O Composer desempenha um papel vital no desenvolvimento com PHP e no *framework* Laravel. Ele é uma ferramenta de gerenciamento de dependências que simplifica a inclusão de bibliotecas e pacotes em projetos.

No contexto do Laravel, o Composer é usado para instalar e atualizar os componentes essenciais do framework e suas dependências, tornando o processo de desenvolvimento mais eficiente e mantendo o ecossistema Laravel atualizado. Ele também ajuda a gerenciar as dependências de seus próprios projetos Laravel, facilitando a integração de bibliotecas e pacotes de terceiros, garantindo um ambiente de desenvolvimento coeso e confiável. Em resumo, o Composer com PHP e Laravel simplifica a gestão de dependências, economizando tempo e facilitando o desenvolvimento de aplicativos web robustos e escaláveis (COMPOSER, 2023).

1.3.6 XAMPP

O XAMPP é um pacote de software livre que inclui um servidor web Apache, banco de dados MySQL e interpretadores para linguagens de programação como PHP e Perl. O nome XAMPP é um acrônimo para os sistemas operacionais suportados: Windows, Linux, macOS e Solaris.

Dalibor Dvorski define o XAMPP como uma distribuição Apache muito comum no desenvolvimento web, ideal para estudantes que desejam desenvolver e testar aplicações em PHP e MySQL.

Além do Apache, MySQL e PHP, o XAMPP também inclui outras ferramentas úteis para desenvolvedores, como phpMyAdmin (para gerenciamento de banco de dados), FileZilla (cliente FTP), Mercury Mail e Tomcat (APACHEFRIENDS, 2023).

1.3.6 Banco de dados relacional

É um tipo de banco de dados que organiza os dados em tabelas relacionais. Essas tabelas são compostas por colunas (campos) e linhas (registros) e são interconectadas através de chaves primárias e estrangeiras.

Uma chave primária é um campo ou conjunto de campos que identifica exclusivamente cada registro em uma tabela. Uma chave estrangeira é um campo em uma tabela que se refere a uma chave primária em outra tabela, permitindo que as tabelas sejam relacionadas.

O uso de um banco de dados relacional permite que os dados sejam armazenados de forma estruturada, facilitando a busca, a organização e a manipulação dos dados. Isso torna o banco de dados relacional uma ferramenta essencial em muitas aplicações de software, como sistemas de gerenciamento de estoque, sistemas bancários, sistemas de gerenciamento de clientes, entre outros.

Chris Date enfatiza em seu livro “An Introduction to Database Systems” (2003), a importância de se seguir o modelo relacional e suas regras, incluindo a integridade referencial e a normalização, para garantir que os dados sejam armazenados e recuperados de forma precisa e eficiente. Ele argumenta que o modelo relacional é a melhor maneira de representar

dados em um sistema de gerenciamento de banco de dados, pois oferece uma estrutura clara e consistente para armazenar, recuperar e manipular dados.

1.3.7 MySQL Workbench

O MySQL Workbench é uma ferramenta gráfica de administração de banco de dados MySQL pertencente à Oracle *Corporation*, que permite aos usuários criar, visualizar e modificar bancos de dados. Essa ferramenta fornece uma interface de usuário intuitiva para gerenciar bancos de dados MySQL, permitindo que os usuários criem e modifiquem tabelas, execute consultas SQL, gerencie usuários e permissões e criar e ilustrar os relacionamentos entre as tabelas, permitindo assim a criação do modelo lógico do banco de dados (MYSQL, 2023).

1.3.8 Draw.io

O Draw.io é uma ferramenta online de diagramação, de código aberto que permite criar vários tipos de diagramas, aqui utilizado para os casos de uso e o modelo entidade-relacionamento.

Nesta ferramenta é possível criar diagramas usando uma variedade de formas, ícones e imagens, permitindo uma personalização no estilo visual dos diagramas. A ferramenta também permite importar e exportar diagramas em vários formatos, como PNG, JPEG, PDF e XML (DIAGRAMS.NET, 2023).

1.3.9 HTML e CSS

HTML é uma linguagem de marcação utilizada para criar e estruturar conteúdo na web. A linguagem HTML utiliza tags, que são códigos especiais inseridos no texto, para definir a estrutura e o formato do conteúdo (HTML, 2023).

Já o CSS, é uma linguagem de estilização utilizada para definir a apresentação visual de uma página web criada com HTML. Em vez de definir o estilo de cada elemento individualmente em HTML, o CSS permite criar um conjunto de regras para ser aplicado a um ou mais elementos. Isso significa que é possível atualizar a aparência de um site inteiro alterando apenas uma folha de estilo CSS (CSS, 2023).

1.3.10 Bootstrap

Bootstrap é um *framework front-end* de código aberto desenvolvido pela equipe do GitHub. Ele é usado para criar interfaces responsivas, amigáveis ao usuário e modernas para aplicativos web e móveis.

A importância da interface responsiva se dá pelo fato de que hoje em dia as pessoas acessam a internet através de uma grande variedade de dispositivos. Para fornecer uma boa experiência de usuário em todos esses dispositivos, é essencial ter uma interface que seja capaz de se adaptar a cada um deles.

O Bootstrap é baseado em HTML, CSS e JavaScript e inclui um conjunto de componentes de interface do usuário pré-estilizados. Componentes que podem ser facilmente integrados em um projeto existente e personalizados para atender às necessidades específicas do projeto (BOOTSTRAP, 2023).

2 DESENVOLVIMENTO

2.1 Caso de Uso

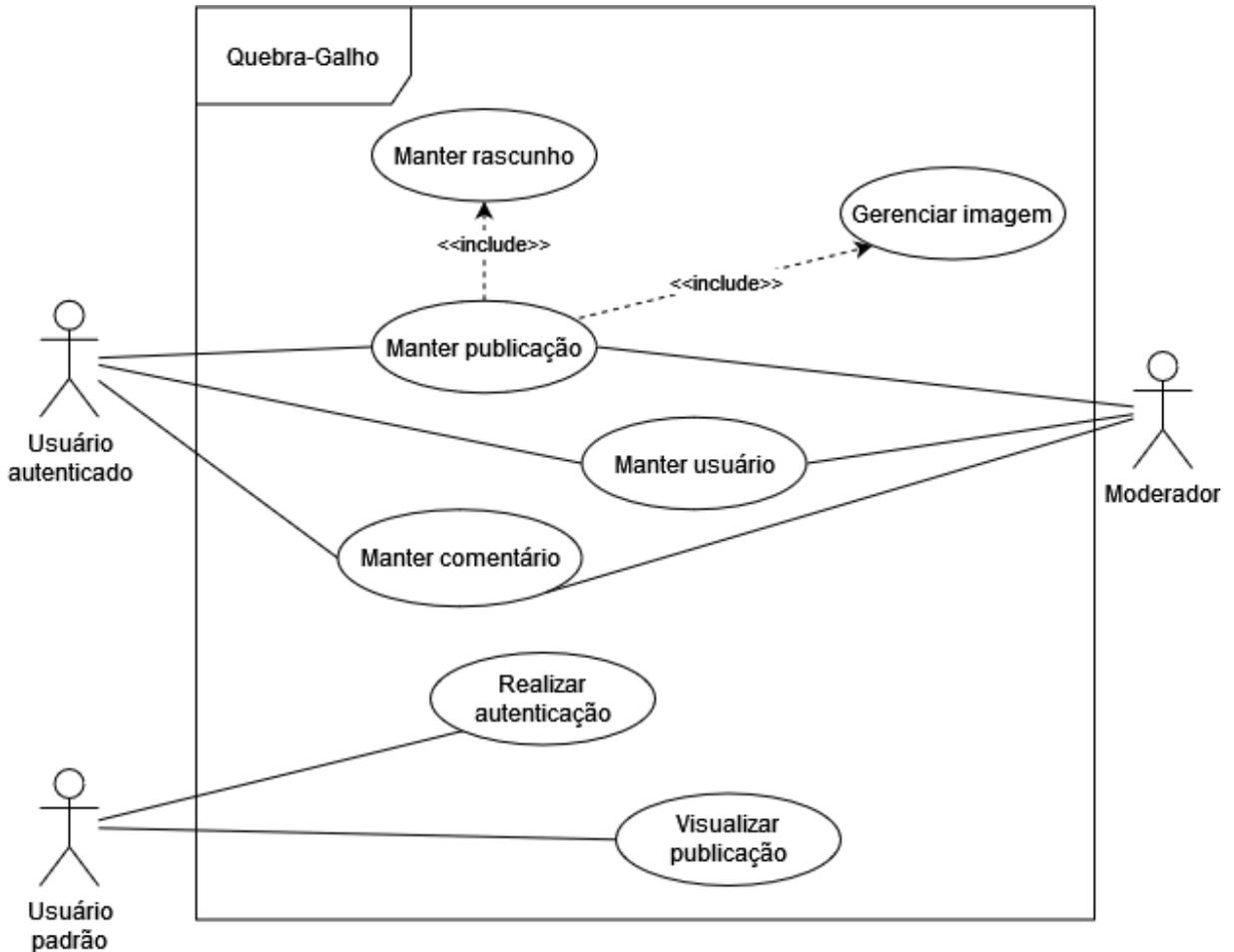
Eduardo Bezerra destaca em seu livro "Desenvolvimento Orientado a Objetos com UML - Uma Introdução Prática com Java" que casos de uso são uma técnica utilizada na análise e modelagem de sistemas orientados a objetos, e seu objetivo é capturar as interações entre os usuários e o sistema em questão. Eles descrevem um cenário em que um usuário utiliza o sistema para realizar uma tarefa específica e são usados para identificar e descrever os requisitos funcionais do sistema.

De modo geral, os casos de uso servem para auxiliar os usuários ou quem vê a modelagem, sobre quais são as funcionalidades do sistema em forma de cenários, para fins de entendimento do sistema como um todo.

2.1.1 Diagrama de Casos de Uso

Para o caso de uso do projeto, foram utilizados três atores, o usuário padrão, que é o usuário que não está logado ao sistema, que pode somente visualizar publicações e realizar autenticação. Já o usuário autenticado, que é o usuário que se encontra conectado ao sistema, já pode criar, editar, apagar sua própria publicação e salvá-la como rascunho para poder postar em outro momento, visualizar seu cadastro de usuário e editá-lo e também adicionar comentário em sua ou outras publicações. O moderador tem acesso total ao sistema, podendo ver todos os usuários, editar seus dados, apagá-los e também poderá apagar publicações e comentários mesmo não tendo sido ele que publicou essas informações.

Figura 5 - Diagrama de Casos de Uso Geral do Sistema



Fonte: Própria (2023).

2.1.2 Especificações de Casos de Uso

A seguir, serão apresentadas as especificações dos casos de uso referentes ao diagrama exposto na figura 5.

Tabela 1: Especificação do caso de uso manter usuário

Identificação UC: MU 01	Nome UC: Manter usuário
Atores: Usuário autenticado, moderador.	
Resumo: Esse caso de uso descreve os passos para manter um determinado usuário.	
Pré-condições: Estar logado no sistema.	
Fluxo principal	

Ações do Ator	Ações do Sistema
1. Clicar no ícone de “Usuário”.	2. Retorna tela com as informações do usuário atual logado no sistema.
Fluxo alternativo I - Alterar informações de um usuário	
3. Localizar o usuário que deseja atualizar as informações, e altera as informações desejadas.	4. Atualiza informações no SGBD e retorna o usuário para a tela de visualização do usuário, para ver os dados que alterou.
Fluxo alternativo II - Criar usuário	
1. Clicar em “Novo por aqui?” na tela de autenticação; 2. Inserir as informações solicitadas pelo sistema.	3. Grava informações no SGBD; 4. Gera uma mensagem ao usuário para informar que a operação foi bem-sucedida. 5. Retorna usuário para tela de login.

Fonte: Própria (2023).

Tabela 2: Especificação do caso de uso manter publicação

Identificação UC: MP 01	Nome UC: Manter publicação
Ator: Usuário autenticado, moderador.	
Resumo: Esse caso de uso descreve os passos para manter uma determinada publicação.	
Pré-condições: Estar logado no sistema.	
Fluxo principal	
Ações do Ator	Ações do Sistema
1. Clicar na publicação na lista de publicações.	2. Retorna <i>view</i> da publicação.
Fluxo alternativo I - Criar publicação	
1. Clicar no ícone “+” na lista de publicações; 3. Insere informações da publicação e a grava.	2. Retorna tela para inserção de publicação; 4. Grava informações no SGBD e gera uma mensagem ao usuário para informar que a operação foi bem-sucedida; 5. Retorna o usuário para a <i>view</i> da

	publicação.
Fluxo alternativo II - Alterar publicação	
3. Clicar em “Alterar publicação”; 4. Alterar as informações desejadas.	5. Grava informações no SGBD e gera uma mensagem ao usuário para informar que a operação foi bem-sucedida; 6. Retorna usuário para a <i>view</i> da publicação.
Fluxo alternativo III - Desativar publicação	
3. Clicar em “Alterar publicação”; 4. Clicar no ícone de apagar a publicação; 6. Confirma a ação.	5. Retorna tela <i>pop-up</i> para confirmação da ação; 7. Desativa a publicação no SGBD e gera uma mensagem ao usuário para informar que a operação foi bem-sucedida; 8. Retorna usuário para a lista de publicações.
Fluxo alternativo IV - Reativar publicação	
3. Clicar em “Alterar publicação”; 4. Clicar no ícone de reexibir a publicação; 6. Confirma a ação.	5. Retorna tela <i>pop-up</i> para confirmação da ação; 7. Desativa a publicação no SGBD e gera uma mensagem ao usuário para informar que a operação foi bem-sucedida; 8. Retorna usuário para a <i>view</i> da publicação recém reativada.

Fonte: Própria (2023).

Tabela 3: Especificação do caso de uso gerenciar imagem

Identificação UC: GI 01	Nome UC: Gerenciar imagem
Ator: Usuário autenticado, moderador.	
Resumo: Esse caso de uso descreve os passos para manter uma imagem ou mais imagens na publicação.	
Pré-condições: Publicação postada.	
Fluxo principal	
Ações do Ator	Ações do Sistema

1. Ir na seção de alteração da publicação; 2. Clicar na imagem.	3. Retorna <i>view</i> da imagem em uma dimensão maior, para melhor visualização.
Fluxo alternativo I - Adicionar foto na publicação	
2. Clicar no ícone para adicionar imagem; 4. Seleciona uma ou mais imagens e as adiciona na publicação.	3. Abre API para usuário buscar imagem online; 5. Atualiza informações no SGBD e gera uma mensagem ao usuário para informar que a operação foi bem-sucedida; 6. Retorna usuário para <i>view</i> da publicação.
Fluxo alternativo II - Excluir imagem	
2. Clicar no ícone de exclusão de imagens; 4. Confirma a exclusão.	3. Retorna tela para confirmação da exclusão; 5. Exclui a imagem e grava as informações no SGBD; 6. Gera uma mensagem ao usuário para informar que a operação foi bem-sucedida. 7. Retorna usuário para a <i>view</i> da publicação.

Fonte: Própria (2023).

Tabela 4: Especificação do caso de uso manter comentário

Identificação UC: MC 01	Nome UC: Manter comentário
Ator: Usuário autenticado, moderador.	
Resumo: Esse caso de uso descreve os passos para manter um comentário em uma publicação.	
Pré-condições: Publicação postada no sistema.	
Fluxo principal	
Ações do Ator	Ações do Sistema
1. Clicar na publicação na lista de publicações; 2. Na seção de comentários, insere comentário na caixa de texto e grava.	3. Retorna <i>view</i> da publicação, na seção de comentários, com comentário inserido.

Fluxo alternativo I - Excluir comentário	
2. Clicar no botão para excluir comentário; 4. Confirma exclusão.	5. Exclui comentário e retorna usuário para seção de comentários da publicação.

Fonte: Própria (2023).

Tabela 5: Especificação do caso de uso manter rascunho

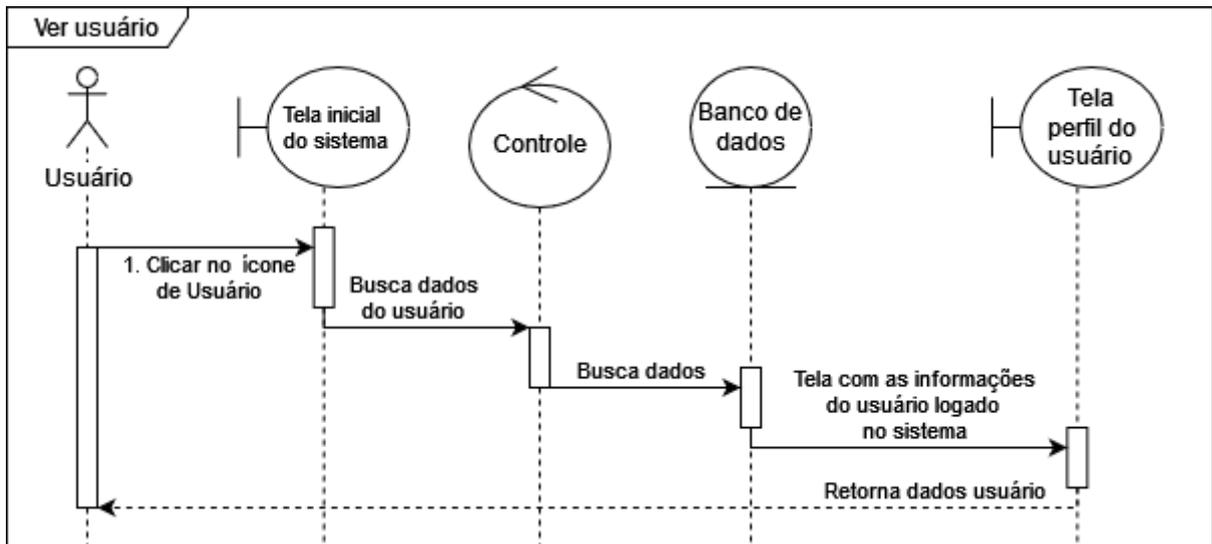
Identificação UC: MR 01	Nome UC: Manter rascunho
Ator: Usuário autenticado, moderador.	
Resumo: Esse caso de uso descreve os passos para manter um rascunho de uma publicação.	
Pré-condições: Publicação postada no sistema.	
Fluxo principal	
Ações do Ator	Ações do Sistema
1. Clicar no ícone “+” na lista de publicações; 2. Insere informações da publicação e clica para salvar.	3. Grava informações no SGBD e gera uma mensagem ao usuário para informar que a publicação foi salva para um envio futuro; 4. Retorna o usuário para a <i>view</i> do rascunho da publicação.
Fluxo alternativo I - Publicar rascunho	
1. Na lista de publicações, clicar no rascunho que deseja publicar, uma vez que eles estarão no topo da lista, na seção de rascunhos; 2. Alterar as informações desejadas e clicar para gravar.	3. Grava informações no SGBD, informa que a operação foi bem-sucedida; 4. Retorna <i>view</i> da publicação recém postada.
Fluxo alternativo II - Excluir rascunho	
1. Na lista de publicações, clicar para exclusão no rascunho que deseja excluir, uma vez que eles estarão no topo da lista, na seção de rascunhos; 2. Confirmar.	5. Retorna tela <i>pop-up</i> para user, perguntando se deseja realmente excluir o rascunho; 6. Grava informações no SGBD, informa que a operação foi bem-sucedida; 7. Redireciona usuário para lista de publicações.

Fonte: Própria (2023).

2.2 Diagrama de Sequência

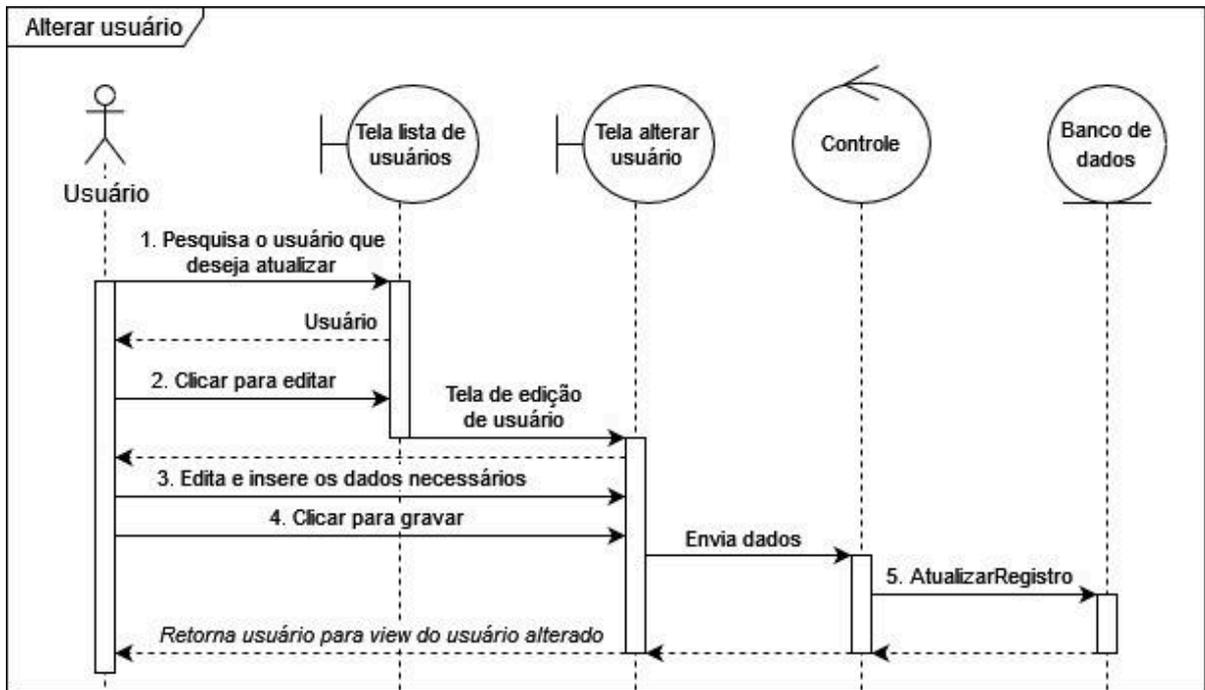
Nesta seção serão apresentados os Diagramas de Sequência, que facilitam o entendimento dos casos de uso citados anteriormente, as Figura 6, 7 e 8 ilustram o diagrama referente ao fluxo principal e aos fluxos alternativos I e II, do Caso de Uso Manter Usuário.

Figura 6 - Diagrama de sequência do fluxo principal do caso de uso manter usuário



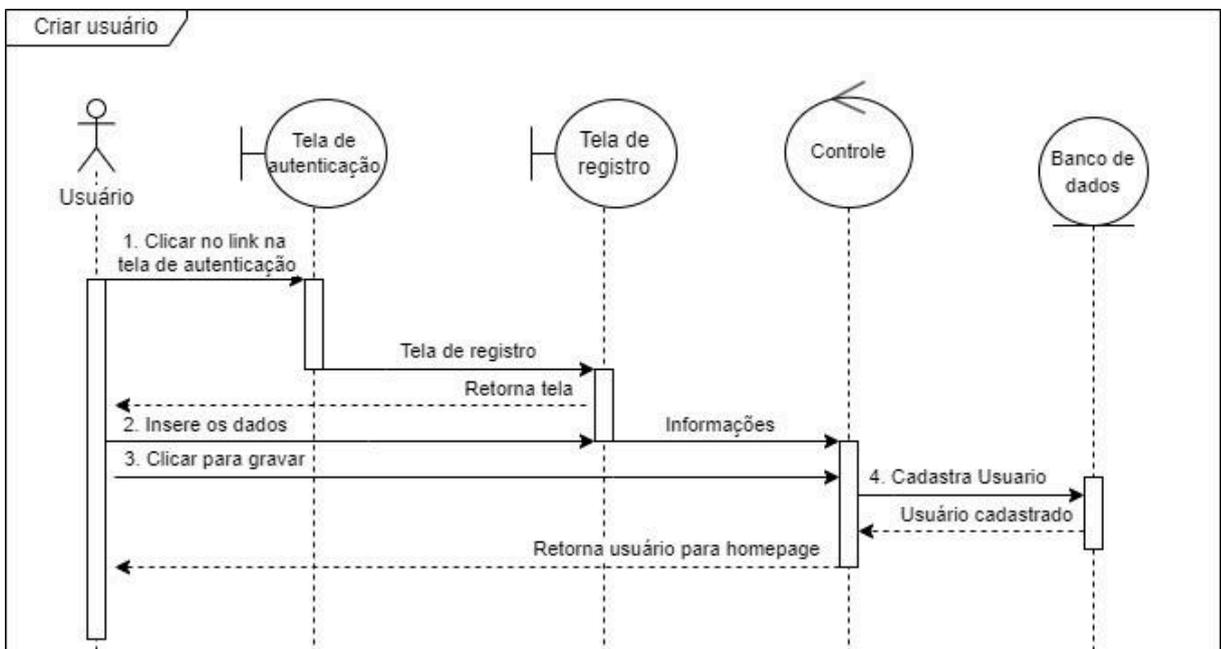
Fonte: Própria (2023)

Figura 7 - Diagrama de sequência do fluxo alternativo I do caso de uso manter usuário



Fonte: Própria (2023)

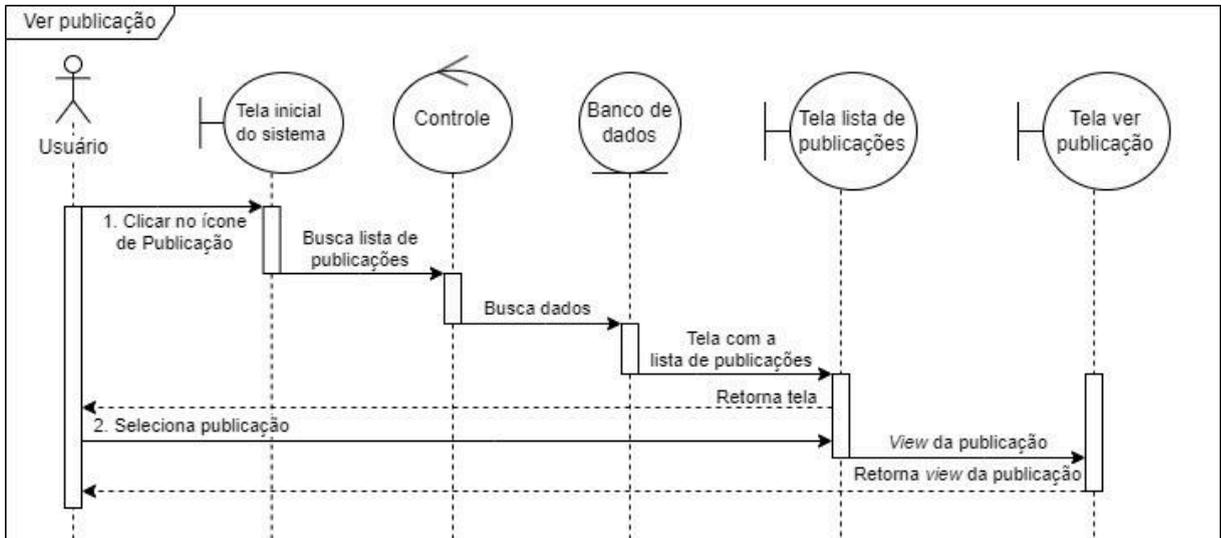
Figura 8 - Diagrama de sequência do fluxo alternativo II do caso de uso manter usuário



Fonte: Própria (2023)

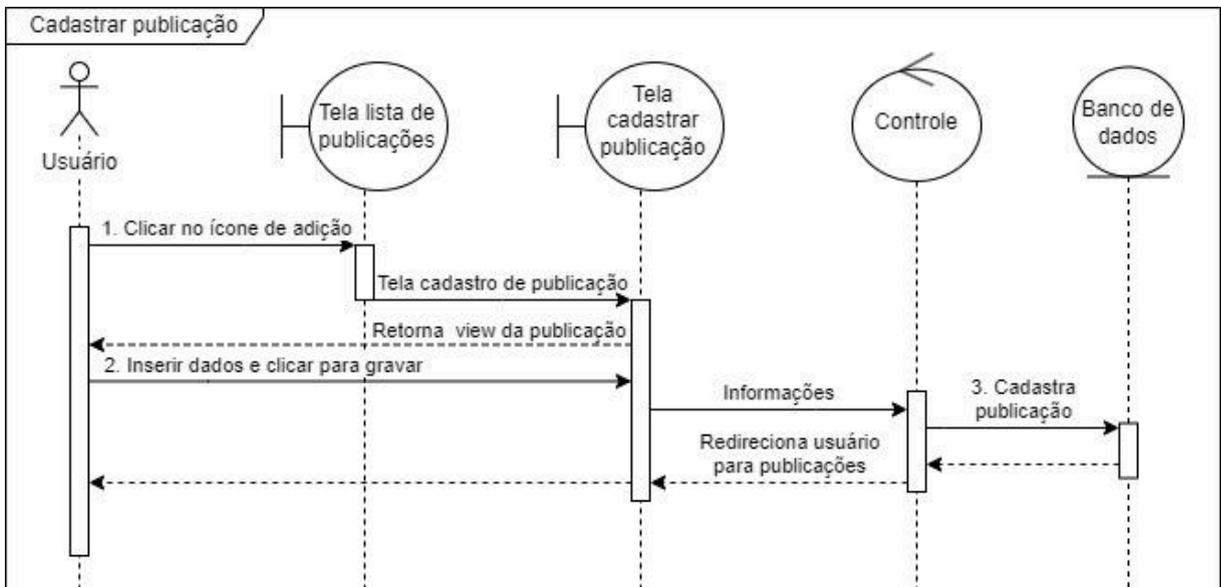
Já as Figuras 9, 10, 11 e 12 apresentam o fluxo principal e os fluxos alternativos do Caso de Uso Manter Publicação.

Figura 9 - Diagrama de sequência do fluxo principal do caso de uso manter publicação



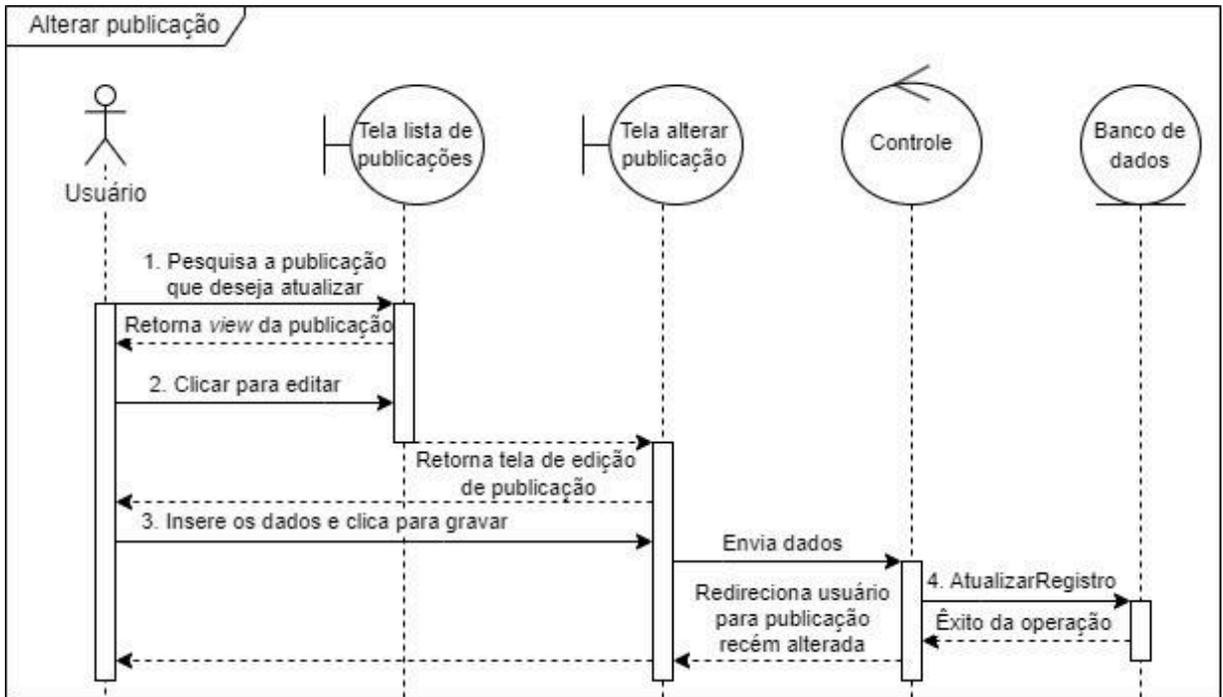
Fonte: Própria (2023)

Figura 10 - Diagrama de sequência do fluxo alternativo I do caso de uso manter publicação



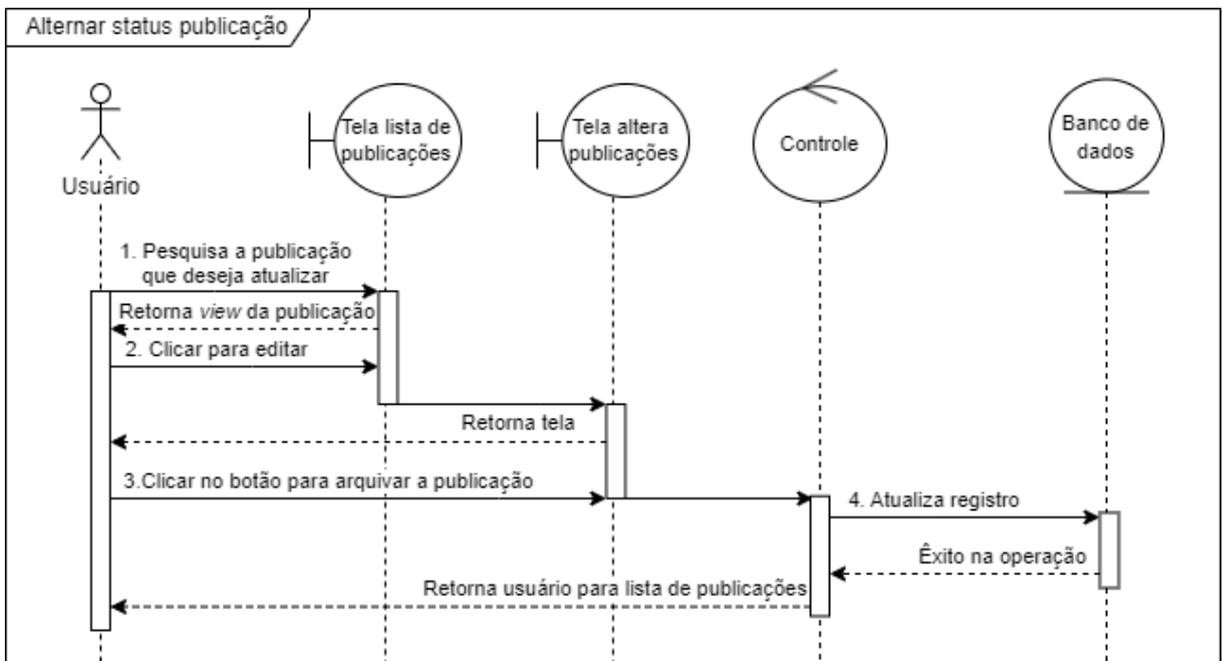
Fonte: Própria (2023)

Figura 11 - Diagrama de sequência do fluxo alternativo II do caso de uso manter publicação



Fonte: Própria (2023)

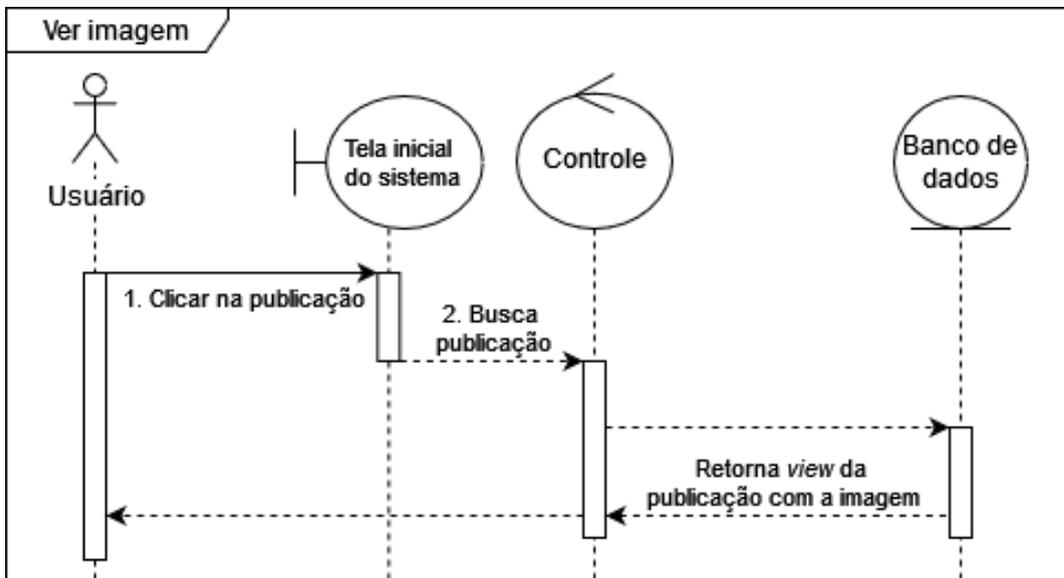
Figura 12 - Diagrama de sequência do fluxo alternativo III e IV do caso de uso manter publicação



Fonte: Própria (2023)

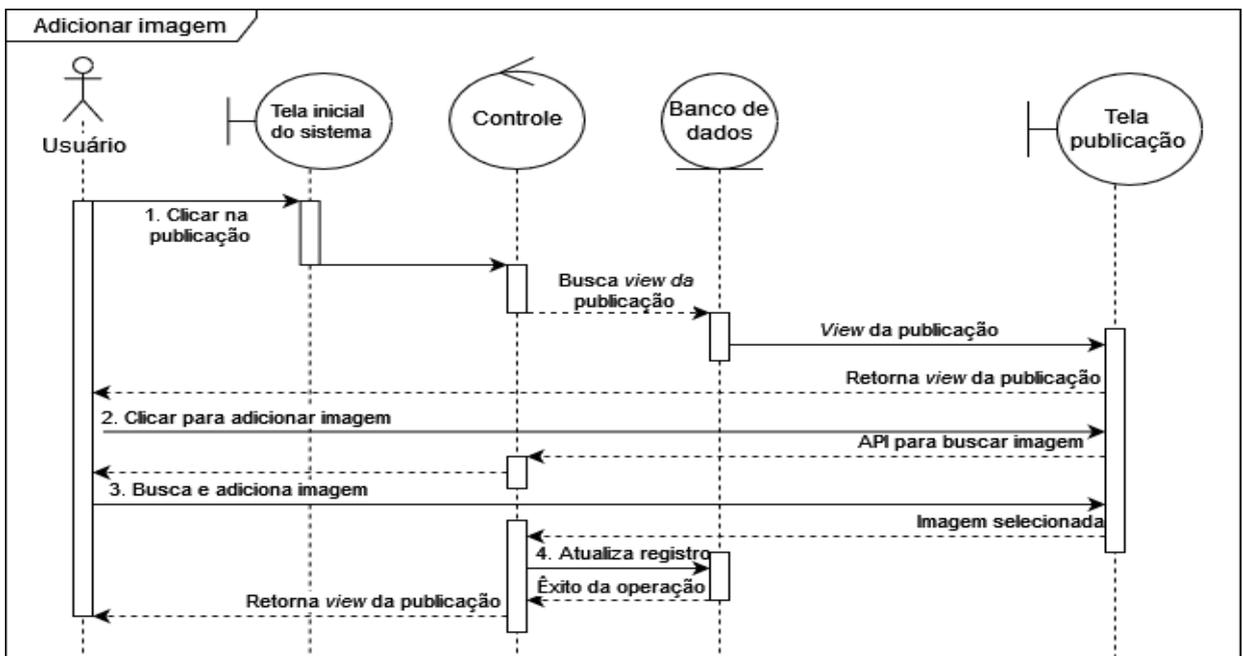
Os diagramas que representam a sequência dos fluxos referentes ao Caso de Uso Gerenciar Imagem, estão presentes nas Figuras 13, 14 e 15.

Figura 13 - Diagrama de sequência do fluxo principal do caso de uso gerenciar imagem



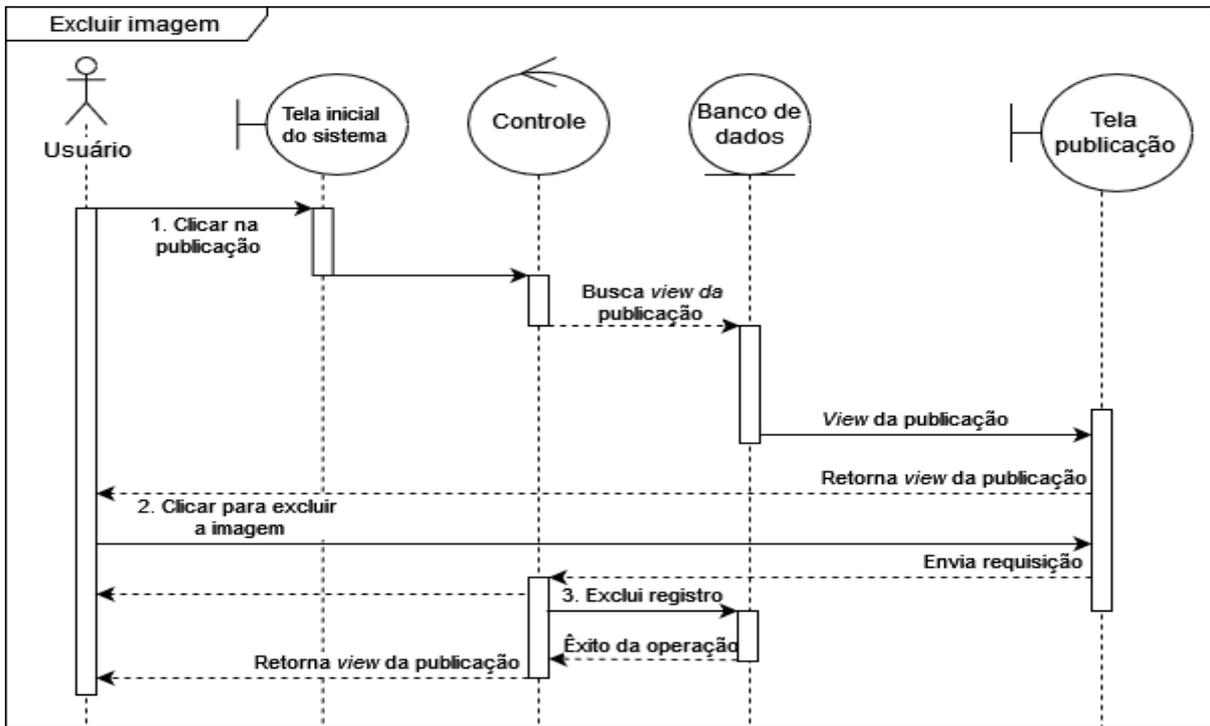
Fonte: Própria (2023)

Figura 14 - Diagrama de sequência do fluxo alternativo I do caso de uso gerenciar imagem



Fonte: Própria (2023)

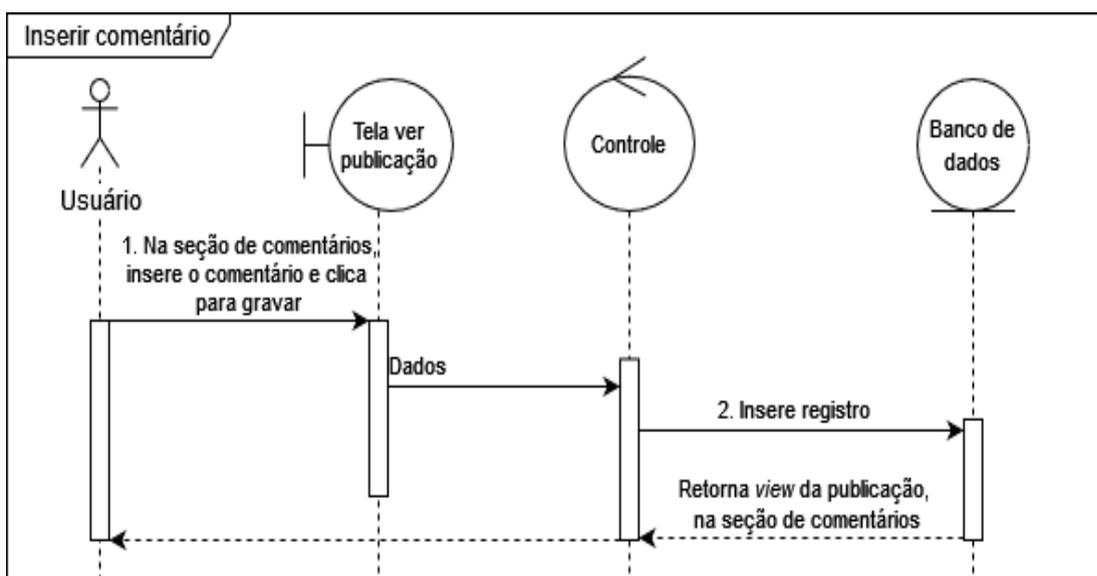
Figura 15 - Diagrama de sequência do fluxo alternativo II do caso de uso gerenciar imagem



Fonte: Própria (2023)

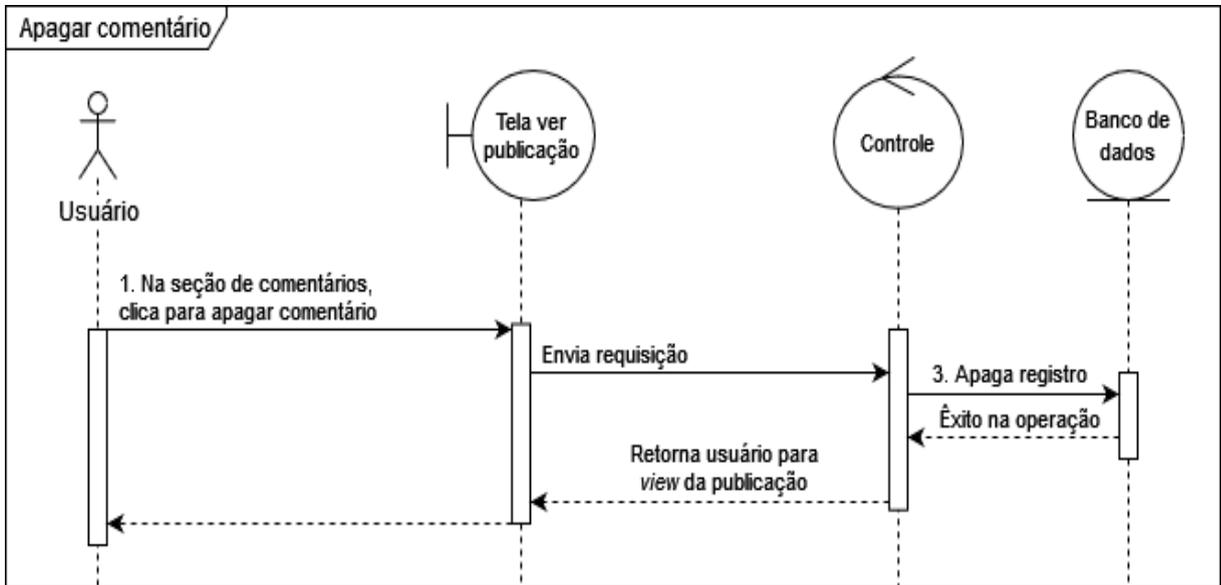
As Figuras 16 e 17 ilustram os diagramas de sequência para o Caso de Uso Manter Comentário.

Figura 16 - Diagrama de sequência do fluxo principal do caso de uso manter comentário



Fonte: Própria (2023)

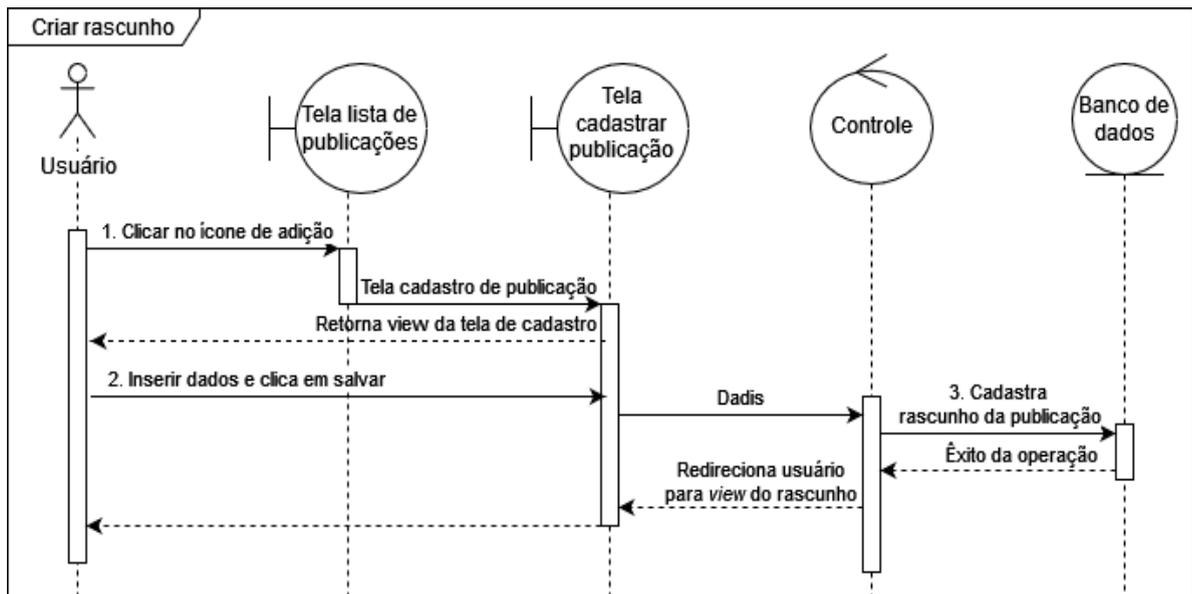
Figura 17 - Diagrama de sequência do fluxo alternativo I do caso de uso manter comentário



Fonte: Própria (2023)

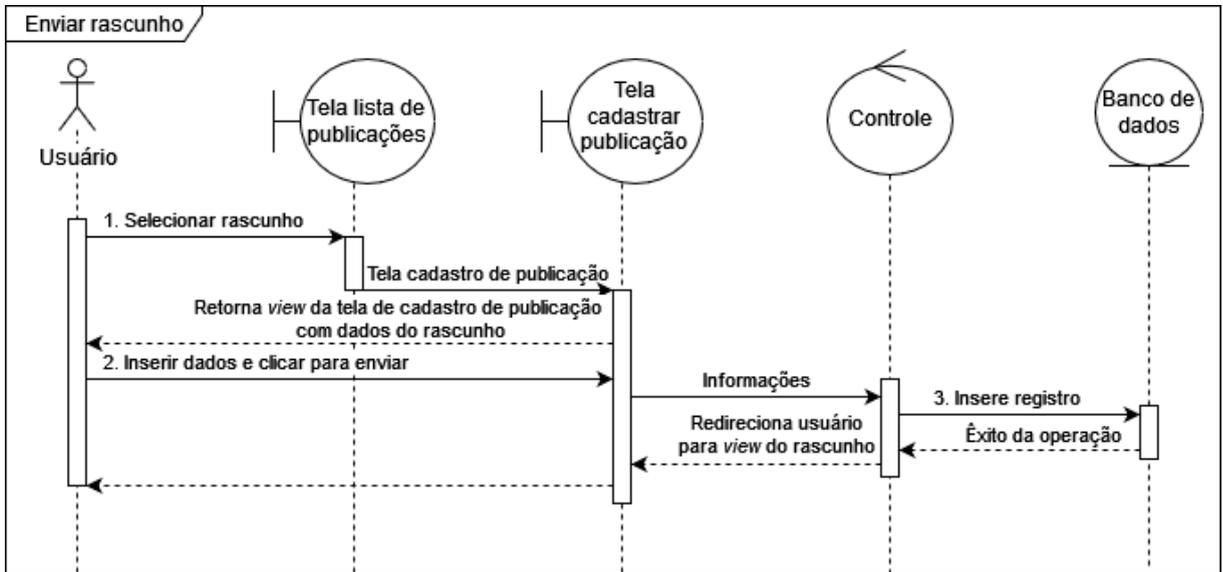
Ilustrados nas Figuras 18, 19 e 20 estão os diagramas de sequência referentes ao Caso de Uso Manter Rascunho.

Figura 18 - Diagrama de sequência do fluxo principal do caso de uso manter rascunho



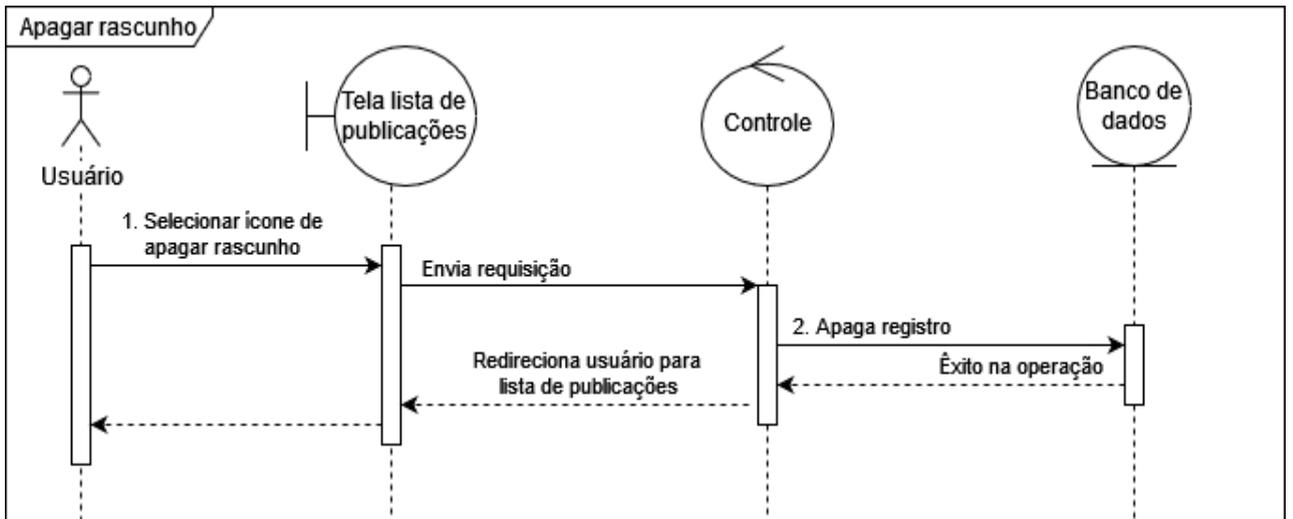
Fonte: Própria (2023)

Figura 19 - Diagrama de sequência do fluxo alternativo I do caso de uso manter rascunho



Fonte: Própria (2023)

Figura 20 - Diagrama de sequência do fluxo alternativo II do caso de uso manter rascunho



Fonte: Própria (2023)

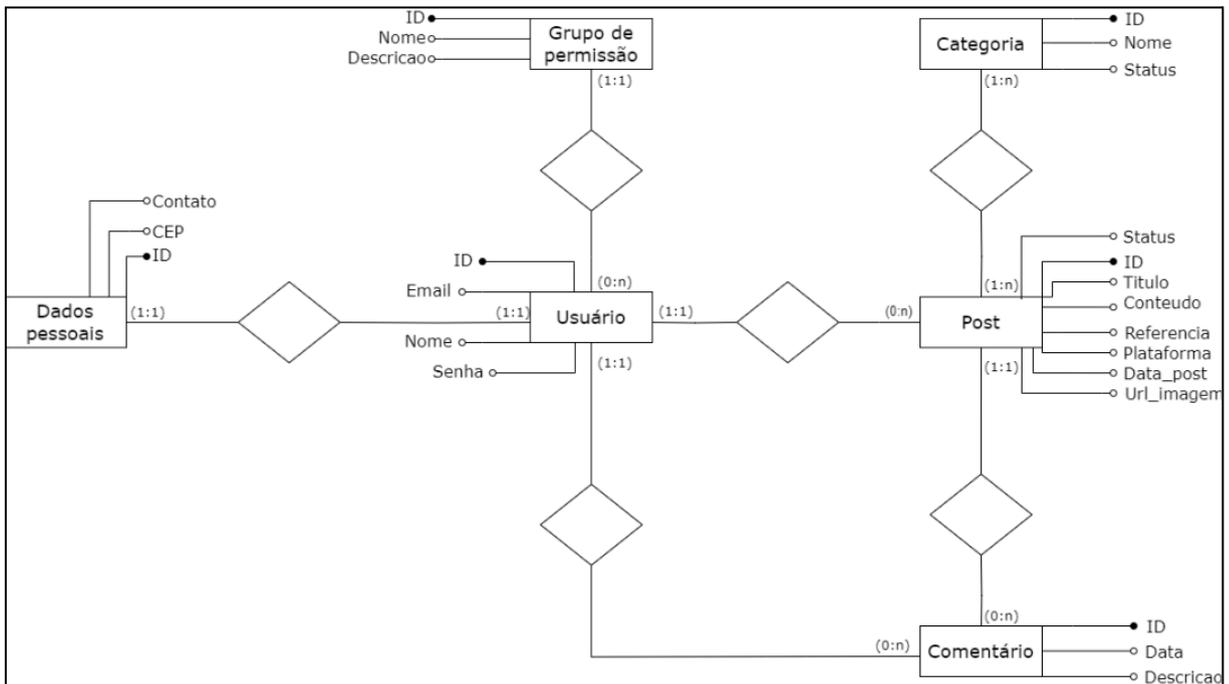
2.3 Banco de Dados

No Quebra-Galho, o banco de dados auxiliará armazenando e organizando os dados referente aos dados de usuários, publicações, comentários, postagens, categorias e grupos de permissão.

2.3.1 Diagrama Entidade Relacionamento

A figura 21 representa o Diagrama Entidade-Relacionamento (DER), onde podemos observar as tabelas de usuário e dados pessoais, que armazenam dados de e-mail, nome, senha, CEP e forma de contato; grupo de permissão o nome do grupo e descrição; categoria o seu nome e um status para saber se a mesma está ativa ou não; comentário a data e a descrição que será o corpo do comentário e a entidade post o status, da mesma maneira que categoria, título, conteúdo, referência das informações informadas, plataforma (web, mobile ou desktop), data da publicação e a URL da imagem relacionada com esse post.

Figura 21 - Modelo entidade relacionamento



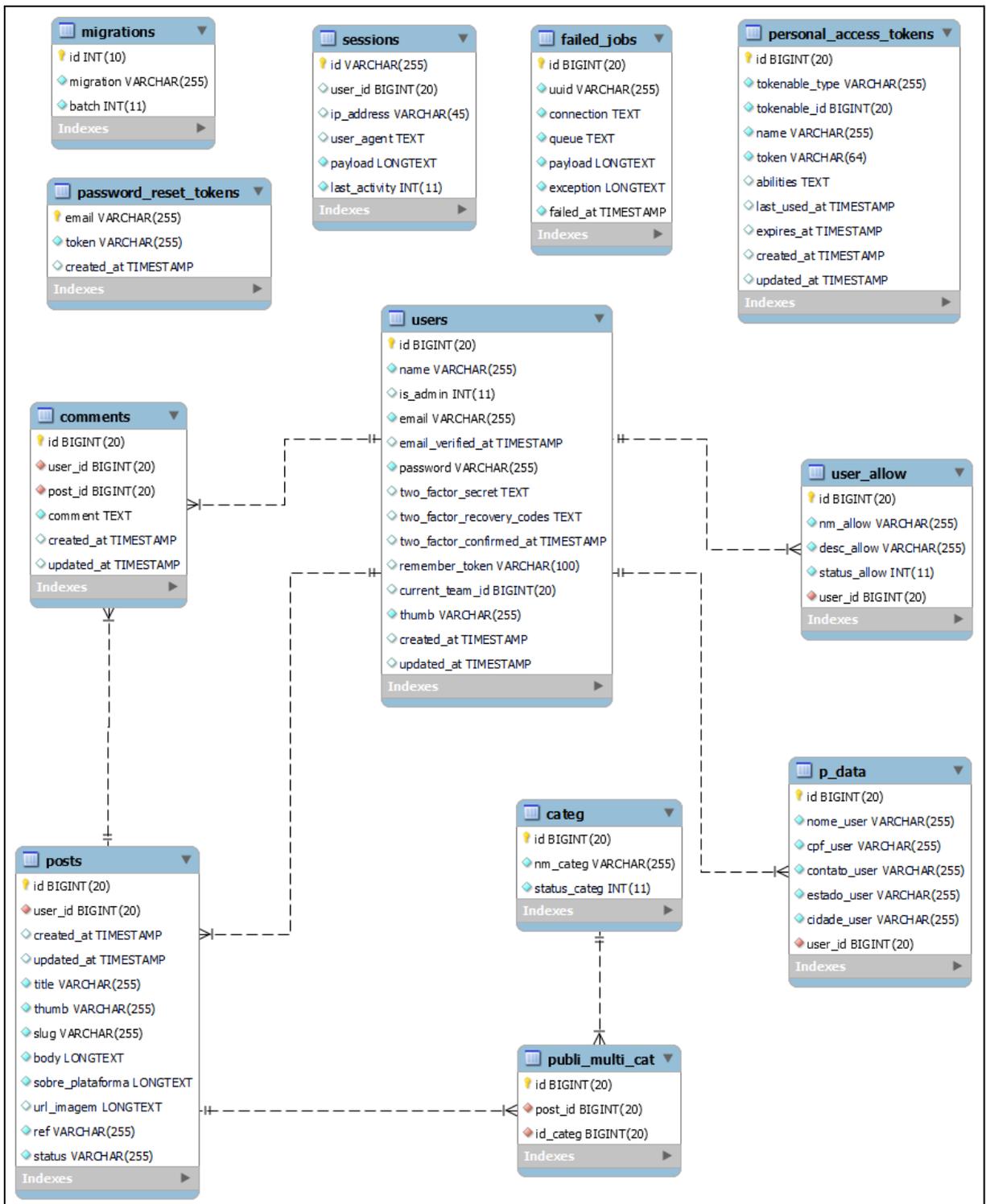
Fonte: Própria (2023)

2.3.2 Modelo Lógico

A modelagem lógica de banco de dados é a etapa em que se define a estrutura do banco de dados, utilizando conceitos e elementos que permitirão sua criação e implementação em um sistema.

Na modelagem lógica seguem as mesmas entidades do diagrama DER, porém com as tabelas do próprio Laravel que são as: de *sessions*, que é um *log* das sessões do sistema, *migrations* que são armazenados dados quantos às *migrations* e *password_reset_tokens* que é relacionado às alterações de senha de usuário.

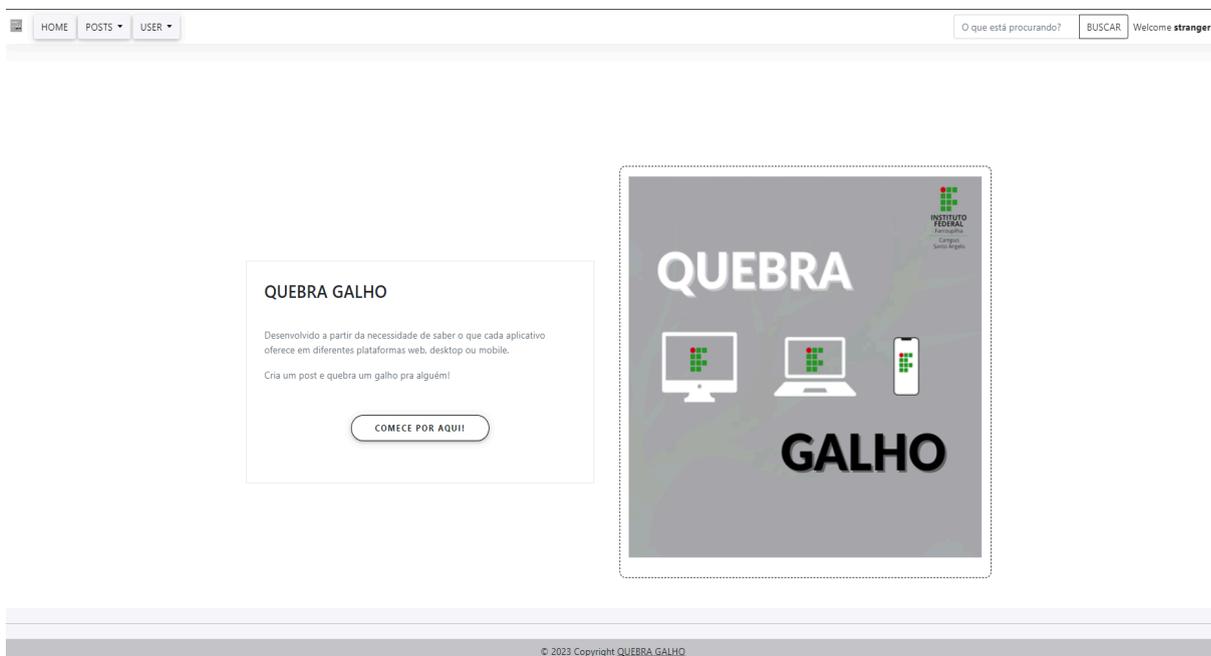
Figura 22 - Modelo lógico



Fonte: Própria (2023)

superior com botões de redirecionamento a outras áreas do site. Essas funcionalidades que aparecem o usuário estando logado ou não.

Figura 24 - Tela inicial do sistema



Fonte: Própria (2023)

Para acesso a algumas funções do sistema, como fazer publicações e comentários é necessário ter registro no sistema e estar com o e-mail verificado, após o usuário se registrar, é enviado automaticamente um e-mail verificação e até ser verificado algumas funcionalidades permanecerão bloqueadas a este usuário. A figura 25 apresenta a tela login do sistema, enquanto a tela 26 mostra a tela de registro.

Figura 25 - Tela de login do sistema



A tela de login do sistema 'QUEBRA GALHO' apresenta um cabeçalho com o nome do sistema e ícones de dispositivos. Abaixo, há um formulário com campos para 'Email' e 'Senha', uma opção 'Lembrar de mim neste computador' e botões para 'Novo por aqui?', 'Esqueci minha senha' e 'ENTRAR'.

QUEBRA GALHO

Email

Senha

Lembrar de mim neste computador

[Novo por aqui?](#)

[Esqueci minha senha](#) **ENTRAR**

Fonte: Própria (2023)

Figura 26 - Tela de registro de usuário



A tela de registro de usuário do sistema 'QUEBRA GALHO' contém um formulário com campos para 'Nome', 'E-mail', 'Senha' e 'Confirmação de Senha'. Também possui um link 'Já registrado?' e um botão 'REGISTRAR'.

QUEBRA GALHO

Nome

E-mail

Senha

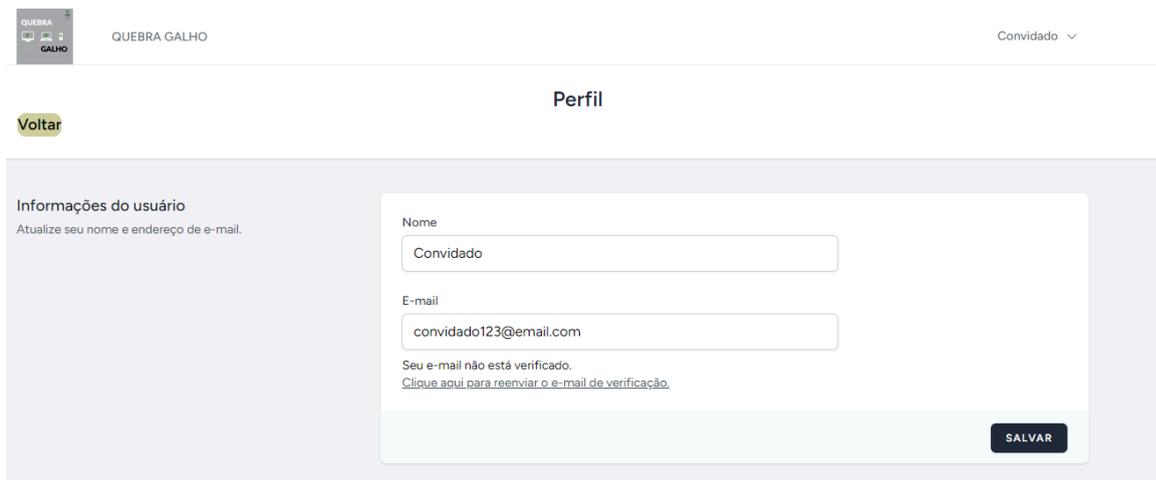
Confirmação de Senha

[Já registrado?](#) **REGISTRAR**

Fonte: Própria (2023)

Na figura 27 ilustra-se a tela de informações do usuário, podendo ser observado que este usuário não está com o e-mail verificado.

Figura 27 - Tela de informações do usuário

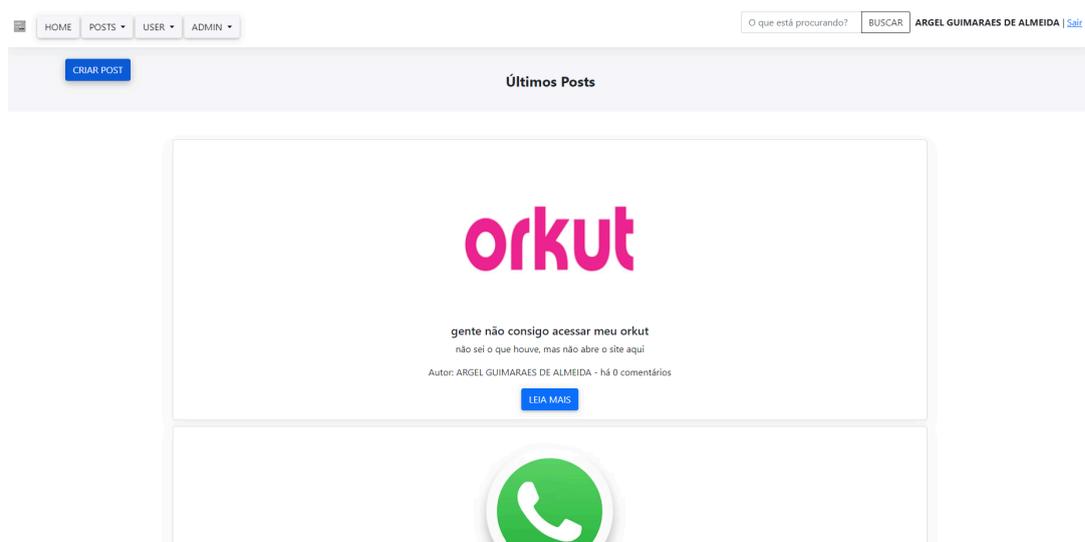


The screenshot shows a user profile page for 'QUEBRA GALHO'. The user is identified as 'Convidado'. The profile information section includes a 'Voltar' button and a heading 'Informações do usuário' with the instruction 'Atualize seu nome e endereço de e-mail.'. The user's name is 'Convidado' and their email is 'convidado123@email.com'. A message states 'Seu e-mail não está verificado.' with a link to 'Clique aqui para reenviar o e-mail de verificação.'. A 'SALVAR' button is located at the bottom right of the form.

Fonte: Própria (2023)

Na figura 28 apresenta-se a página das últimas publicações realizadas no sistema.

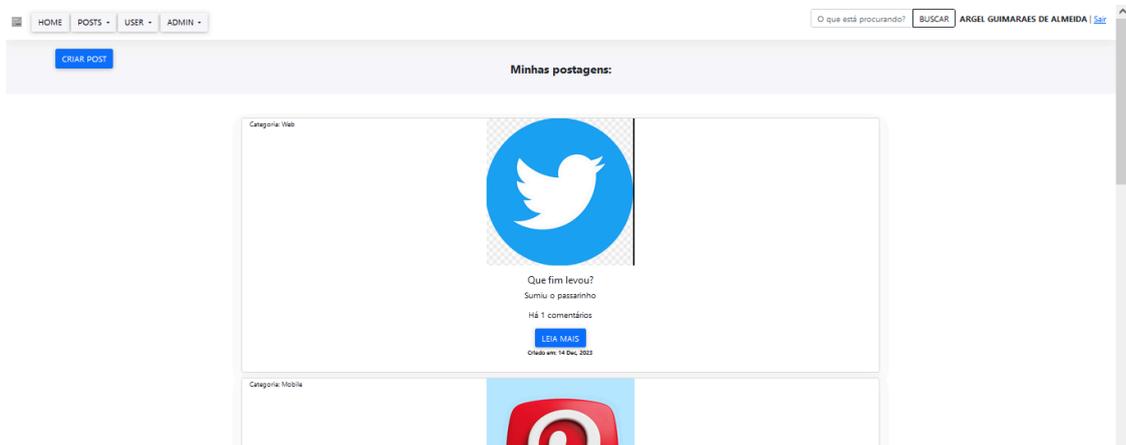
Figura 28 - Tela das publicações



Fonte: Própria (2023)

Na figura 29 é apresentada a tela das publicações do usuário *logado*, tela que é acessível a partir da barra de navegação superior em “Meus posts”.

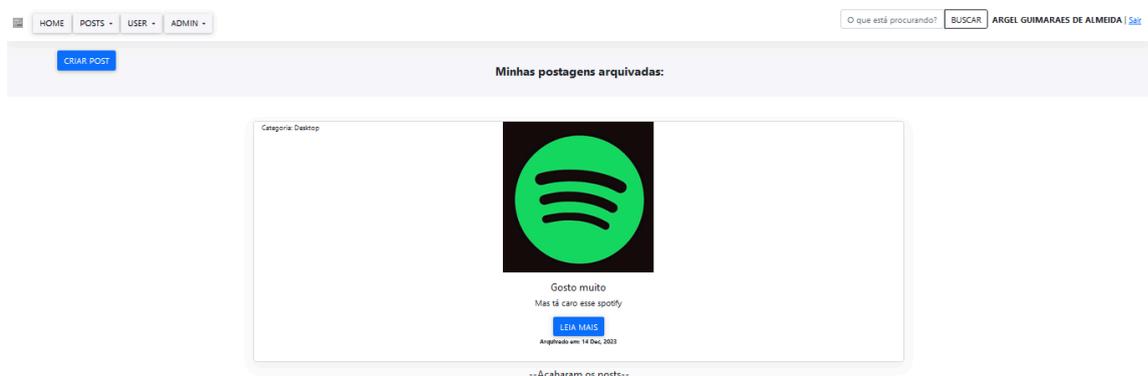
Figura 29 - Tela de publicações por usuário



Fonte: Própria (2023)

A figura 30 nos introduz à tela das publicações arquivadas do usuário *logado*, publicações que após terem sido postadas, foram arquivadas. Esta tela é acessível a partir da barra de navegação superior em “Meus posts arquivados”.

Figura 30 - Tela de publicações por usuário



Fonte: Própria (2023)

A seguir, na figura 31 consta a tela de usuários, com acesso somente para usuários com nível de administrador.

Figura 31 - Tela de Usuários



The screenshot shows a web interface for user management. At the top, there are navigation tabs for HOME, POSTS, and USER. A search bar is present with the text 'Buscar um usuário' and a 'BUSCAR' button. The user's name 'ARGEL GUIMARAES DE ALMEIDA' and a 'Sair' link are visible in the top right. The main content area is titled 'Todos os usuários' and contains a table with the following data:

ID	Nome	Email	Criado em:	Mudar senha	Deletar	Admin
6	Rosalia Kohler	michel.brown@esamole.net	2 Nov. 2023			ATUALIZAR PRIVILÉGIOS
7	Olivia Prosacco DDS	laila.gersen@esamole.com	2 Nov. 2023			ATUALIZAR PRIVILÉGIOS

Fonte: Própria (2023)

Ao tentar acessar alguma função que exige o usuário autenticado no sistema e o mesmo não está com seu e-mail verificado, o sistema irá retornar a tela apresentada na figura 30.

Figura 32 - Tela de verificação de e-mail



Fonte: Própria (2023)

A figura 33 ilustra a tela de cadastro de publicações, disponível após o usuário realizar a confirmação de e-mail.

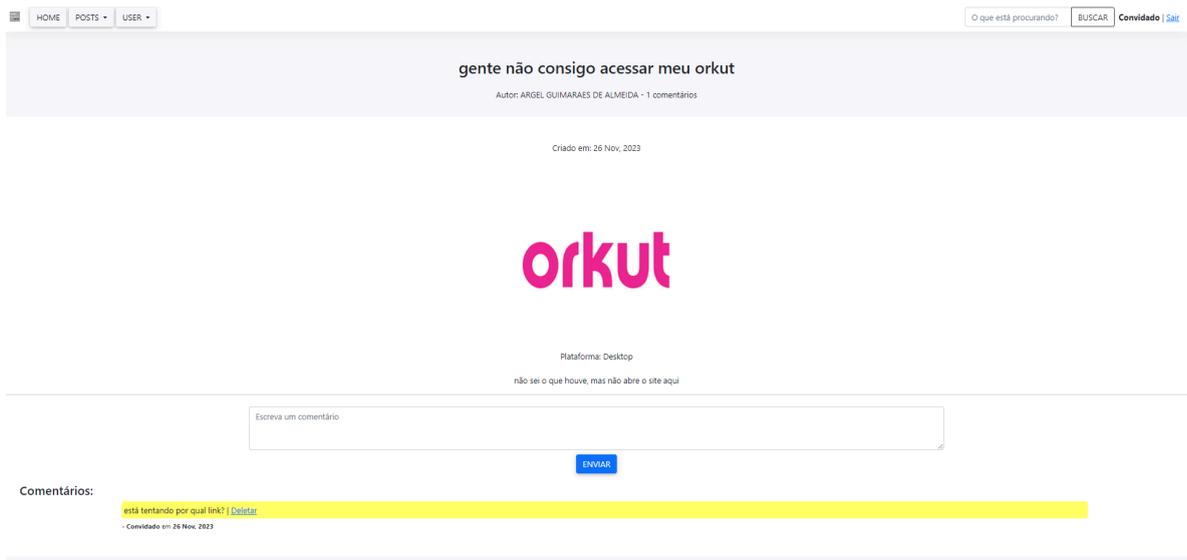
Figura 33 - Tela de criação de publicação

A imagem mostra a tela de criação de uma publicação. No topo, há uma barra de navegação com os links "HOME", "POSTS" e "USER". À direita, há uma barra de busca com o texto "O que está procurando?" e o botão "BUSCAR". Abaixo da barra de navegação, há o título "Novo post". O formulário contém os seguintes campos: "Título" (campo de texto), "Fale sobre:" (campo de texto com ícone de upload), "Qual a plataforma?" (com opções de radio button para "Web", "Desktop" e "Mobile"), "Referência:" (campo de texto). Abaixo do campo "Referência:", há um campo de busca para arquivos com o texto "Procurar:" e "Nenhum arquivo selecionado.". Abaixo disso, há o texto "Formatos de arquivo suportados: .jpeg, .png, .jpg, .gif, .svg". No rodapé do formulário, há três botões: "CRIAR POST" (em azul), "SALVAR COMO RASCUNHO" (em azul) e "VOLTAR" (em cinza). No rodapé da página, há o texto "© 2023 Copyright: Aopé Guimarães".

Fonte: Própria (2023)

A tela que representa os dados de uma publicação, é apresentada na Figura 33, e na parte inferior é possível observar a seção de comentários, podendo visualizar os comentários presentes ou inseri-lo.

Figura 32 - Tela da publicação



Fonte: Própria (2023)

3 RESULTADOS E DISCUSSÕES

3.1 Questionário de avaliação e testes dos Sistemas

Foram realizados testes do Sistema *Quebra Galho* presencialmente com cinco voluntários. O objetivo dos testes foi verificar a confiabilidade e usabilidade do sistema, além de identificar possíveis falhas que precisavam ser corrigidas.

Após os testes os usuários receberam um *link* de acesso para um formulário do Google previamente desenvolvido, onde havia perguntas sobre:

- O grau de usabilidade do sistema em uma escala de um a cinco, onde um representa muito difícil de usar e cinco muito fácil;
- Opiniões em uma escala de ruim a excelente sobre os principais módulos do sistema:

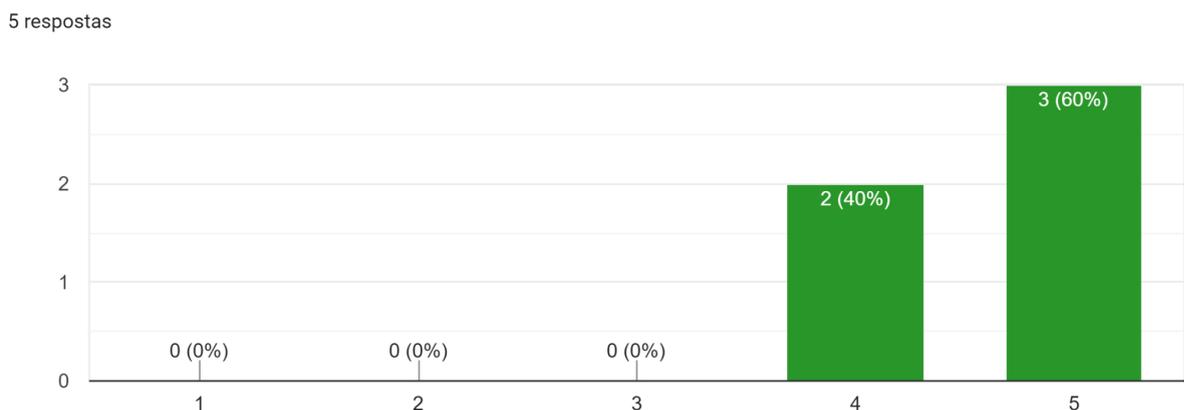
- Login;
- Usuário;
- Publicação;
- Comentários.
- Probabilidade de recomendar o sistema em uma escala de um a cinco;
- O *feedback* dos usuários sobre quais funções acharam mais úteis e se sentiram falta de algum recurso no sistema.

3.1.1 Resultados do Sistema

O sistema foi testado com cinco usuários com diferentes níveis de conhecimento sobre a informática e a área de desenvolvimento web, onde para três deles foi explicado como funcionava o sistema antes de usarem, e para dois foram testes “às cegas”, para ter uma amostragem diferente de dados.

O gráfico de barras da Figura 31 demonstra o índice de satisfação quanto a usabilidade do sistema, onde 60% das respostas foram para o índice cinco e 40% para o índice quatro, obtendo um alto grau de satisfação quanto a usabilidade.

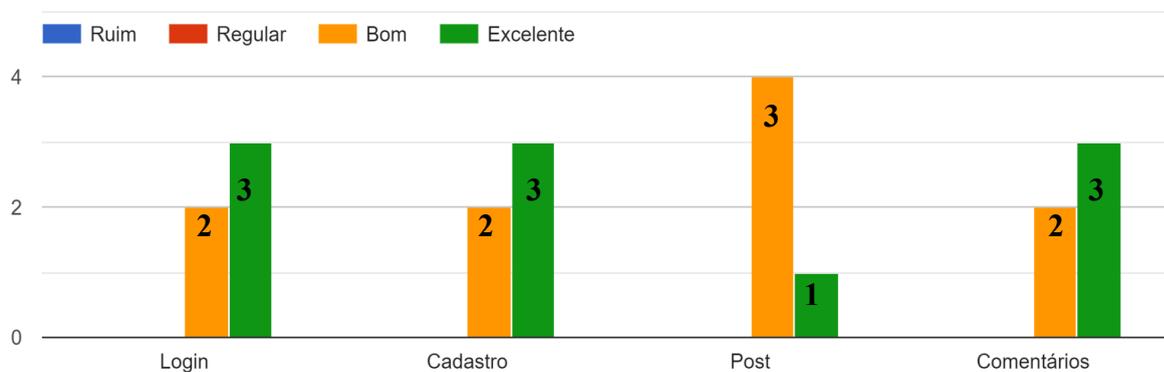
Figura 31 - Como você avaliaria a facilidade de uso do sistema em uma escala de 1 a 5?



Fonte: Própria (2023)

O gráfico de barras da Figura 32 apresenta as respostas sobre a aprovação dos usuários aos principais módulos, onde todos os votos foram para bom ou excelente.

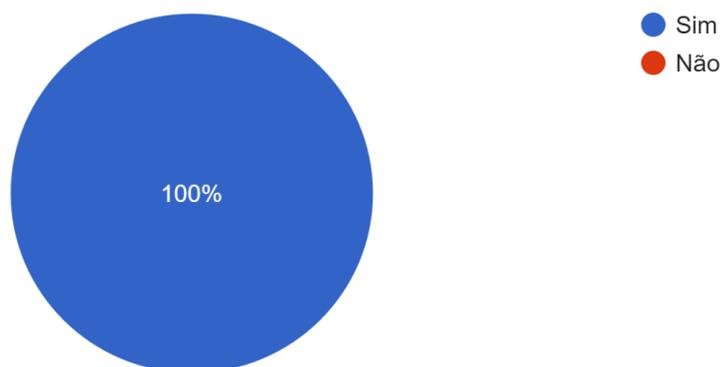
Figura 32 - Sobre as funcionalidades



Fonte: Própria (2023)

A Figura 33 mostra que todos os usuários votaram concordando que o design e layout do sistema são intuitivos, totalizando 100% dos votos.

Figura 33 - O design e layout do sistema são intuitivos para você? (Fáceis de usar)

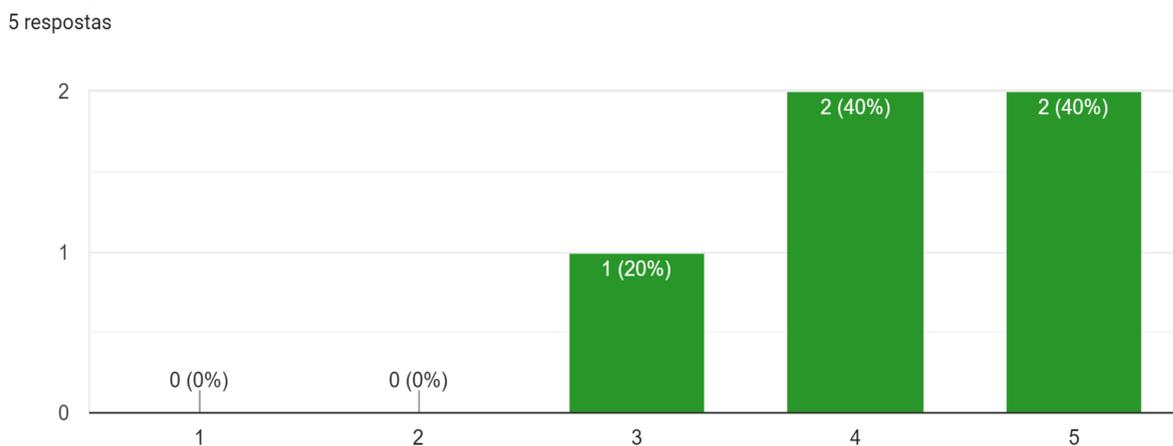


Fonte: Própria (2023)

A Figura 34 apresenta a probabilidade dos usuários recomendarem o sistema, onde 1

representa “baixa probabilidade” e 5 representa “alta probabilidade”. Sendo assim, foram obtidas uma resposta para a categoria “três”, duas respostas para a categoria “quatro”, e duas respostas para a categoria “cinco”. Os resultados demonstram que a maioria dos usuários poderia recomendar o sistema a outras pessoas.

Figura 33 - Qual a probabilidade de você recomendar o sistema?



Fonte: Própria (2023)

A questão dissertativa foi utilizada para coletar ideias de sugestão de melhorias no sistema e também para ver o que os usuários acharam mais útil.

Nas questões dissertativas foram deixadas sugestões de layout:

- Inserção de botão de *login* em todas as áreas do site, não somente na tela inicial como estava anteriormente
- Botão de criação de publicações em todas as telas relacionadas às publicações;
- Botão de voltar em algumas telas relacionadas às publicações e usuários;
- Busca não somente pelo post, mas também pelas categorias.

E também foram deixados alguns elogios ao sistema:

- Ao recurso de gravação de comentários antes de realizar o post;
- Ao sistema de arquivamento, rascunho e arquivamento de post;

- Local visível para o arquivamento e postagens;
- A sua usabilidade.

4 CONSIDERAÇÕES FINAIS

Com base nos testes realizados e nas horas de desenvolvimento do projeto, foram identificadas melhorias que podem ser implementadas no futuro, como a troca dentro do sistema das exclusões por desativações, pensando que sempre quem estiver usando pode querer desfazer uma ação. Além disso, foi notado que há uma necessidade de oferecer mais opções de personalização para os usuários, como mudança de cores, fontes e layouts. Outra sugestão seria a inclusão de um sistema de avaliação de postagens, para que os usuários possam indicar se um conteúdo é útil ou não, o que pode ajudar a melhorar a qualidade das discussões. Também nota-se que a plataforma poderia se beneficiar de um sistema de moderação mais eficiente, com a possibilidade de os próprios usuários denunciarem conteúdos inapropriados. Essas são apenas algumas ideias que podem ser implementadas para tornar o sistema ainda mais útil e agradável para todos os usuários. E por fim, uma função muitas vezes citada, foi a aprovação dos posts, que seria útil para essa filtragem do que é útil ou não, juntamente com o sistema de notificações, para o usuário saber o que está acontecendo com sua publicação até ela ser publicada.

Em uma era em que a internet oferece uma sobrecarga de informações, é essencial que os sistemas sejam confiáveis e fáceis de usar. Por isso, cada vez mais empresas investem em tecnologias que garantam a segurança e a praticidade dos usuários. Desde softwares de proteção de dados até sistemas de inteligência artificial que facilitam a navegação, a busca por soluções inovadoras é constante. Entretanto, é importante lembrar que a humanização dessas tecnologias também é fundamental. Afinal, não basta ser confiável e fácil de usar, é preciso que os usuários se sintam acolhidos e compreendidos em suas necessidades. Os sistemas que conseguem aliar tecnologia e empatia têm muito a ganhar em um mundo cada vez mais conectado.

O sistema cumpre com seus objetivos de confiabilidade com os níveis de usuário implementados e o sistema de autenticação e também a vantagem do sistema é a sua facilidade de uso. Com uma interface simples e intuitiva, os usuários podem navegar facilmente pelo sistema e encontrar as informações que precisam sem dificuldade. Isso torna o sistema acessível para usuários de todas as idades e habilidades técnicas, além de se constituir como um canal de informações relevantes sobre aplicativos, que pode ser construído colaborativamente.

REFERÊNCIAS

APACHEFRIENDS. Sobre o XAMPP. Disponível em: https://www.apachefriends.org/pt_br/about.html. Acesso em: 16 mar. 2023.

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. Rio de Janeiro: Elsevier. ISBN-13: 978-85-352-2626-3. 2015.

BOOTSTRAP. Introdução ao Bootstrap. Disponível em: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>. Acesso em: 09 mar. 2023.

COMPOSER. Introdução. Disponível em: <https://getcomposer.org/doc/00-intro.md>. Acesso em: 16 mar. 2023.

COODESH. O que é POO em programação? Disponível em: <https://coodesh.com/blog/dicionario/o-que-e-poo-em-programacao/>. Acesso em: 16 mar. 2023.

CSS. CSS. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em: 20 abr. 2023.

DATE, Chris J. An Introduction to Database Systems. 8. ed. Boston: Pearson/Addison Wesley, 2003.

DEVMEDIA. Os 4 pilares da Programação Orientada a Objetos. Disponível em: <https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>. Acesso em: 16 mar. 2023.

DIAGRAMS.NET. About diagrams.net. Disponível em: <https://www.diagrams.net/about>. Acesso em: 09 mar. 2023.

DVORSKY, Dalibor D. Installing, Configuring, and Developing with XAMPP. Março de 2007. Disponível em: <http://dalibor.dvorski.net/downloads/docs/installingconfiguringdevelopingwithxampp.pdf>. Acesso em: 16 mar. 2023.

Fowler, Martin. Patterns of Enterprise Application Architecture. Addison-Wesley, 2003.

GAMMA, Erich et al. Padrões de Projeto: Soluções reutilizáveis de software Orientado a Objetos. Porto Alegre: Bookman, 2000.

Horstmann, C. S. (2013). Big Java: Late Objects. John Wiley & Sons.

HTML. HTML: Linguagem de marcação de hipertexto. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 20 abr. 2023.

LARAVEL. Documentação do Laravel. Disponível em: <https://laravel.com/docs/10.x>. Acesso em: 16 mar. 2023.

MYSQL. Introdução ao MySQL Workbench. Disponível em: <https://dev.mysql.com/doc/>. Acesso em: 16 mar. 2023.

PHP. Manual do PHP. Disponível em: https://www.php.net/manual/pt_BR/. Acesso em: 20 mar. 2023.

WORLD WIDE WEB CONSORTIUM (W3C). HTML & CSS. Disponível em: <https://www.w3.org/standards/webdesign/htmlcss>. Acesso em: 16 mar. 2023.

APÊNDICE A : MODELO FÍSICO DO SISTEMA

-- Projeto: Quebra Galho

-- Autor: Argel Guimarães

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- Schema quebra galho

CREATE SCHEMA IF NOT EXISTS `quebra galho` DEFAULT CHARACTER SET utf8mb4 ;

USE `quebra galho` ;

-- Table `quebra galho`.`users`

CREATE TABLE IF NOT EXISTS `quebra galho`.`users` (
 `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
 `name` VARCHAR(255) NOT NULL,
 `is_admin` INT(11) NULL DEFAULT NULL,
 `email` VARCHAR(255) NOT NULL,
 `email_verified_at` TIMESTAMP NULL DEFAULT NULL,
 `password` VARCHAR(255) NOT NULL,
 `two_factor_secret` TEXT NULL DEFAULT NULL,
 `two_factor_recovery_codes` TEXT NULL DEFAULT NULL,
 `two_factor_confirmed_at` TIMESTAMP NULL DEFAULT NULL,
 `remember_token` VARCHAR(100) NULL DEFAULT NULL,
 `current_team_id` BIGINT(20) UNSIGNED NULL DEFAULT NULL,
 `thumb` VARCHAR(255) NOT NULL,

```

`created_at` TIMESTAMP NULL DEFAULT NULL,
`updated_at` TIMESTAMP NULL DEFAULT NULL,
PRIMARY KEY (`id`),
UNIQUE INDEX `users_email_unique` (`email` ASC))
ENGINE = InnoDB
AUTO_INCREMENT = 223
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_unicode_ci;
-----
-- Table `quebra galho`.`posts`
-----

CREATE TABLE IF NOT EXISTS `quebra galho`.`posts` (
  `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
  `user_id` BIGINT(20) UNSIGNED NOT NULL,
  `created_at` TIMESTAMP NULL DEFAULT NULL,
  `updated_at` TIMESTAMP NULL DEFAULT NULL,
  `title` VARCHAR(255) NOT NULL,
  `thumb` VARCHAR(255) NOT NULL,
  `slug` VARCHAR(255) NOT NULL,
  `body` LONGTEXT NOT NULL,
  `sobre_plataforma` LONGTEXT NOT NULL,
  `url_imagem` LONGTEXT NULL DEFAULT NULL,
  `ref` VARCHAR(255) NOT NULL,
  `status` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `posts_user_id_foreign` (`user_id` ASC),
  CONSTRAINT `posts_user_id_foreign`
    FOREIGN KEY (`user_id`)
    REFERENCES `quebra galho`.`users` (`id`)
    ON DELETE CASCADE)
ENGINE = InnoDB

```

AUTO_INCREMENT = 185

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4_unicode_ci;

-- Table `quebra galho`.`categ`

```
CREATE TABLE IF NOT EXISTS `quebra galho`.`categ` (  
  `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `nm_categ` VARCHAR(255) NOT NULL,  
  `status_categ` INT(11) NOT NULL,  
  `publi_id` BIGINT(20) UNSIGNED NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `categ_publi_id_foreign` (`publi_id` ASC),  
  CONSTRAINT `categ_publi_id_foreign`  
    FOREIGN KEY (`publi_id`)  
    REFERENCES `quebra galho`.`posts` (`id`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_unicode_ci;
```

-- Table `quebra galho`.`comments`

```
CREATE TABLE IF NOT EXISTS `quebra galho`.`comments` (  
  `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `user_id` BIGINT(20) UNSIGNED NOT NULL,  
  `post_id` BIGINT(20) UNSIGNED NOT NULL,  
  `comment` TEXT NOT NULL,  
  `created_at` TIMESTAMP NULL DEFAULT NULL,  
  `updated_at` TIMESTAMP NULL DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `comments_user_id_foreign` (`user_id` ASC),
```

```

INDEX `comments_post_id_foreign` (`post_id` ASC),
CONSTRAINT `comments_post_id_foreign`
  FOREIGN KEY (`post_id`)
  REFERENCES `quebra galho`.`posts` (`id`)
  ON DELETE CASCADE,
CONSTRAINT `comments_user_id_foreign`
  FOREIGN KEY (`user_id`)
  REFERENCES `quebra galho`.`users` (`id`)
  ON DELETE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 123
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_unicode_ci;
-----
-- Table `quebra galho`.`failed_jobs`
-----

CREATE TABLE IF NOT EXISTS `quebra galho`.`failed_jobs` (
  `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
  `uuid` VARCHAR(255) NOT NULL,
  `connection` TEXT NOT NULL,
  `queue` TEXT NOT NULL,
  `payload` LONGTEXT NOT NULL,
  `exception` LONGTEXT NOT NULL,
  `failed_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP(),
  PRIMARY KEY (`id`),
  UNIQUE INDEX `failed_jobs_uuid_unique` (`uuid` ASC))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_unicode_ci;
-----
-- Table `quebra galho`.`migrations`

```

```
-----  
CREATE TABLE IF NOT EXISTS `quebra galho`.`migrations` (  
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `migration` VARCHAR(255) NOT NULL,  
  `batch` INT(11) NOT NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 18  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_unicode_ci;  
-----
```

```
-- Table `quebra galho`.`p_data`  
-----
```

```
CREATE TABLE IF NOT EXISTS `quebra galho`.`p_data` (  
  `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `nome_user` VARCHAR(255) NOT NULL,  
  `cpf_user` VARCHAR(255) NOT NULL,  
  `contato_user` VARCHAR(255) NOT NULL,  
  `estado_user` VARCHAR(255) NOT NULL,  
  `cidade_user` VARCHAR(255) NOT NULL,  
  `user_id` BIGINT(20) UNSIGNED NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `p_data_cpf_user_unique` (`cpf_user` ASC),  
  INDEX `p_data_user_id_foreign` (`user_id` ASC),  
  CONSTRAINT `p_data_user_id_foreign`  
    FOREIGN KEY (`user_id`)  
    REFERENCES `quebra galho`.`users` (`id`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_unicode_ci;  
-----
```

-- Table `quebra galho`.`password_reset_tokens`

```
-----  
CREATE TABLE IF NOT EXISTS `quebra galho`.`password_reset_tokens` (  
  `email` VARCHAR(255) NOT NULL,  
  `token` VARCHAR(255) NOT NULL,  
  `created_at` TIMESTAMP NULL DEFAULT NULL,  
  PRIMARY KEY (`email`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_unicode_ci;  
-----
```

-- Table `quebra galho`.`personal_access_tokens`

```
-----  
CREATE TABLE IF NOT EXISTS `quebra galho`.`personal_access_tokens` (  
  `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `tokenable_type` VARCHAR(255) NOT NULL,  
  `tokenable_id` BIGINT(20) UNSIGNED NOT NULL,  
  `name` VARCHAR(255) NOT NULL,  
  `token` VARCHAR(64) NOT NULL,  
  `abilities` TEXT NULL DEFAULT NULL,  
  `last_used_at` TIMESTAMP NULL DEFAULT NULL,  
  `expires_at` TIMESTAMP NULL DEFAULT NULL,  
  `created_at` TIMESTAMP NULL DEFAULT NULL,  
  `updated_at` TIMESTAMP NULL DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `personal_access_tokens_token_unique` (`token` ASC),  
  INDEX `personal_access_tokens_tokenable_type_tokenable_id_index` (`tokenable_type` ASC, `tokenable_id`  
  ASC))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_unicode_ci;  
-----
```

-- Table `quebra galho`.`publi_multi_cat`

```
-----  
CREATE TABLE IF NOT EXISTS `quebra galho`.`publi_multi_cat` (  
  `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `post_id` BIGINT(20) UNSIGNED NOT NULL,  
  `id_categ` BIGINT(20) UNSIGNED NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `publi_multi_cat_post_id_foreign` (`post_id` ASC),  
  INDEX `publi_multi_cat_id_categ_foreign` (`id_categ` ASC),  
  CONSTRAINT `publi_multi_cat_id_categ_foreign`  
    FOREIGN KEY (`id_categ`)  
      REFERENCES `quebra galho`.`categ` (`id`),  
  CONSTRAINT `publi_multi_cat_post_id_foreign`  
    FOREIGN KEY (`post_id`)  
      REFERENCES `quebra galho`.`posts` (`id`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_unicode_ci;  
-----
```

-- Table `quebra galho`.`sessions`

```
-----  
CREATE TABLE IF NOT EXISTS `quebra galho`.`sessions` (  
  `id` VARCHAR(255) NOT NULL,  
  `user_id` BIGINT(20) UNSIGNED NULL DEFAULT NULL,  
  `ip_address` VARCHAR(45) NULL DEFAULT NULL,  
  `user_agent` TEXT NULL DEFAULT NULL,  
  `payload` LONGTEXT NOT NULL,  
  `last_activity` INT(11) NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `sessions_user_id_index` (`user_id` ASC),  
  INDEX `sessions_last_activity_index` (`last_activity` ASC))
```

```

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4_unicode_ci;

-----

-- Table `quebra galho`.`user_allow`
-----

CREATE TABLE IF NOT EXISTS `quebra galho`.`user_allow` (
  `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
  `nm_allow` VARCHAR(255) NOT NULL,
  `desc_allow` VARCHAR(255) NOT NULL,
  `status_allow` INT(11) NOT NULL,
  `user_id` BIGINT(20) UNSIGNED NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `user_allow_user_id_foreign` (`user_id` ASC),
  CONSTRAINT `user_allow_user_id_foreign`
    FOREIGN KEY (`user_id`)
      REFERENCES `quebra galho`.`users` (`id`))
ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4_unicode_ci;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```