

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

MARCO ANTONIO PEGORARO

**SISTEMA WEB PARA DIVULGAÇÃO DE MORADIA, ATIVIDADES
COMPLEMENTARES E VAGAS DE ESTÁGIOS**

PATO BRANCO

2023

MARCO ANTONIO PEGORARO

**SISTEMA WEB PARA DIVULGAÇÃO DE MORADIA, ATIVIDADES
COMPLEMENTARES E VAGAS DE ESTÁGIOS**

**WEB SYSTEM FOR HOUSING DISSEMINATION, COMPLEMENTARY
ACTIVITIES AND INTERNSHIP VACANCIES**

Trabalho de conclusão de curso de graduação,
apresentado como requisito para obtenção do título de
Tecnólogo em Análise e Desenvolvimento de Sistemas
da Universidade Tecnológica Federal do Paraná
(UTFPR).

Orientadora: Prof^a. Andreia Scariot Beulke.

PATO BRANCO

2023



[4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

MARCO ANTONIO PEGORARO

**SISTEMA WEB PARA DIVULGAÇÃO DE MORADIA, ATIVIDADES
COMPLEMENTARES E VAGAS DE ESTÁGIOS**

Trabalho de conclusão de curso de graduação,
apresentado como requisito para obtenção do título de
Tecnólogo em Análise e Desenvolvimento de Sistemas
da Universidade Tecnológica Federal do Paraná
(UTFPR).

Data de aprovação: 19/junho/2023

Andreia Scariot Beulke
Mestrado
Universidade Tecnológica Federal do Paraná

Mariza Miola Dosciatti
Doutorado
Universidade Tecnológica Federal do Paraná

Lucília Yoshie Araki
Mestrado
Universidade Tecnológica Federal do Paraná

PATO BRANCO

2023

RESUMO

A rotina de um estudante universitário pode ser limitada devido ao pouco tempo para realizar as atividades concomitantes ao trabalho. Assim, é necessário que o aluno estabeleça meios alternativos para organizar seu tempo, como definir metas e prioridades para conciliar as suas atividades diárias. A fim de auxiliar estudantes que se encontram nesse contexto, esse trabalho tem por objetivo desenvolver uma aplicação *web* para publicação e divulgação de informações relevantes para os acadêmicos. O sistema possui funcionalidades semelhantes às de uma rede social, para que alunos, professores e empresas possam buscar e divulgar vagas de estágio, horas complementares, vagas de moradia e outros, como por exemplo eventos e esportes. O sistema conta com opções para permitir que os usuários possam interagir de forma mais engajada com publicações de interesse em comum relacionadas ao ambiente acadêmico. A experiência que a maioria das pessoas têm em utilizar redes sociais torna o aplicativo mais atrativo para o uso cotidiano. O projeto foi desenvolvido com o uso das ferramentas Spring Boot para a realização do *back-end* da aplicação, React e o *framework* Bootstrap para o desenvolvimento do *front-end*.

Palavras-chave: redes sociais; universitários; aplicação *web*; busca de informação.

ABSTRACT

The routine of a college student can be limited due to the lack of time to carry out concurrent activities with work. Thus, it is necessary for the student to establish alternative means to organize their time, such as setting goals and priorities to reconcile their daily activities. In order to assist students in this context, this project aims to develop a web application for publishing and disseminating relevant information for academics. The system has functionalities similar to those of a social network, allowing students, professors, and companies to search for and advertise internships, complementary hours, housing vacancies, and others. The "others" category includes events, sports, etc. The system includes features that enable users to engage more actively with publications of common interest related to the academic environment. The familiarity that most people have with using social networks makes the application more appealing for everyday use. The project was developed using Spring Boot tools for the back-end implementation, React and the Bootstrap framework for front-end development.

Keywords: social networks; university students; web application; information search.

LISTA DE QUADROS

Quadro 1 – Lista de ferramentas e tecnologias	21
Quadro 2 – Requisitos funcionais	25
Quadro 3 – Requisitos não funcionais	26
Quadro 4 – Operação de inclusão nos casos de uso “manter”	27
Quadro 5 – Operação alterar nos casos de uso “manter”	28
Quadro 6 – Operação de exclusão nos casos de uso “manter”	29
Quadro 7 – Operação de aprovar publicação	30
Quadro 8 – Operação de seguir perfis de usuário	30
Quadro 9 – Operação de interagir com publicação.....	31

LISTA DE CÓDIGOS FONTE

Listagem 1 – Configuração de permissão.....	41
Listagem 2 – <i>Controller</i> das publicações.....	42
Listagem 3 – Interface contendo métodos padrões do <i>service</i>	44
Listagem 4 – Classe implementando os métodos padrões.....	44
Listagem 5 – <i>Repository</i> de publicação	45
Listagem 6 – Busca de publicações	46
Listagem 7 – Método para salvar curtida	47
Listagem 8 – Método envio de e-mail para publicações reprovadas.....	48
Listagem 9 – Método verificação de token.....	50
Listagem 10 – Método de busca de publicações.....	50
Listagem 11 – Estruturação do cartão do sistema.....	51

LISTA DE FIGURAS

Figura 1 – Diagrama de Caso de Uso	27
Figura 2 – Diagrama de entidades e relacionamentos do banco de dados	32
Figura 3 – Tela inicial do sistema	33
Figura 4 – Tela de perfil	34
Figura 5 – Tela de edição de perfil.....	35
Figura 6 – Tela de administrador	36
Figura 7 – Confirmação de aprovação de publicação.....	36
Figura 8 – Tela de reprovação de publicação	37
Figura 9 – E-mail contendo o conteúdo de reprovação de publicação	37
Figura 10 – Tela de criação de publicação.....	38
Figura 11 – Tela de busca por publicações	38
Figura 12 – Tela de perfil externo	39
Figura 13 – Tela de publicações curtidas.....	40
Figura 14 – Organização dos pacotes <i>back-end</i>	40
Figura 15 – Métodos dentro do <i>useEffect</i>	49

LISTA DE ABREVIATURAS E SIGLAS

CRUD	<i>Create, Read, Update, Delete</i>
CSS	<i>Cascading Style Sheets</i>
DTO	<i>Data Transfer Object</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
RF	Requisitos Funcionais
RNF	Requisitos Não Funcionais
SQL	<i>Structured Query Language</i>
TCC	Trabalho de conclusão de curso
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	19
1. INTRODUÇÃO	13
1.1 CONSIDERAÇÕES INICIAIS.....	13
1.2 OBJETIVOS.....	14
1.2.1 OBJETIVO GERAL.....	14
1.2.2 OBJETIVOS ESPECÍFICOS.....	15
1.3 JUSTIFICATIVA.....	15
1.4 ESTRUTURA DO TRABALHO	16
2. REFERENCIAL TEÓRICO.....	17
2.1 COTIDIANO UNIVERSITÁRIO E ACESSO À INFORMAÇÃO.....	17
2.2 REDES SOCIAIS	18
3. MATERIAIS E MÉTODO.....	21
3.1 MATERIAIS.....	21
3.2 MÉTODO.....	22
4. RESULTADOS.....	24
4.1 ESCOPO DO SISTEMA.....	24
4.2 MODELAGEM DO SISTEMA	25
4.3 APRESENTAÇÃO DO SISTEMA.....	33
4.4 IMPLEMENTAÇÃO DO SISTEMA	40
5 CONCLUSÃO	54

1. INTRODUÇÃO

Este capítulo apresenta as considerações iniciais, os objetivos e a justificativa da realização deste trabalho. No final do capítulo é apresentada a organização do texto que contém a estruturação dos seus capítulos.

1.1 Considerações Iniciais

A rotina de um estudante universitário é bem atarefada, pois muitos estudantes precisam trabalhar e estudar para conseguir manter os custos básicos como moradia, alimentação, transporte e educação. Segundo pesquisas, o número de estudantes que trabalham ao mesmo tempo que fazem um curso superior subiu de 97 mil (11,2%) em 2015 para 110 mil (12,8%) nos últimos anos. A tendência é que esse número continue a crescer, no Espírito Santo por exemplo, em 2020 esse número chegou já a 116 mil (13,2%), batendo o recorde da região (TRIBUNAONLINE, 2020).

Dessa forma, o tempo de um estudante se torna escasso, por realizarem as atividades universitárias e as inerentes ao trabalho concomitantes e, assim, precisam organizar a rotina de trabalho com a vida acadêmica. Deste modo, os estudantes precisam estabelecer metas, prioridades e objetivos para conciliar essas atividades de forma satisfatória em ambas as esferas.

Muitos estudantes vêm de outras regiões ou estados e precisam encontrar um local para residir, e a maioria prefere ou precisa dividir aluguel com outros estudantes para diminuir os custos mensais. Além disso, no cenário acadêmico, estudantes precisam fazer Estágio Curricular Supervisionado, seja na modalidade Não Obrigatório para ter uma renda mensal, ou obrigatório que é uma unidade curricular que faz parte da integralização dos cursos de graduação, além de atividades complementares que também devem ser desenvolvidas dentro do prazo de conclusão do curso.

A ampla divulgação de informações como moradia, vagas de estágio, oportunidades de emprego e de atividades complementares são importantes para que os estudantes se mantenham atualizados sobre as oportunidades que surgem. Uma forma de divulgar essas oportunidades é por meio dos recursos tecnológicos, sendo as redes sociais o mais conhecido e utilizado.

Desde o surgimento da Internet, as redes sociais vêm transformando o dia a dia das pessoas. Uma das grandes tendências que a Internet proporcionou foram as redes sociais, que trouxeram grandes mudanças na forma que nos relacionamos e vivemos o cotidiano. As redes sociais conseguiram se tornar uma tendência mundial pois “com apenas um clique, qualquer pessoa pode acessar uma informação específica e manter contato com pessoas que estão distantes” (CRISTINA, 2014).

Tendo em vista que atualmente a maioria das pessoas possui uma conta pessoal em uma rede social, sendo utilizadas por pessoas de diferentes faixas etárias. Pesquisas indicam que a média global de tempo diário acessando Internet é 6 horas 42 minutos e a média do Brasil é 9 horas 29 minutos (WE ARE SOCIAL, 2019). Isso demonstra como a internet, e as redes sociais presentes nelas foram realmente uma febre mundial, especialmente no Brasil. Outros dados indicam que aproximadamente 70% dos jovens entre 15 e 24 anos, incluídos em uma pesquisa de campo, utilizavam as redes sociais como fonte de busca de informação sobre produtos, notícias, entre outros (CARVALHO; MURBACK, 2014, p. 15).

Nesse cenário, considerando a importância da divulgação de informações relevantes para os acadêmicos e o amplo acesso às redes sociais, neste trabalho foi desenvolvido um sistema web com funcionalidades semelhantes às de uma rede social para que alunos, professores e empresas possam buscar e divulgar vagas de estágio, horas complementares e moradia, para tornar esse processo mais otimizado.

1.2 Objetivos

A seguir serão apresentados o objetivo geral e os objetivos específicos do presente trabalho, sendo o objetivo geral relacionado com o resultado principal da realização deste trabalho e os objetivos específicos sendo um complemento para o geral em termos de funcionalidades do sistema.

1.2.1 Objetivo Geral

Desenvolver um sistema web para divulgação de informações relevantes para acadêmicos, tais como: vagas de moradia, estágio e oportunidades para a realização de atividades complementares.

1.2.2 Objetivos Específicos

- Possibilitar a divulgação de informações relevantes para acadêmicos como oportunidades de moradia, estágio e atividades complementares.
- Disponibilizar um ambiente centralizado para a divulgação de oportunidades para acadêmicos visando amplo acesso.
- Permitir o acesso e a busca de informações de forma rápida para os estudantes porque estarão centralizadas em um único ambiente.

1.3 Justificativa

Os celulares e computadores permitiram que o ser humano mudasse a maneira de se comunicar e ver o mundo. Junto com o advento da Internet vieram diversas facilidades para o dia a dia das pessoas, pois surgiram diversos sites, sistemas e redes sociais para facilitarem demandas cotidianas como compra e venda de produtos e serviços, entretenimento ou divulgação e compartilhamento de publicações de qualquer natureza.

No ambiente acadêmico, muitas vezes, os estudantes preferem não morar em repúblicas, pensões ou, ainda, a cidade onde ele irá estudar não oferece esse tipo de serviço. Assim, o estudante precisa encontrar um local para morar, como uma casa, apartamento, apart-hotel ou similar e, muitas vezes, é necessário dividir o aluguel com outra pessoa, pois o orçamento para arcar com as despesas de moradia individualmente pode não ser viável. Além disso, durante o período em que o aluno permanece na universidade é necessário fazer estágio e horas complementares para concluir o curso, ou ainda o aluno precisa trabalhar para manter os seus custos mensais. Dessa forma, o estudante precisa manter-se atualizado das ofertas de emprego, estágio e outras informações que são consideradas importantes ou que são necessárias para sua permanência na cidade e na universidade.

Assim sendo, ambientes virtuais, que permitem o compartilhamento de informações necessárias para os estudantes, podem ser um meio interessante de agregar e disponibilizar conteúdos de forma a permitir o acesso e a busca de informações mais rapidamente. É nesse cenário que o desenvolvimento desse trabalho se justifica, pois propõe o desenvolvimento de um sistema web semelhante às redes sociais para o compartilhamento e busca das informações supracitadas.

De forma geral, o intuito de desenvolver o sistema proposto é tornar o acesso às informações para os estudantes mais rápido porque elas estão centralizadas em um único ambiente. Isso permite que o estudante possa encontrar essas informações mais facilmente, pois atualmente, é necessário que eles passem um tempo considerável em busca de informações relacionadas à universidade, tendo que acessar diversos meios digitais para encontrar o que precisa, fazendo com que o processo seja demorado.

1.4 Estrutura do Trabalho

Este texto está organizado em capítulos dos quais este é o primeiro e apresenta as considerações iniciais, os objetivos e a justificativa do trabalho. O Capítulo 2 apresenta o referencial teórico e no Capítulo 3 são apresentados os materiais e o método utilizados para o desenvolvimento do trabalho. No Capítulo 4 está o resultado da realização do trabalho.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta os tópicos principais que apoiam o desenvolvimento deste trabalho de conclusão de curso. Serão abordadas informações referentes ao cotidiano da rotina universitária, que muitas vezes pode ser agitada e com desafios a serem enfrentados. Além disso, também será discutido a interação dos alunos e o acesso à informação e às redes sociais, buscando compreender as formas que isso ocorre e o impacto que essas interações exercem diariamente.

2.1 Cotidiano Universitário e Acesso à Informação

O cotidiano estudantil pode ser desafiador pois a rotina dos estudantes, muitas vezes, é corrida, pois o estudante precisa comparecer assiduamente às aulas e realizar as atividades acadêmicas, como aprender os conteúdos ministrados em aulas, realizar atividades de verificação de aprendizagem como exercícios, trabalhos e avaliações, participar de projetos de pesquisa e extensão, realizar atividades complementares para enriquecer o processo de ensino-aprendizagem, realizar Estágio Obrigatório e o Trabalho de Conclusão de Curso (TCC). Além disso, muitos estudantes precisam trabalhar para conseguir manter os custos básicos, como moradia, alimentação, transporte e educação. Segundo pesquisa realizada entre jovens, com idade entre 19 a 24 anos que estudam, o percentual daqueles que também trabalham subiu de 45,4% em 2016 para 48,3% em 2019 (SURGIU, 2020).

Tendo em vista essa realidade, atualmente muitos estudantes utilizam as mídias sociais para conseguir realizar seus afazeres diários de forma mais rápida e simples. Alguns exemplos para demonstrar essa maior facilidade e rapidez que as redes sociais trazem para o cotidiano estudantil são a busca de produtos em mercados e lojas, maior facilidade em se comunicar com colegas de profissão ou com acadêmicos, a busca por informações referentes ao seu curso de graduação.

Devido ao advento da internet, a forma de obter qualquer tipo de informação se tornou algo mais fácil, pois os jovens estão inseridos em um contexto no qual a informação é distribuída em variadas mídias, em tempo real, de fácil acesso e disponíveis em praticamente todos os lugares (GROSSI et al., 2014). Com o uso da internet e as redes sociais presentes nela:

Os indivíduos podem trocar informações, experiências, criar comunidades virtuais ou fóruns de discussão que tratam de temas específicos, bem como

divulgar ideias ou conceitos que acreditam ser importantes para si e para as comunidades nas quais estão inseridos (GROSSI et al., 2014, p. 11).

Essa facilidade proporcionada pela internet é de extrema importância para os estudantes, pois além de ser um ambiente que muitos já conhecem, por já terem nascido na era da informação, ela também proporciona uma maior otimização de tempo. “Redes sociais de compartilhamento podem prover contribuições para a educação tais como melhorar a comunicação entre alunos, gerar oportunidade de conhecerem novos ambientes ou pessoas e eliminar problemas de comunicação nos ambientes de ensino” (MESE, 2018, p. 191, tradução nossa). Essas vantagens também podem ser vistas em outros setores da sociedade como o setor de venda de produtos, no qual, devido às redes sociais, grandes empresas agora têm um meio de comunicação mais fácil e rápido para conseguir entender o gosto e tendências entre seus consumidores, além de também de facilitar na divulgação de seus novos produtos.

2.2 Redes Sociais

Com o surgimento dos primeiros computadores após a segunda guerra mundial e, posteriormente, a criação da internet, o mundo sofreu uma grande transformação com essas tecnologias.

Com uma breve panorâmica da história de vários séculos de civilização humana, descobrimos que a informação tem sido um dos principais fatores para o desenvolvimento e evolução das civilizações. As sociedades que de forma rápida e com precisão compilam e analisam as suas informações e as utilizam corretamente têm tido sempre mais sucesso do que outras (TAYEBI, 2019, p. 245, tradução nossa).

Corroborando com Taeby (2019), a informação sempre foi importante, pois desde a época primitiva o ser humano vivia em pequenos grupos ou comunidades e utilizava uma forma mais rudimentar de comunicação e informação como gestos, gritos, pinturas e desenhos, e, com o uso da internet e das redes sociais, ela se tornou ainda mais importante, pois torna a propagação da informação e comunicação rápida e com alcance mundial. Contudo, pode-se dizer que em todas as épocas, desde a primitiva até a contemporânea, a informação e a comunicação são os principais meios para se relacionar e garantir grupos de contatos.

No mundo contemporâneo, as redes sociais são um exemplo de mídia de relacionamento que mais se assemelha a uma sociedade humana e permitindo aos

indivíduos se comunicarem com outras pessoas, independentemente do tempo, lugar, restrições políticas, culturais e econômicas (TAYEBI, 2019, p. 245, tradução nossa). Esse estilo de comunicação e a velocidade que as redes sociais trouxeram, mudaram completamente diversas áreas da sociedade. A forma de busca de informação na área da educação e pesquisa, atualmente não é como se os tradicionais livros e revistas científicas impressos tivessem perdido seus status, mas a rapidez com que se atualizam os tornam ineficazes para alguns campos de conhecimento, sendo inevitável que estes se adaptem ao formato virtual de acesso, como evidenciado em diversas editoras ou revistas eletrônicas (LORENA, 2019). Isso mostra que os livros físicos, artigos científicos ou consensos de entidades ou órgãos oficiais nunca deixarão de serem utilizados, mas é evidente que a internet proporciona maior rapidez de atualização e publicação desses materiais. Além disso, a internet permite que as pessoas se comuniquem de forma mais rápida que a tradicional (ligações telefônicas ou contatos pessoais), pois fornece recursos de envio de mensagens instantâneas e rápidas de forma coletiva, formando assim, uma rede de comunicação.

Conforme destaca Cerqueira e Silva (2011), a sociedade é organizada em rede e as redes sociais sempre existiram, pois, cada pessoa é um ponto na rede e estabelece conexões com outras pessoas, formando grupos de interação. Os autores ainda destacam que com os avanços tecnológicos, os sites participativos e de auto expressão fazem com que as pessoas interajam umas com as outras compartilhando informações aumentando a proximidade entre elas.

Sendo assim, as empresas e organizações têm se inserido no cenário das mídias sociais como importante ferramenta para fortalecer a comunicação e a publicidade organizacional. Isso porque as redes sociais permitem que as informações sejam disseminadas de forma rápida e fácil abrangendo pessoas do mundo inteiro em um curto período de tempo.

Nesse sentido, os consumidores têm acesso rápido às informações sobre um determinado produto ou serviço, tendo opiniões favoráveis ou desfavoráveis que são compartilhadas entre os usuários nas redes sociais.

Dentro do ambiente empresarial também se nutriu a ideia de que era possível cultivar “comunidades” paralelamente a marcas reconhecidas. A expectativa era de que esses ambientes virtuais servissem para promover valores associados a um produto ou a uma marca, mas logo ficou aparente que áreas de publicação livre como fóruns e salas de bate-papo frequentemente

tornavam-se canais de expressão para consumidores descontentes (SPYER, 2007, p. 26).

Assim, observa-se que as redes sociais fazem parte do cotidiano das pessoas ao redor do mundo. A maneira cotidiana das pessoas utilizarem a internet é por meio delas, de forma que, geralmente, pessoas de todas as idades compreendem o que são redes sociais e de como utilizá-las. Pesquisas indicam que 79% dos usuários ativos no Brasil fazem parte de alguma rede social, sendo que o país possui em torno de 35 milhões de pessoas conectadas (CARVALHO; MURBACK, 2014). Além disso, o acesso diário por parte dos brasileiros é um dos mais altos do mundo, segundo informações a média global de tempo diário acessando Internet é 6 horas e 42 minutos e a média do Brasil é 9 horas e 29 minutos (WE ARE SOCIAL, 2019). Tendo em vista esses dados, é difícil não notar como as redes sociais tomaram conta da nossa sociedade e de como estão tornando o nosso mundo um lugar muito mais conectado e sem barreiras, algo jamais visto na história humana até o presente momento.

3. MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e o método utilizados para o desenvolvimento deste trabalho. Os materiais elencam as ferramentas e as tecnologias utilizadas na modelagem, banco de dados, linguagem e *frameworks* para programação *front-end* e *back-end*. O método está relacionado a um modelo de processo prescritivo que visa ordenar e estruturar o desenvolvimento do sistema.

3.1 Materiais

O Quadro 1 apresenta a lista de ferramentas e tecnologias que serão utilizadas para o desenvolvimento deste trabalho.

Quadro 1 – Lista de ferramentas e tecnologias

Ferramenta / Tecnologia	Versão	Finalidade
Intellij Idea Ultimate	2022.2.1	Ambiente de desenvolvimento
CSS	3	Linguagem de estilização de conteúdo
SpringBoot (MVC, Data e Security)	2.7.3	<i>Framework Java</i>
Bootstrap	5.1	Biblioteca de estilização de conteúdo
HTML	5	Linguagem de marcação de conteúdo
PostgreSQL	13.2	Banco de Dados
DBeaver	21.0.2	Ferramenta de administração de banco de dados
Java	1.7	Linguagem de programação
LucidChart	-	Modelagem do sistema
React-ts	18.2.0	<i>Framework front-end</i>

Fonte: Autoria própria (2023).

A seguir será apresentada uma descrição mais detalhada do React e do Spring, por serem as principais tecnologias utilizadas no desenvolvimento deste projeto para justificar sua escolha e destacar os benefícios que elas proporcionam para o desenvolvimento do sistema.

O React é uma biblioteca JavaScript que serve para a criação de interfaces de usuário. Foi criada em 2011 pelo time do Facebook, surgiu com o objetivo de otimizar a atualização e a sincronização de atividades simultâneas no *feed* de notícias da rede social. Contudo, por ter demonstrado uma grande eficiência nos anos seguintes, o React foi incorporado à interface de outras redes sociais do grupo, como o Instagram. Em 2013, o código do React foi aberto à comunidade, dando início a sua popularização e hoje se tornou uma das mais prestigiadas bibliotecas JavaScript do mercado. O uso do React traz diversas vantagens para o desenvolvimento da aplicação, pois ele conta com diversos componentes e *plugins* que podemos fazer uso durante o desenvolvimento, dessa forma, criando as telas do sistema de maneira mais rápida e fazendo uso de funcionalidades avançadas sem muito esforço. Além disso, atualmente o React traz integração com diversos outros sites e *frameworks* que podem ser utilizados para o desenvolvimento do *front-end*.

O Spring Boot é um *framework* Java *open source* que deriva do Spring, que é um *framework* criado em 2003 por Rod Johnson. O Spring Boot tem o objetivo de facilitar processos em aplicações Java, ele traz mais agilidade para o processo de desenvolvimento uma vez que consegue reduzir o tempo gasto com as configurações iniciais dos projetos.

3.2 Método

O projeto foi desenvolvido utilizando o método de desenvolvimento ágil por ser flexível e colaborativo para a criação de software. Em vez de um processo sequencial, ele enfatiza a entrega incremental e interação constante. Esse método valoriza a comunicação e a adaptação, permitindo ajustes conforme necessário. Como resultado, é possível responder melhor às mudanças, colaborar de forma eficiente e produzir produtos de alta qualidade.

Como pode ser visto na 'DevMedia' (Introdução ao Desenvolvimento Ágil, 2007), o manifesto ágil possui alguns princípios claros, como:

- **Adaptação às mudanças:** Em vez de resistir às mudanças nos requisitos, o desenvolvimento ágil aceita e incorpora ajustes ao longo do processo, garantindo que o produto final esteja alinhado com as necessidades e expectativas.

- **Auto-organização e responsabilidade:** O desenvolvimento ágil propõe que os membros sejam auto-organizados, tomando decisões colaborativas e assumindo responsabilidade pelo sucesso do projeto.
- **Interação e *feedback*:** O desenvolvimento ágil é baseado em ciclos curtos de trabalho, onde cada iteração resulta em um incremento do produto. O feedback contínuo é incorporado para melhorar o produto e direcionar o próximo ciclo.
- **Qualidade e melhoria contínua:** A busca pela qualidade é uma prioridade no desenvolvimento ágil, com ênfase em testes contínuos, integração e melhoria constante dos processos e práticas de desenvolvimento.

Em suma, os princípios do desenvolvimento ágil promovem uma abordagem flexível, colaborativa e centrada no cliente, permitindo a entrega de valor de forma mais eficiente e eficaz, adaptando-se às mudanças e buscando constantemente a excelência.

4. RESULTADOS

Este capítulo apresenta o resultado do presente trabalho que é o desenvolvimento de sistema *web* que possuirá funcionalidades parecidas com as das redes sociais e servirá para a divulgação de ofertas de estágio/emprego, atividades de horas complementares e vagas de moradia para estudantes.

4.1 Escopo do Sistema

O sistema *web* desenvolvido tem por objetivo auxiliar e atender estudantes que estão em busca de vagas de estágio, horas complementares e moradia. O sistema tem o propósito disponibilizar estas informações em um único ambiente.

A interface do sistema *web* tem o intuito de ser simples, com funcionalidades bem definidas. O usuário tem como principais funções buscar e ofertar vagas de estágio ou emprego, eventos, moradia ou oportunidades de atividades complementares e outras informações que podem ser importantes para os acadêmicos, como venda ou empréstimo de um móvel e equipamento, por exemplo. A ferramenta de busca do sistema tem um mecanismo de filtragem, para facilitar a busca por informações na qual o usuário pode filtrar a partir de alguma palavra-chave ou por categoria da publicação.

Para realizar o uso do sistema será necessário criar uma conta relacionada ao seu perfil. Para isso, o usuário precisa preencher um formulário que é disponibilizado ao acessar o sistema e clicar na opção para criar uma conta. Nas publicações, aparecerá a mensagem do usuário, e a categoria da publicação. As categorias das publicações são inicialmente divididas em cinco, sendo elas: moradia, evento, hora complementar, estágio ou outros. Novas categorias poderão ser adicionadas a partir do desejo dos usuários. O processo será simples, basta os usuários utilizarem a ferramenta de requisição de categoria no sistema, que irá pedir para os usuários inserirem a nova categoria que desejam inserir no sistema. Após isso, essa requisição de nova categoria será enviada para um administrador, que aceitará ou não a nova categoria, caso seja aceita, será incluída no sistema. O sistema possui a funcionalidade na qual os usuários poderão seguir uns aos outros para verem com mais frequência publicações referentes aos usuários seguidos e as publicações no sistema possuem as opções para curtir e compartilhar. Na área de texto, ao criar

as publicações, os usuários podem fazer o *upload* de vídeos, imagens ou então escrever textos.

As publicações registradas no sistema devem ser aprovadas pelo usuário com perfil de administrador para evitar que conteúdos indesejados ou fora de contexto sejam publicados. Caso a publicação seja indeferida, o sistema deverá enviar um e-mail ao usuário informando o motivo pelo qual sua publicação não ter sido aprovada.

Os usuários possuem perfis editáveis no qual poderão conectar com suas redes sociais e terão uma área para escrever um pouco sobre eles. Além disso, na publicação, na área de contato ficará explícito o nome do usuário que realizou a publicação, ao clicar no nome, o site redireciona o usuário para o perfil de quem fez a publicação.

4.2 Modelagem do Sistema

O Quadro 2 apresenta os Requisitos Funcionais (RF) que foram identificados para o sistema. Esses requisitos identificam as funcionalidades essenciais do sistema.

Quadro 2 – Requisitos funcionais

Identificação	Nome	Descrição
RF01	Manter usuário	Os usuários serão do tipo usuário comum (qualquer pessoa que deseja criar um perfil no sistema) que tenha interesse em divulgar uma publicação (vagas de moradia, estágio ou emprego e oportunidades para a realização de atividades complementares).
RF02	Manter publicação	Os usuários poderão manter publicações de seu interesse, podendo inserir, alterar e excluir os conteúdos. Cada publicação deverá ser vinculada a uma categoria, podendo ser: estágio, moradia, evento, emprego ou atividade complementar. Nas publicações o usuário poderá inserir vídeos, imagens ou texto referente ao conteúdo que será publicado.
RF03	Aprovar publicação	Para evitar que uma publicação tenha conteúdos indesejados ou fora de contexto, o usuário com perfil de administrador deverá aprovar a publicação.
RF04	Interagir com publicação	Os usuários poderão interagir com publicações de outros usuários, sendo possível curtir, compartilhar e comentar em uma publicação.
RF05	Seguir perfis de usuários	Usuários poderão acompanhar outros perfis para receberem notificações dos usuários seguidos de maneira mais frequente. Dessa forma, quando um usuário é seguido, é

		possível compartilhar ou curtir alguma publicação que será recebida no seu <i>feed</i> .
RF06	Pesquisar publicações	As publicações poderão ser pesquisadas a partir da categoria ou palavra chave.
RF07	Pesquisar usuários	Os usuários poderão ser pesquisados pelo nome. Na pesquisa é necessário digitar as letras do nome do usuário desejado e aparecerá uma lista contendo o nome e suas respectivas imagens de perfil.
RF08	Manter perfil de usuário	Após se cadastrar no sistema, o usuário terá um perfil próprio para modificar e inserir suas informações.
RF09	Filtrar publicações	Será possível filtrar publicações de acordo com a categoria desejada pelo usuário. Por exemplo: filtrar uma publicação de estágio, moradia ou emprego.

Fonte: Autoria própria (2023).

O Quadro 3 apresenta os Requisitos Não Funcionais (RNF) identificados para o sistema. Os requisitos não funcionais não estão relacionados diretamente às funcionalidades de um sistema, referindo-se mais ao desempenho, usabilidade e segurança do sistema.

Quadro 3 – Requisitos não funcionais

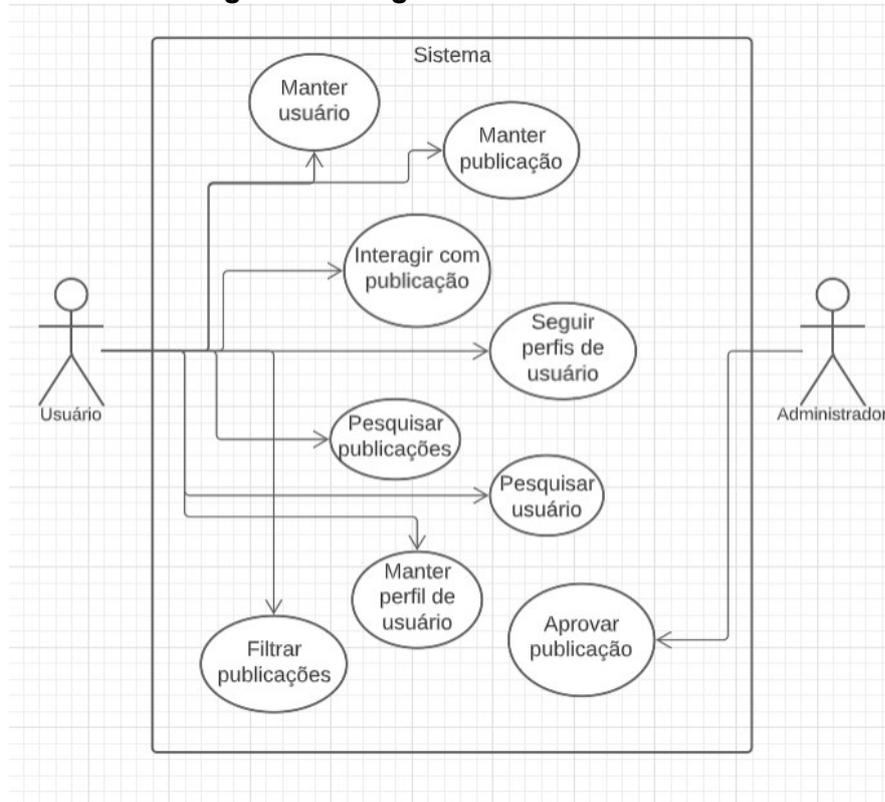
Identificação	Nome	Descrição
RNF01	Acesso ao sistema	O acesso ao sistema deverá ser efetuado a partir da inserção dos dados de login e senha do usuário. Caso ele ainda não possua os dados para realizar a autenticação no sistema, o sistema deve indicar o ambiente de cadastro para o usuário.
RNF02	Uso de <i>design</i> responsivo nas interfaces gráficas	O sistema deve contar com um <i>front-end</i> responsivo para que o <i>layout</i> possa ser compreendido e lido em diferentes dispositivos como <i>smartphones</i> ou <i>tablets</i> .
RNF03	Compatibilidade com diferentes navegadores	O sistema deve possuir compatibilidade com os principais navegadores <i>web</i> da atualidade, como Opera, Firefox, Chrome, etc.

Fonte: Autoria própria (2023).

A Figura 1 apresenta o diagrama de casos de uso, que contém as funcionalidades básicas do sistema e seus atores, sendo eles: usuário e administrador. O ator “Usuário” refere-se a todos os usuários do sistema, que podem ser professores, alunos e outras pessoas relacionadas ao ambiente acadêmico. Esses usuários possuem as mesmas permissões. O usuário com perfil de administrador irá

aprovar as publicações realizadas por outros usuários para garantir a integridade e credibilidade das informações publicadas.

Figura 1 – Diagrama de Caso de Uso



Fonte: Autoria própria (2023).

A seguir, é apresentada a expansão dos principais casos de uso do sistema. Os casos de uso definidos como “Manter” referem-se aos cadastros com as operações de inclusão, alteração, exclusão e consulta.

O Quadro 4 apresenta a expansão dos casos de uso para incluir um novo registro no sistema.

Quadro 4 – Operação de inclusão nos casos de uso “manter”

<p>Caso de uso: Manter operação incluir.</p> <p>Descrição: Esse caso de uso refere-se a operação de inclusão dos registros nos casos de uso “manter”.</p> <p>Evento Iniciador: Necessidade de incluir registros.</p>
<p>Atores: Usuário.</p>
<p>Precondições: Dados necessários disponíveis. Usuário estar na tela de cadastro do sistema.</p>

<p>Pós-condições: Inclusão realizada com sucesso.</p>
<p>Sequência típica de eventos (Fluxo Principal):</p> <ol style="list-style-type: none"> 1. [IN] O usuário seleciona a opção “cadastrar-se”. 2. [IN] O usuário informa seus dados nos campos apresentados pelo sistema. (1 exceção). 3. [OUT] O sistema emite uma mensagem de confirmação de novo cadastro. (2 exceções). 4. [OUT] O sistema redireciona o usuário para nova página.
<p>Tratamento de Exceções e Variantes:</p> <p>Exceção 2a: Dados incongruentes.</p> <p>2a.1 [OUT] O sistema informa que algum campo não foi preenchido ou foi preenchido com valor errôneo.</p> <p>2a.2 [OUT] Retorna ao passo 2.</p> <p>Exceção 3a: Erro de execução do sistema.</p> <p>3a.1 [OUT] O sistema informa que houve um erro na inserção dos dados.</p> <p>3a.2 [OUT] Retorna ao passo 2.</p>

Fonte: Autoria própria (2023).

O Quadro 5 apresenta a expansão do caso de uso da operação que permite alterar os dados no sistema.

Quadro 5 – Operação alterar nos casos de uso “manter”

<p>Caso de uso: Manter operação alterar.</p> <p>Descrição: Esse caso de uso refere-se a operação de edição dos registros nos casos de uso “manter”.</p> <p>Evento Iniciador: Necessidade de editar registros.</p>
<p>Atores: Usuário.</p>
<p>Precondições:</p> <ul style="list-style-type: none"> O usuário deve possuir uma conta no sistema. Usuário autenticado no sistema. Dados necessários disponíveis. Usuário estar na tela de alteração do sistema.
<p>Pós-condições: Alterações realizadas com sucesso.</p>
<p>Sequência típica de eventos (Fluxo Principal):</p> <ol style="list-style-type: none"> 1. [IN] O usuário seleciona a seção de informação ou informação que deseja editar. 2. [IN] O usuário informa os campos com os valores atuais deles e para qual informação deseja modificar. (1 exceção). 3. [OUT] O sistema emite uma mensagem de confirmação de atualização de dados. (2 exceções). 4. [OUT] O sistema esconde a tela de modificação de informações.

Tratamento de Exceções e Variantes:**Exceção 2a: Dados incongruentes.**

2a.1 [OUT] O sistema informa que algum campo não foi preenchido ou foi preenchido com valor errôneo.

2a.2 [OUT] Retorna ao passo 2.

Exceção 3a: Erro de execução do sistema.

3a.1 [OUT] O sistema informa que houve um erro na alteração dos dados.

3a.2 [OUT] Retorna ao passo 2.

Fonte: Autoria própria (2023).

O Quadro 6 apresenta a expansão do caso de uso da operação de exclusão dos dados no sistema.

Quadro 6 – Operação de exclusão nos casos de uso “manter”

<p>Caso de uso: Manter operação excluir</p> <p>Descrição: Esse caso de uso refere-se a operação de exclusão dos registros nos casos de uso “manter”.</p> <p>Evento Iniciador: Necessidade de excluir registros.</p>
<p>Atores: Usuário.</p>
<p>Precondições:</p> <ul style="list-style-type: none"> O usuário deve possuir uma conta no sistema. Usuário autenticado no sistema. Dados necessários disponíveis. Usuário estar na tela de exclusão do sistema.
<p>Pós-condições:</p> <p>Informação excluída com sucesso.</p>
<p>Sequência típica de eventos (Fluxo Principal):</p> <ol style="list-style-type: none"> 1. [IN] O usuário seleciona a opção “excluir”. 2. [OUT] O sistema emite mensagem de confirmação para exclusão da informação (1 exceção). 3. [IN] O usuário confirma a opção de exclusão da informação. 4. [OUT] O sistema emite uma mensagem de confirmação de exclusão de dados (2 exceções). 5. [OUT] O sistema redireciona o usuário para nova página.
<p>Tratamento de Exceções e Variantes:</p> <p>Exceção 2a: Usuário nega opção de exclusão.</p> <ul style="list-style-type: none"> 2a.1 [OUT] O sistema informa que usuário escolheu opção de negar exclusão. 2a.2 [OUT] Retorna ao passo 5. <p>Exceção 3a: Erro de execução do sistema.</p> <ul style="list-style-type: none"> 3a.1 [OUT] O sistema informa que houve um erro na exclusão dos dados. 3a.2 [OUT] Retorna ao passo 2.

Fonte: Autoria própria (2023).

O Quadro 7 apresenta a expansão do caso de uso para aprovar uma publicação.

Quadro 7 – Operação de aprovar publicação

<p>Caso de Uso: Aprovar publicação.</p> <p>Descrição: Administrador faz a verificação e aprovação de novas publicações que irão aparecer dentro do sistema.</p> <p>Evento Iniciador: Necessidade de verificação de publicações de usuário para não permitir publicações nocivas ou sem conteúdo coerente com o proposto pelo sistema.</p>
<p>Atores: Administrador.</p>
<p>Precondições: Administrador deve estar autenticado no sistema.</p>
<p>Pós-condições: Publicação aprovada para aparecer nas publicações do sistema.</p>
<p>Sequência típica de eventos (Fluxo Principal):</p> <ol style="list-style-type: none"> 1. [IN] O administrador seleciona uma publicação na lista de publicação pendentes. 2. [OUT] O sistema abre nova página com as informações da publicação selecionada. 3. [IN] O administrador aceita a postagem da publicação. (1 exceção). 4. [OUT] O sistema emite uma mensagem de confirmação de exclusão de dados. 5. [OUT] O sistema redireciona o administrador para nova página.
<p>Tratamento de Exceções e Variantes:</p> <p>Exceção 1a: Administrador nega postagem de publicação.</p> <ol style="list-style-type: none"> 1a.1 [IN] O administrador insere no campo de mensagem o motivo de negar a publicação. 1a.2 [OUT] O sistema emite uma mensagem de confirmação. 1a.3 [OUT] O sistema redireciona o administrador para nova página.

Fonte: Autoria própria (2023).

O Quadro 8 apresenta a expansão do caso de uso para seguir perfis de usuário.

Quadro 8 – Operação de seguir perfis de usuário

<p>Caso de Uso: Seguir perfis de usuário.</p> <p>Descrição: Usuário consegue seguir outro usuário para ver as publicações desse usuário seguido com maior frequência.</p> <p>Evento Iniciador: Necessidade do usuário de ficar mais informado sobre um outro perfil.</p>

Atores: Usuário.
Precondições: O usuário deve possuir uma conta no sistema. Usuário autenticado no sistema. O usuário deve estar na página de perfil de outro usuário.
Pós-condições: Usuário seguido com sucesso.
Sequência típica de eventos (Fluxo Principal): 1. [IN] O usuário acessa o perfil de um outro usuário. 2. [IN] O usuário pressiona o botão “seguir” presente na página de perfil do outro usuário. 3. [OUT] O sistema emite uma mensagem de confirmação. (1 exceção). 4. [OUT] O sistema emite uma mensagem de novo seguidor para o usuário que foi seguido.
Tratamento de Exceções e Variantes: Exceção 3a: Erro de execução do sistema. 3a.1 [OUT] O sistema informa que houve um erro na execução da ação. 3a.2 [OUT] Retorna ao passo 1.

Fonte: Autoria própria (2023).

O Quadro 9 apresenta a expansão do caso de uso para interagir com publicações.

Quadro 9 – Operação de interagir com publicações

Caso de Uso: Interagir com publicações. Descrição: Usuários do sistema podem escrever comentários, usar imagens, vídeos ou então curtir e compartilhar publicações de outros usuários. Evento Iniciador: Necessidade do usuário interagir com publicações cadastradas no sistema.
Atores: Usuário.
Precondições: O usuário deve possuir uma conta no sistema. Usuário autenticado no sistema.
Pós-condições: Interação com a postagem (curtir, comentar ou compartilhar) realizado com sucesso.
Sequência típica de eventos (Fluxo Principal): 1. [IN] O usuário acessa a opção para comentar, compartilhar ou curtir uma publicação. (1 exceção). 2. [OUT] O sistema emite uma mensagem de confirmação. 3. [OUT] O sistema emite uma mensagem de nova interação para o usuário dono da publicação.

Tratamento de Exceções e Variantes:**Exceção 1a: Usuário seleciona opção curtir ou compartilhar.**

1a.1 [OUT] O sistema emite pedido de confirmação de ação.

1a.2 [OUT] Retorna ao passo 2.

Exceção 1b: Usuário seleciona opção para comentar publicação.

1b.1 [OUT] O sistema abre formulário para o usuário escrever seu comentário.

1b.2 [IN] O usuário digita no formulário seu comentário.

1b.3 [OUT] Retorna passo 2.

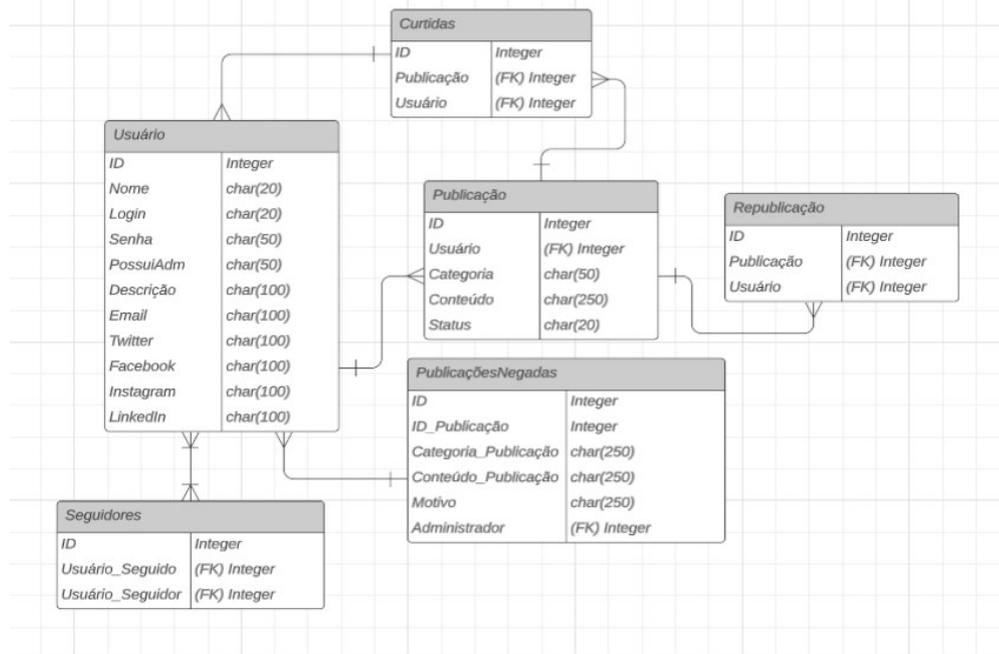
Exceção 1c: Usuário não digita informação no campo de comentário.

1c.1 [OUT] O sistema emite uma mensagem de erro de campo não preenchido.

1c.1 [OUT] Retorna passo 1.

Fonte: Autoria própria (2023).

A Figura 2 apresenta o diagrama de entidades e relacionamentos do banco de dados da aplicação.

Figura 2 – Diagrama de entidades e relacionamentos do banco de dados**Fonte: Autoria própria (2023).**

As tabelas apresentadas anteriormente possuem suas funções, sendo elas:

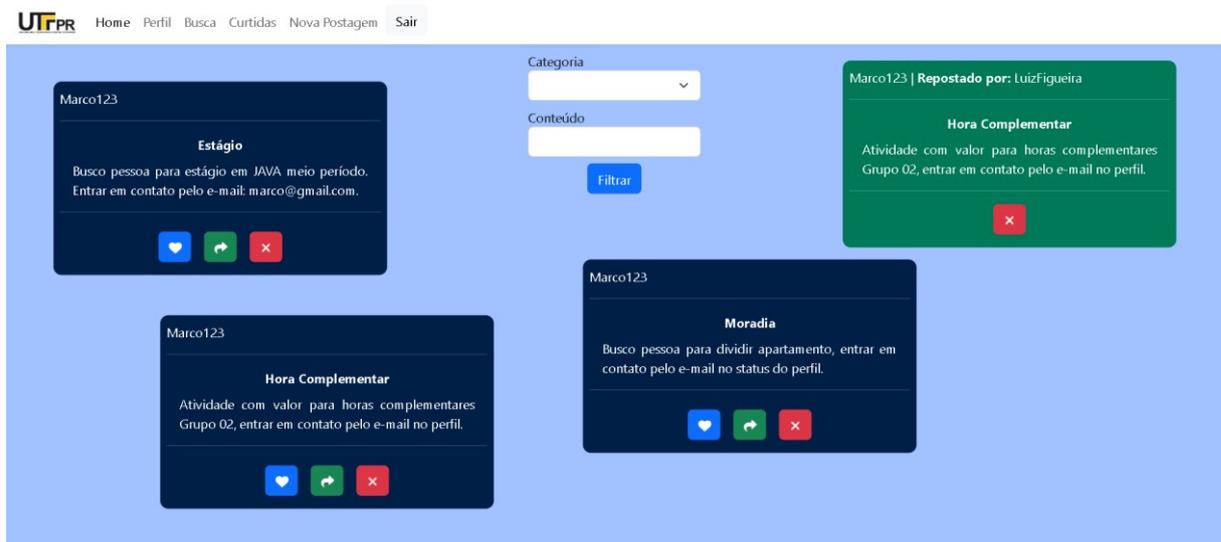
- **Usuário:** Tem a função de salvar as informações e função dos usuários que se cadastrarem no sistema.
- **Seguidores:** Será para salvar os seguidores de cada um dos usuários.
- **Publicação:** Servirá para salvar as publicações e identificar com o campo “status” se a publicação já foi avaliada por um administrador.

- **Curtidas:** Essa tabela salvará as curtidas da publicação, salvando o usuário que curtiu e qual publicação foi curtida.
- **Republicação:** Terá função parecida com a tabela “Curtidas” mas salvará as republicações da publicação.
- **Administrador:** Terá o intuito de armazenar as credenciais de administrador dos usuários do sistema, ressaltando que não serão todos os usuários que terão a função de administrador.
- **Publicações Negadas:** Persistirá publicações que não forem aprovadas pelos administradores. Nela será instanciado o administrador que não aprovou a publicação e o texto explicando o motivo de não ter aprovado.

4.3 Apresentação do Sistema

As telas a seguir apresentam as principais funcionalidades implementadas no sistema. A tela inicial, apresentada na Figura 3, é a de listagem de anúncio.

Figura 3 – Tela inicial do sistema



Fonte: Autoria própria (2023).

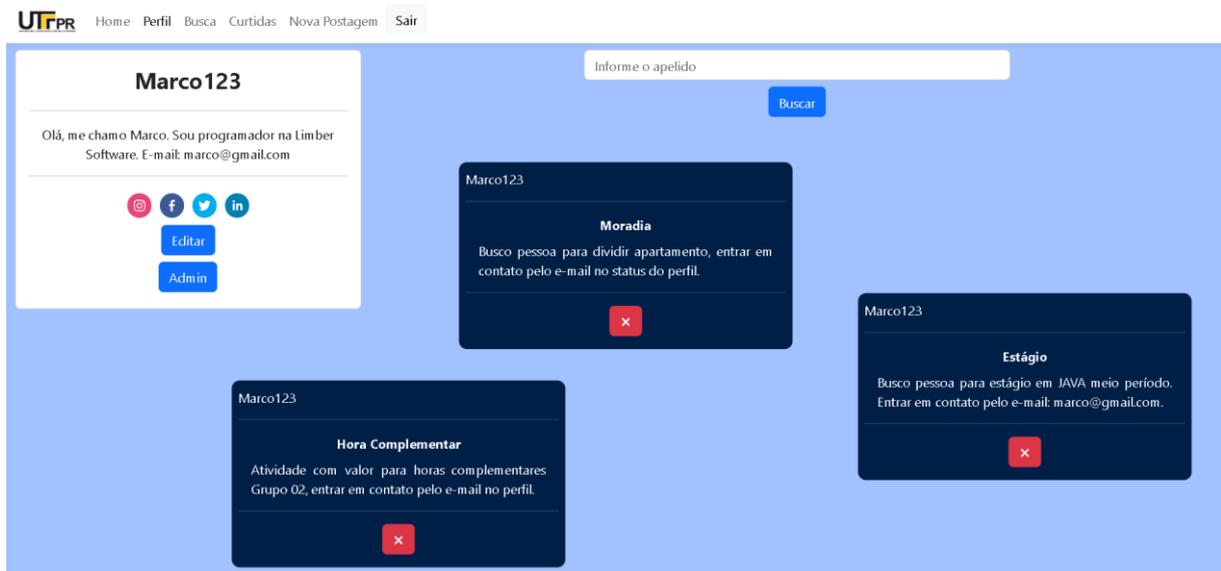
A Figura 3 mostra a tela inicial do sistema na qual é possível ver as seguintes informações:

- **Cartões de Publicações:** os cartões de publicações são compostos pelo usuário que fez a publicação na parte superior esquerda e a categoria e conteúdo da publicação ao centro.

- Cartões de Republicação: são similares ao cartão de publicação, apenas se diferenciam pela cor e por conterem também o nome do usuário que fez a republicação.
- Barra de navegação: estará presente em todas as telas para que o usuário possa utilizá-la para interagir com o sistema.
- Filtro de postagens: no centro da tela há dois campos de preenchimento (categoria, conteúdo) e logo abaixo um botão para que o usuário possa filtrar os cartões presentes na página inicial.

A Figura 4 mostra a tela de perfil de usuário que possui uma área para mostrar as publicações realizadas pelo usuário, sendo possível excluí-las. Na esquerda da página fica o nome do usuário, a descrição de perfil, os links de direcionamento para as outras redes e dependendo o nível de perfil de usuário pode haver um ou dois botões. Caso o usuário não possua permissão de administrador aparecerá apenas o botão de edição do perfil, caso o usuário tenha a permissão aparecerá o botão de administrador.

Figura 4 – Tela de perfil



Fonte: Autoria própria (2023).

A Figura 5 apresenta a tela de edição de perfil. Nessa tela, o usuário poderá alterar seu nome que é o que irá aparecer para os outros usuários, não o que é utilizado para realizar a autenticação no sistema. Além disso, ele poderá alterar os dados como a descrição de perfil e o *link* das suas outras redes.

Figura 5 – Tela de edição de perfil

UTPR Home Perfil Busca Curtidas Nova Postagem Sair

Editar Perfil

Apelido
Marco123

Instagram
https://www.instagram.co

Facebook
https://www.instagram.cc

Twitter
https://www.instagram.co

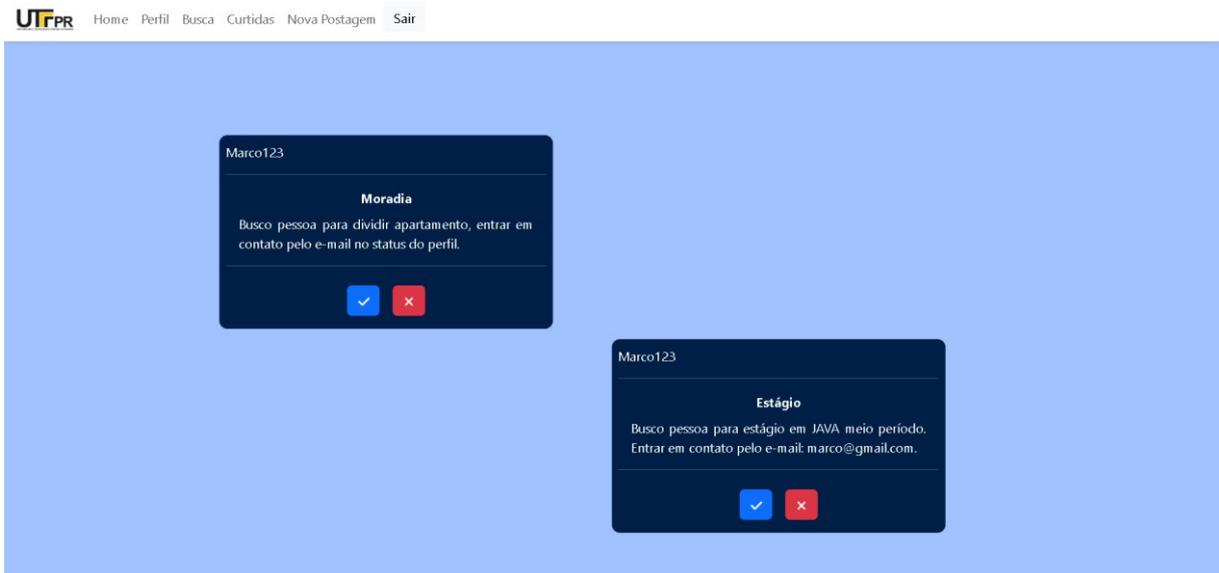
LinkedIn
https://www.instagram.cc

Descrição do perfil
Olá, me chamo Marco. Sou programador na Limber Software. E-mail: marco@gmail.com

Salvar

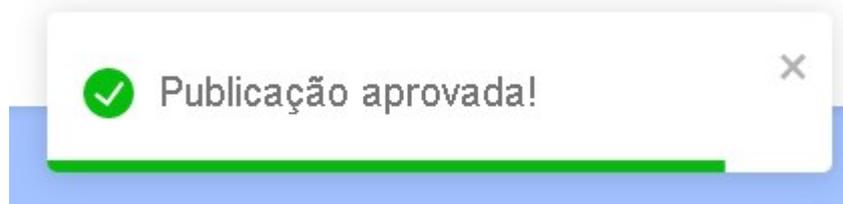
Fonte: Autoria própria (2023).

A tela de administrador do sistema, apresentada na Figura 6, irá mostrar uma lista das publicações a serem aprovadas. Para que as publicações apareçam para os outros usuários elas devem ser aprovadas pelo administrador. O botão azul possibilita que o administrador possa aprovar uma publicação. O botão vermelho é utilizado para reprovar uma publicação. Podem haver diversos motivos para se reprovar uma publicação, entre eles, a falta de informações, informações incongruentes, não incluir formas de contato e outros.

Figura 6 – Tela de administrador

Fonte: Autoria própria (2023).

Caso a publicação seja aprovada pelo administrador, aparecerá uma confirmação na tela, como mostra a Figura 7.

Figura 7 – Confirmação de aprovação de publicação

Fonte: Autoria própria (2023).

Caso o administrador reprove a publicação, ele será redirecionado para a tela da Figura 8.

Figura 8 – Tela de reprovação de publicação

Fonte: Autoria própria (2023).

Como mostra a Figura 8, a tela de reprovação de publicação possui um formulário, no qual mostra o conteúdo da publicação reprovada e um campo para preenchimento em que o administrador deve inserir de forma obrigatória o motivo pelo qual reprovou a publicação. Após o preenchimento deste campo, o administrador clicará no botão “Enviar” a publicação será cancelada e não aparecerá no sistema e o usuário responsável pela publicação será notificado por e-mail, conforme mostra a Figura 9.

Figura 9 – E-mail contendo o conteúdo de reprovação de publicação

Prezado usuário, sua postagem foi recusada por um administrador. A postagem em questão foi:

Categoria: Moradia

Conteúdo: Busco companheiro de quarto para morar junto e dividir contas do AP. Mais informações entrar em contato.

Razão da recusa: Por favor, insira uma forma de contato ou no conteúdo da mensagem ou em seu perfil, não encontramos em nenhum dos dois.

Fonte: Autoria própria (2023).

A tela de criação de publicação, apresentada na Figura 10, é utilizada quando o usuário deseja escrever uma publicação a qual deverá estar associada à uma categoria. Após o usuário realizar sua publicação ela será disponibilizada na tela do administrador (Figura 6) o qual analisará se o conteúdo está apto para ser publicado.

Figura 10 – Tela de criação de publicação

Fonte: Autoria própria (2023).

Para criar uma publicação é necessário informar a categoria a que a publicação pertence e o seu respectivo conteúdo, conforme os campos da Figura 9.

A Figura 11 mostra a tela referente a busca de publicações. Nesta tela é possível buscar as publicações com por categoria, pelo conteúdo da publicação ou por ambos. Nesse caso, a busca ocorrerá com base em todas as publicações cadastradas no sistema e não somente naquelas que o usuário segue.

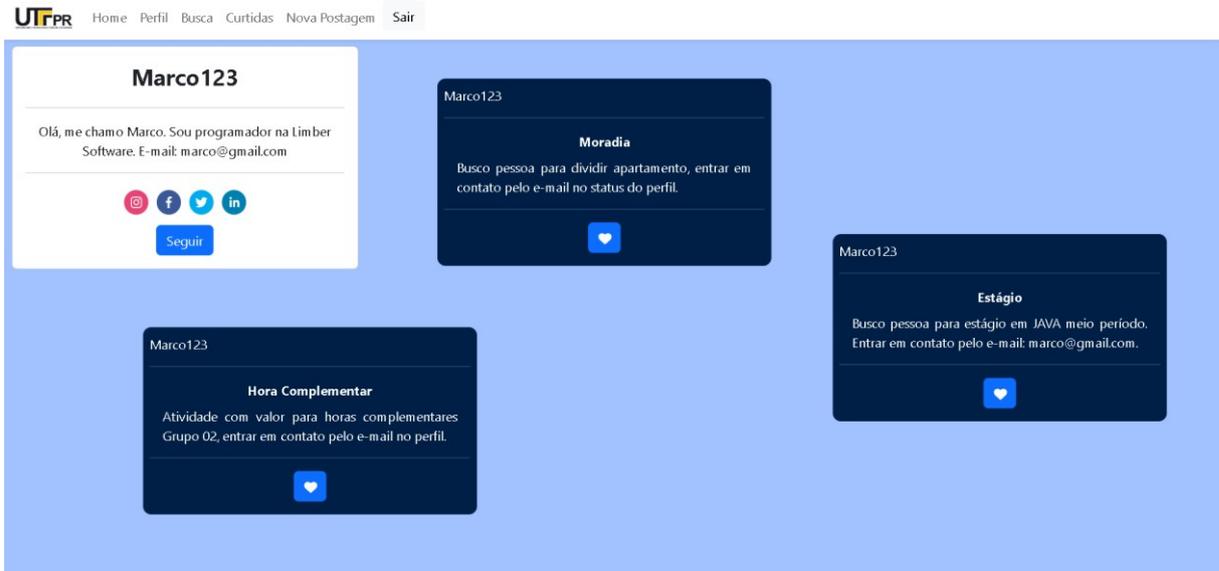
Figura 11 – Tela de busca por publicações

Fonte: Autoria própria (2023).

A Figura 12 mostra a tela de perfil de usuário externo com as publicações do usuário e um botão para poder segui-lo. Quando um usuário é seguido, as postagens

e reportagens desse usuário irão aparecer na tela inicial do sistema. A busca de usuário e ida ao perfil podem ocorrer de duas formas: ao clicar no nome do usuário a partir do nome presente na publicação ou pelo campo de pesquisa (Figura 4).

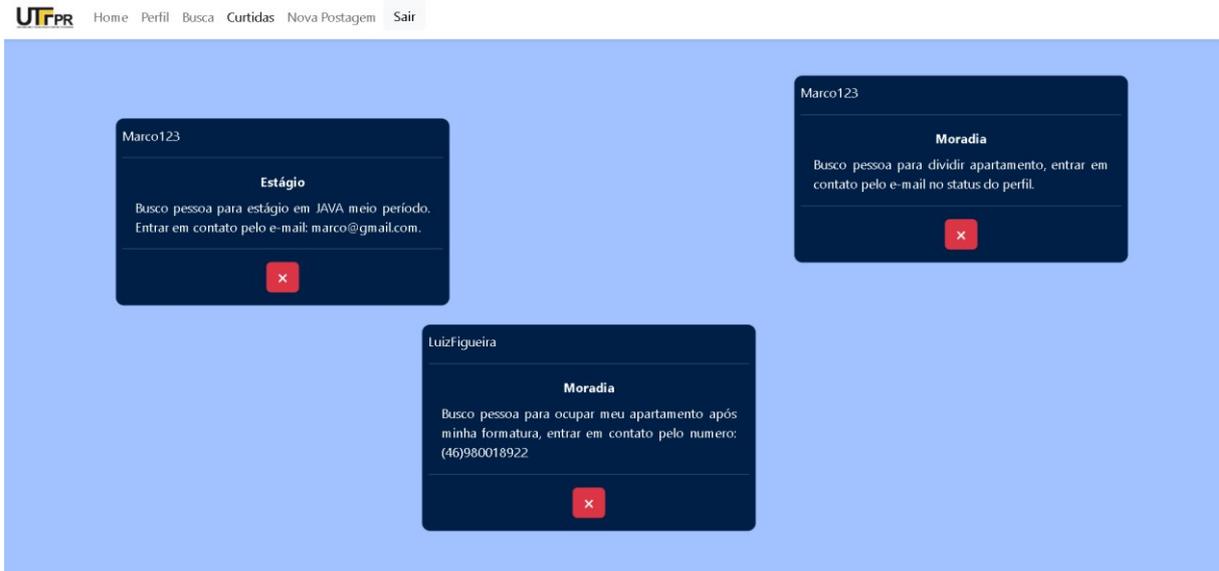
Figura 12 – Tela de perfil externo



Fonte: Autoria própria (2023).

A tela de publicações curtidas, apresentada na Figura 13, mostra as publicações curtidas e um botão dentro da publicação para que o usuário possa remover as publicações dessa lista, caso não deseje mais visualizar e ter essa publicação como curtida.

Figura 13 – Tela de publicações curtidas



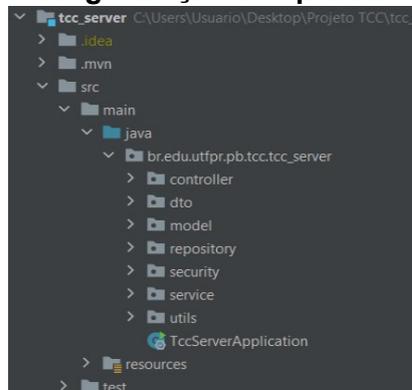
Fonte: Autoria própria (2023).

4.4 Implementação do Sistema

A apresentação da implementação do sistema é realizada por meio de partes relevantes do código desenvolvido, visando exemplificar o uso das tecnologias presentes no Quadro 1 do Capítulo 3.

Inicialmente foi construído o banco de dados, utilizando o PostgreSQL como Sistema de Gerenciamento de Banco de Dados (SGBD) e o DBeaver para a administração e desenvolvimento. No projeto *back-end*, Java foi utilizado como linguagem de programação, juntamente com o Spring. A Figura 14 representa a estrutura dos pacotes no projeto *back-end*.

Figura 14 – Organização dos pacotes *back-end*



Fonte: Autoria própria (2023).

A Listagem 1 apresenta o código-fonte da classe *SecurityConfig* que está na pasta *security*.

Listagem 1 – Configuração de permissão

```

@EnableWebSecurity
@Configuration
public class WebSecurity {
    private final AuthService authService;
    private final AuthenticationEntryPoint
authenticationEntryPoint;

    public WebSecurity(AuthService authService,
AuthenticationEntryPoint authenticationEntryPoint) {
        this.authService = authService;
        this.authenticationEntryPoint =
authenticationEntryPoint;
    }

    @Bean
    @SneakyThrows
    public SecurityFilterChain filterChain(HttpSecurity http) {
        AuthenticationManagerBuilder
authenticationManagerBuilder =
http.getSharedObject(AuthenticationManagerBuilder.class);

authenticationManagerBuilder.userDetailsService(authService)
        .passwordEncoder(passwordEncoder());
        AuthenticationManager authenticationManager =
authenticationManagerBuilder.build();

        http.csrf().disable()
            .exceptionHandling()
            .authenticationEntryPoint(authenticationEntryPoint)
                .and()
                .cors()
                .and()
                .authorizeRequests()
                .antMatchers(HttpMethod.POST,
"/users/**").permitAll()
            .antMatchers(HttpMethod.GET, "/error/**").permitAll()
                .anyRequest().authenticated()
                .and()
            .authenticationManager(authenticationManager)
                .addFilter(new
JWTAuthenticationFilter(authenticationManager, authService) )
                .addFilter(new
JWTAuthorizationFilter(authenticationManager, authService) )
            .sessionManagement().sessionCreationPolicy(SessionCreationPoli
cy.STATELESS);

        return http.build();
    }
}

```

```

    }
}

```

Fonte: Autoria própria (2023).

O código da Listagem 1 apresenta a configuração implementada no sistema. A classe `WebSecurity` é anotada com `@EnableWebSecurity` e `@Configuration`, pois ela é responsável por definir as medidas de segurança para o aplicativo. A classe `WebSecurity` possui duas dependências injetadas por meio do construtor: `AuthService` e `AuthenticationEntryPoint`. Essas dependências servem para fornecer serviços de autenticação e tratamento de exceções relacionadas à autenticação.

O método `filterChain` é anotado com `@Bean` e retorna um objeto `SecurityFilterChain`, responsável por configurar a cadeia de filtros de segurança. Outras configurações que estão presentes incluem a habilitação do suporte ao CORS com `http.cors()`, e a configuração das regras de autorização por meio de `authorizeRequests()`. Nesse trecho, são especificadas as permissões para os caminhos `/users/**` (permitindo métodos POST) e `/error/**` (permitindo métodos GET) sem autenticação, enquanto qualquer outra solicitação requer autenticação.

A Listagem 1 mostra de forma geral como ocorreu a implementação da camada de segurança utilizando recursos do Spring Security.

Listagem 2 – Controller das publicações

```

@RestController
@RequestMapping("publications")
public class PublicationController {

    private final PublicationsService publicationsService;
    private ModelMapper modelMapper;

    public PublicationController(PublicationsService
publicationsService, ModelMapper modelMapper) {
        this.publicationsService = publicationsService;
        this.modelMapper = modelMapper;
    }

    @PostMapping
    @ResponseStatus(HttpStatus.CREATED)
    public GenericResponse createPublications(@RequestBody
@Valid Publications publications) {
        publicationsService.save(publications);
        return new GenericResponse("Publicação inserida com
sucesso");
    }
}

```

```

    }

    @DeleteMapping("{id}")
    public GenericResponse delete(@PathVariable Long id) {
        publicationsService.delete(id);
        return new GenericResponse("Publicação deletada com
sucesso");
    }

    @GetMapping
    public ResponseEntity<List<PublicationsDto>> findAll() {
        return ResponseEntity.ok(
            publicationsService.findAllByUser()
                .stream()
                .map(this::convertEntityToDto)
                .collect(Collectors.toList())
        );
    }
}

```

Fonte: Aatoria própria (2023).

A Listagem 2, apresenta o código do *controller* para as requisições relacionadas à publicação. A anotação `@RestController` indica que é uma classe *Application Programming Interface* (API) *Representational State Transfer* (REST), já a anotação `@RequestMapping` define o caminho da requisição.

As anotações `@PostMapping`, `@DeleteMapping` e `@GetMapping` estão indicando o verbo HTTP da requisição, ou seja, se enviada uma requisição *post* para o caminho da aplicação `/publications` o método `createUser()` será acessado.

O método `createPublications()`, recebe como *payload* no corpo da solicitação uma publicação. Essa publicação é validada com base nas anotações de validação `@Valid` aplicadas à classe. A resposta retorna um objeto `GenericResponse` com uma mensagem indicando o sucesso da ação.

O método `delete()`, recebe como parâmetro de caminho `{id}`, uma publicação é excluída com base no ID fornecido. Isso é feito chamando o método `delete` do serviço `publicationsService`.

No método `findAll()` todas as publicações associadas a um usuário são buscadas usando o serviço `publicationsService`. O resultado é convertido em uma lista de objetos `PublicationsDto` usando o *modelMapper*, que mapeia a entidade `Publications` para o DTO correspondente. Em seguida, a lista é encapsulada em uma resposta do tipo `ResponseEntity`, retornando um código de status HTTP 200 (OK) e a lista de DTOs de publicações.

A Listagem 3 representa a interface que contém os métodos comuns para os *services*, sendo eles quatro métodos principais: *findOne*, *findAll*, *save* e *delete*. O método *findOne* retorna uma entidade com base em um ID. O método *findAll* retorna uma lista de entidades. O método *save* salva uma entidade e retorna a entidade salva. O método *delete* exclui uma entidade com base em um ID.

Listagem 3 – Interface contendo métodos padrões do *service*

```
public interface CrudService <T, ID extends Serializable> {
    T findOne(ID id);

    List<T> findAll();

    T save(T entity);

    void delete(ID id);
}
```

Fonte: Autoria própria (2023).

Na Listagem 4 pode-se observar a implementação dessa interface por parte de uma das classes. A anotação `@Service` permite que o Spring detecte essa classe e a gerencie como um componente dentro da aplicação, o que significa que ela pode ser injetada em outras classes quando necessário.

Todas as classes *service* possuem um *serviceImpl*, as classes *Impl* estendem as classes de *service* para implementarem as classes abstratas. O construtor da classe recebe um parâmetro *Repository*, o qual é uma dependência da classe, que se refere, neste caso, ao *repository* do usuário.

Listagem 4 – Classe implementando os métodos padrões

```
@Override
public User save(User user) {
    user.setPassword(passwordEncoder.encode(user.getPassword()));
    return userRepository.save(user);
}

@Override
public User findOne(Long id) {
    return userRepository.findById(id).orElse(null);
}

@Override
public List<User> findAll() {
    return userRepository.findAll();
}
```

```

@Override
public void delete(Long id) {
    userRepository.deleteById(id);
}

```

Fonte: A autoria própria (2023).

A Listagem 5 contém o código-fonte da classe *repository* presente no sistema. Cada método utiliza diferentes critérios de filtro para buscar publicações específicas no banco de dados. Por exemplo, *findAllByUserId* retorna todas as publicações associadas a um determinado ID de usuário. *findAllByUserIdInAndStatus* retorna todas as publicações associadas a uma lista de IDs de usuário e com um determinado *status*. *findAllByStatusAndContentContaining* retorna todas as publicações com um determinado status e que contenham um determinado conteúdo.

Esses métodos são úteis para realizar buscas personalizadas e filtrar publicações com base em diferentes critérios, permitindo a obtenção de resultados mais específicos e relevantes de acordo com diferentes necessidades.

Listagem 5 – Repository de publicação

```

List<Publications> findAllByUserId(Long id);

List<Publications> findAllByStatus(String status);

List<Publications> findAllByStatusAndUserId(String status, Long id);

List<Publications> findAllByCategoryAndStatus(String category, String status);

List<Publications> findAllByUserIdInAndStatus(List<Long> ids, String status);

List<Publications> findAllByStatusAndContentContaining(String status, String content);

List<Publications>
findAllByStatusAndCategoryAndContentContaining(String status, String category, String content);

```

Fonte: A autoria própria (2023).

A classe *Repository* encapsula a lógica de acesso e persistência de dados, fornecendo métodos para executar operações de banco de dados. A interface *PublicationsRepository* estende a interface *JpaRepository*, que faz parte do Spring

Data JPA, essa que fornece todos os métodos padrões para operações Create, Read, Update, Delete (CRUD) e permite montar consultas usando *Query Methods*, como pode ser visto em diversos métodos na interface da Listagem 5.

Esses métodos feitos a partir do *Query Methods* foram utilizados em diversas partes do projeto, um exemplo pode ser visto na Listagem 6, onde é mostrada a busca de publicações.

Listagem 6 – Busca de publicações

```

@Override
@Transactional
public List<Publications> findAllByUserIdIn() {
    Object principal =
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
    User user =
userService.findByUsername(principal.toString());
    List<Long> ids =
getIdFollowers(followRepository.findAllByFollower(user));

    return
publicationsRepository.findAllByUserIdInAndStatus(ids,
"validPost");
}

private List<Long> getIdFollowers(List<Follow> follows) {
    List<Long> idFollowers = new ArrayList<>();
    for (Follow follow : follows) {
        idFollowers.add(follow.getFollowed().getId());
    }
    return idFollowers;
}

```

Fonte: Autoria própria (2023).

A Listagem 6 utiliza o método *findAllByUserIdIn()* para buscar as postagens dos usuários com base no *status* de *follow*. Dessa forma, o sistema busca apenas as publicações de usuários que o usuário autenticado seguiu.

A anotação *@Transactional* é uma anotação do Spring Framework que indica que todas as operações de banco de dados realizadas dentro do método devem ser executadas em uma única transação. Isso garante que, se ocorrer uma exceção durante a execução do método, todas as operações de banco de dados serão revertidas para manter a integridade dos dados.

A função `SecurityContextHolder.getContext()` serve para a partir do `Token` do usuário, o sistema consegue recuperar o nome do usuário, e usando o `findByUsername()` do `repository` de `users`, é possível recuperar as informações do usuário sem ser necessário ele inserir elas.

Após encontrar o usuário é utilizado o método `findAllByFollower()`, passando como parâmetro o usuário encontrado. Esse método retorna uma lista de quem o usuário está seguindo. Como o retorno desta consulta é um `array` de informações, foi implementado o método `getIdsFollowers()` para retirar e formar apenas uma lista de ID 's. Após obter os ID 's, foi utilizado o método `findAllByUserIdInAndStatus()` para achar as postagens com base nos ID 's presentes no `array` e no `status` da publicação sendo `"validPost"`.

O trecho de código da Listagem 7 apresenta o método que faz a inserção dos `likes` nas publicações. Ao receber uma `Publications`, o método cria uma nova instância de `Likes`. Em seguida, ele obtém o usuário autenticado a partir da função `SecurityContextHolder.getContext().getAuthentication().getPrincipal()`. A partir disso, é buscado o usuário correspondente usando o método `userService.findByUsername`. O usuário encontrado é associado ao objeto `likes` assim como a publicação que foi recebida. Por fim, o método retorna o objeto que foi salvo.

Listagem 7 – Método para salvar curtida

```
@Override
public Likes save(Publications publications) {
    Likes likes = new Likes();

    Object principal = SecurityContextHolder.getContext()
        .getAuthentication().getPrincipal();

    User user = userService.findByUsername(principal
        .toString());

    likes.setUser(user);
    likes.setPublication(publications);

    return likesRepository.save(likes);
}
```

Fonte: Autoria própria (2023).

O método da Listagem 8 é responsável pelo processo de cancelamento de uma publicação reprovada. Primeiro, é obtido o usuário autenticado por meio do

SecurityContextHolder e do método de busca do *userService*. Em seguida, é recuperada a publicação relacionada com base no ID fornecido na instância de *DeniedPublications* e o usuário associado à publicação.

Uma mensagem de e-mail, em formato HTML, é criada usando o *emailSender* para enviar uma notificação ao usuário da publicação cancelada. O conteúdo do e-mail é definido no *htmlMsg* e as informações necessárias, como remetente, destinatário e assunto, são configuradas no objeto *helper*. O e-mail é enviado por meio do *emailSender.send()*. Em seguida, a publicação é excluída usando o serviço *publicationsService.delete()* com base no ID. Por fim, a instância de *DeniedPublications* é salva e retornada como resultado do método.

Listagem 8 – Método envio de e-mail para publicações reprovadas

```
@Override
public DeniedPublications save(DeniedPublications
deniedPublications) {

    Object principal =
SecurityContextHolder.getContext().getAuthentication()
.getPrincipal();

    User user =
userService.findByUsername(principal.toString());

    deniedPublications.setResponsibleAdmin(user);

    Publications publication = publicationsService
.findOne(deniedPublications.getPublicationId());

    User userPublication = publication.getUser();

    MimeMessage message = emailSender.createMimeMessage();
    MimeMessageHelper helper = new MimeMessageHelper(message,
"utf-8");

    String htmlMsg = "...";

    try {
        helper.setText(htmlMsg, true);
        helper.setFrom("marco@gmail.com");
        helper.setTo(userPublication.getEmail());
        helper.setSubject("Cancelamento Postagem");
    } catch (MessagingException e) {
        e.printStackTrace();
    }
}
```

```

    emailSender.send(message);

    publicationsService.delete(deniedPublications
        .getPublicationId());

    return
    deniedPublicationsRepository.save(deniedPublications);
}

```

Fonte: Autoria própria (2023).

A Figura 15 mostra a utilização do *useEffect* em um componente React. Ele é utilizado para realizar ações quando o componente em questão for renderizado pela primeira vez. No caso da Figura 15, o *useEffect* executa a função *checkToken()*, *findFollowPosts()* e *findFollowReposts()* logo em seguida que o componente é montado. Como foi adicionado os colchetes ao final do método, essas ações ocorrerão apenas uma vez, caso contrário, sempre que o componente estiver em uso, as ações dentro do método do *useEffect* ocorrem.

Figura 15 – Métodos dentro do *useEffect*

```

useEffect(() => {
    checkToken();
    findFollowPosts();
    findFollowReposts();
}, []);

```

Fonte: Autoria própria (2023).

O método *checkToken()* da Listagem 9 é responsável por verificar a validade do *Token* de autenticação. Inicialmente, a função chama o método *checkToken()* do serviço *AuthService* para fazer uma solicitação assíncrona para verificar o *Token*. Caso a solicitação seja bem-sucedida, a resposta é mostrada no *console* e a variável *apiError* é definida como falsa, indicando que não houve erro. Se ocorrer um erro durante a verificação, o serviço *AuthService* é chamado para realizar o *logout* do usuário, invalidando o *Token* atual. Em seguida, a função *isAuthenticated* é chamada para verificar se o usuário está autenticado. A página é recarregada utilizando a função *window.location.reload()* para realizar o *logout* e redirecionar o usuário para a página de *login*.

Listagem 9 – Método verificação de *Token*

```
const checkToken = () => {
  AuthService.checkToken()
    .then((response) => {
      console.log(response);
      setApiError(false);
    })
    .catch((errorResponse) => {
      AuthService.logout();
      AuthService.isAuthenticated();
      window.location.reload();
      setApiError(true);
      console.log(errorResponse);
    });
}
```

Fonte: Autoria própria (2023).

O código da Listagem 10 é responsável por buscar e recuperar os *posts* de pessoas que estão sendo seguidas. Primeiro, a função chama o método *findFollowPosts* do serviço *PostService* para fazer uma solicitação assíncrona e obter as publicações. Se a resposta da solicitação não tiver nenhum dado, ou seja, o objeto *response.data* não tiver chaves, a variável *noContent* é definida como verdadeira, indicando que não há conteúdo disponível. Caso contrário, a variável *noContent* é definida como falsa. Em seguida, os dados da resposta são atribuídos às variáveis *data* e *originalData*. Essas variáveis são utilizadas para armazenar os posts retornados, permitindo manipulações e filtros adicionais, se necessário. No caso de um erro a variável *apiError* é definida como verdadeira e o erro é registrado no *console*.

Listagem 10 – Método de busca de publicações

```
const findFollowPosts = () => {
  PostService.findFollowPosts()
    .then((response) => {
      if (Object.keys(response.data).length == 0) {
        setNoContent(true);
      } else {
        setNoContent(false);
      }
      setData(response.data);
    });
}
```

```

        setOriginalData(response.data);
        setApiError(false);
    })
    .catch((responseError) => {
        setApiError(true);
        console.log(responseError);
    });
}

```

Fonte: A autoria própria (2023).

A Listagem 11 mostra como foi feita a renderização dinâmica da lista de publicações. Utilizando a função *map*, cada objeto dentro do *array data* é mapeado para um componente individual.

Cada publicação é envolvida em um componente chamado *Draggable*, o que possibilita que ele seja arrastado pela tela. O posicionamento inicial é definido por meio das coordenadas x e y geradas aleatoriamente. O conteúdo da publicação é renderizado em uma *tag* `<div>` com estilo personalizado, exibindo o nome do usuário, a categoria do post e o próprio conteúdo. Foi feita uma divisória horizontal por meio da *tag* `<hr />` para separar as informações.

Além disso, na interface são exibidos três botões: botão de curtir, um botão de compartilhar e um botão de remover. Cada botão possui um ícone, um estilo personalizado e funções específicas.

Listagem 11 – Estruturação do cartão do sistema

```

{data.map((post: IPostCard) => (
<Draggable key={post.id} defaultPosition={{ x: randomX(), y: randomY() }}>
<div style={{ position: "absolute" }}>
  <div className='p-2' style={{(styles here...)}}>

    <div className="d-flex justify-content-start">
      <Link style={{ color: "white", textDecoration: "none" }}
to={` /exteriorProfilePost/${post.user!.id}`}>
        {post.user!.nickName}
      </Link>
    </div>
  </div>

<hr />

```

```

<div className='pb-2'>
  <b> {post.category} </b>
</div>

<div className='ps-3 pe-3' style={{ textAlign: "justify" }}>
  {post.content}
</div>
<hr />

  <button
    type="button"
    onClick={() => onClickLikePost(post)}
    style={{ width: 40 }}
    className="m-2 btn btn-primary">
<FontAwesomeIcon style={{ color: "white" }} icon={faHeart} />
  </button>

  <button
    type="button"
    onClick={() => onClickRepost(post)}
    style={{ width: 40 }}
    className="m-2 btn btn-success">
<FontAwesomeIcon style={{ color: "white" }} icon={faShare} />
  </button>

  <button
    type="button"
    onClick={() => onClickRemovePost(post.id!)}
    style={{ width: 40 }}
    className="m-2 btn btn-danger">
<FontAwesomeIcon style={{ color: "white" }} icon={faXmark} />
  </button>
</div>
</div>
</Draggable>
))}

```

Fonte: Autoria própria (2023).

A decisão de não implementar um sistema de comentários nas publicações foi tomada levando em consideração a preocupação de que os usuários podem entrar no sistema por um longo período de tempo. Dessa forma, isso pode resultar em

frustração para os usuários que deixaram comentários na publicação, além de representar uma possível perda de oportunidades apresentadas na publicação. Diante dessa preocupação, optou-se por deixar que cada usuário escolhesse o meio de comunicação pelo qual deseja ser contatado por aqueles interessados em suas publicações. Essa flexibilidade oferece aos usuários a liberdade de escolher o método de comunicação mais adequado à suas preferências, podendo ser por meio de e-mail, telefone celular, outras plataformas de *chat online*, entre outras opções disponíveis.

A decisão de não implementar a inserção de vídeos e imagens nas publicações foi baseada na consideração de que o armazenamento dessas informações poderia resultar em problemas e limitações no armazenamento e na gestão dessas informações. Vídeos e imagens são dados muito volumosos, inserir esses arquivos diretamente no banco de dados tradicional poderia levar a um aumento significativo no tamanho do banco, tornando as manipulações e interações com o banco mais complexas e demoradas.

A abordagem mais adequada a ser empregada seria utilizar armazenamento de arquivos com o Docker, que oferece uma maneira eficiente e escalável de armazenar e gerenciar vídeos e imagens. Por conta disso, foi adicionado como uma possibilidade de trabalho futuro a implementação do Docker no projeto.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento, incluindo a modelagem e a implementação de um sistema web com inspirações em funcionalidades de redes sociais para efetuar a oferta e busca de horas complementares, vagas de moradia, estágio e outras oportunidades. Embora o sistema não possua todas as funcionalidades de uma rede social, ele reúne diversos aspectos delas tornando sua utilização semelhante, facilitando o aprendizado e uso por parte de seus usuários.

A caracterização do sistema com elementos nas redes sociais levou a elaboração do referencial teórico baseado em redes sociais atuais. Foi apresentado o impacto das redes sociais na sociedade atual, e dados motivos para que o sistema desenvolvido fizesse uso de algumas características dessas redes. A partir dessa base teórica, foi possível criar um sistema que atende às necessidades dos usuários, proporcionando uma experiência semelhante às redes sociais já existentes.

A base do desenvolvimento do sistema se deu em dispor um site que auxiliasse os estudantes universitários a encontrarem e divulgarem informações sobre horas complementares, eventos e vagas de moradia, estágio ou emprego. As redes sociais se juntam a ideia do projeto pois, com o uso de elementos das redes sociais, a utilização do sistema se tornaria algo mais instintivo para os universitários, pelo fato da maior parcela deles serem jovens e também por boa parte deles já fazerem uso de, ao menos, uma rede social.

Para desenvolver as telas do sistema foi utilizado o *framework* React. Esse *framework* permite o desenvolvimento de *Single-Page Application* (SPA) de forma simples e com alto reaproveitamento de código. Outra facilidade proporcionada pela ferramenta é o mapeamento das informações recebidas. Apenas utilizando a função de *.map* faz com que seja simples a recuperação das informações e a passagem dessas informações para componentes que serão exibidos em telas. Destaca-se, ainda, a grande quantidade de bibliotecas presentes para o React, dessa forma facilitando em diversas partes do desenvolvimento, como na realização das requisições, realizando apenas a instalação dos pacotes do *react-router-dom* faz com que seja necessário apenas uma linha de código para se realizar as requisições para o *back-end*.

Java foi utilizada como linguagem de programação no *back-end*, juntamente com o ecossistema Spring. A utilização do Spring agilizou o desenvolvimento devido

aos diversos recursos presentes nele como, por exemplo, o gerenciamento de dependências que ocorre por meio da injeção de dependência. O Spring Data JPA proporcionou a implementação de grande parte do código necessário para a camada de manipulação de dados. Recursos como, por exemplo, autenticação, autorização baseada na permissão do usuário e armazenamento da senha criptografada foram realizados fazendo uso do Spring Security na aplicação.

Conexão com outras redes sociais, sistema de algoritmo de sugestões de postagens, sistema de envio de mensagem entre usuários por meio da aplicação, sistema de compartilhamento de postagem para outras redes, criação de sistema de exclusão de postagem após tempo determinado, desenvolvimento de versão mobile do projeto, implementação do Docker no projeto são indicações de trabalhos futuros para complementar o que foi desenvolvido e apresentado neste trabalho.

REFERÊNCIAS

BERGER, Katharina et al. **A Review of Information Systems Research on Online Social Networks**. Disponível em: <https://aisel.aisnet.org/cais/vol35/iss1/8/>. Acesso em: 16 set. 2022.

CARVALHO, Vanessa O; MURBACK, Fábio G. R. **ESTUDO DA UTILIZAÇÃO DAS REDES SOCIAIS DIGITAIS NAS EMPRESAS BRASILEIRAS**. Disponível em: https://www.pucpcaldas.br/graduacao/administracao/revista/artigos/v2014/Artigo41_2014.pdf. Acesso em: 17 set. 2022.

DEVMEDIA, 2007. VINICIUS, Introdução ao Desenvolvimento Ágil. Disponível em: <https://www.devmedia.com.br/introducao-ao-modelo-cascata/29843>. Acesso em: 01 mai. 2023.

DEVMEDIA, 2011. Ciclos de Vida do Software. Disponível em: <https://www.devmedia.com.br/ciclos-de-vida-do-software/21099>. Acesso em: 07 set. 2022.

DEVMEDIA, 2013. HIGOR, Introdução ao Modelo Cascata. Disponível em: <https://www.devmedia.com.br/introducao-ao-modelo-cascata/29843>. Acesso em: 07 set. 2022.

GROSSI, Márcia G. R. et al. **A utilização das tecnologias digitais de informação e comunicação nas redes sociais pelos universitários**. Disponível em: <https://periodicos.ufsc.br/index.php/textodigital/article/view/1807-9288.2014v10n1p4>. Acesso em: 15 set. 2022.

IBGE, 2017. Acesso a Internet e a televisão e posse de telefone móvel celular para uso pessoal 2017. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/livros/liv101631_informativo.pdf. Acesso em: 12 set. 2022.

LORENA, Suélem Barros de et al. **Análise do Acesso à Informação Acadêmica entre Estudantes de Medicina Inseridos numa Metodologia Ativa de Aprendizagem**. Disponível em: <https://www.scielo.br/j/rbem/a/jYbDrvc6McwhMqrphT5g7f/abstract/?lang=pt#>. Acesso em: 16 set. 2022.

MESE, Can. **The use of social networks among university students**. Disponível em: https://www.researchgate.net/publication/332462369_The_use_of_social_networks_among_university_students. Acesso em: 18 set. 2022.

ROCHA, Adair José dos Santos. **COMO O CELULAR INTERFERE NO TEMPO DE ESTUDO DOS ESTUDANTES UNIVERSITÁRIOS?**. Disponível em: <https://e-revista.unioeste.br/index.php/ideacao/article/view/24126>. Acesso em: 16 set. 2022.

SANTOS, Rodrigo O. dos. **Algoritmos, engajamento, redes sociais e educação**. Disponível em:

<https://periodicos.uem.br/ojs/index.php/ActaSciEduc/article/view/52736>. Acesso em: 15 set. 2022.

SELVARAJAH, Ushamalar. **The intention of using Facebook by postgraduate students for knowledge sharing: an empirical study**. Disponível em: <https://www.inderscienceonline.com/doi/10.1504/IJMIE.2021.111812>. Acesso em: 18 set. 2022.

SURGIU, 2020. Cresce o número de jovens que estudam e trabalham.. Disponível em: <https://surgiu.com.br/2020/08/26/cresce-numero-de-jovens-que-estudam-e-trabalham/>. Acesso em: 23 ago. 2022.

TAYEBI, Seyed Mohammad. **The role of information systems in communication through social media**. Disponível em: https://www.researchgate.net/publication/330991644_The_role_of_information_systems_in_communication_through_social_media. Acesso em: 17 set. 2022.

TRIBUNAONLINE, 2020. Recorde de jovens que estudam e trabalham. Disponível em: <https://tribunaonline.com.br/economia/recorde-de-jovens-que-estudam-e-trabalham-62721>. Acesso em: 21 set. 2022.

WE ARE SOCIAL; HOOTSUITE. **Digital 2019: Brazil. All the data and trends you need to understand, Internet, social media, mobile, and e-commerce behaviours in 2019**. Disponível em: <https://datareportal.com/reports/digital-2019-brazil>. Acesso em: 10 set. 2022.