

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**MURAL WEB: UM APLICATIVO MÓVEL QUE DISPONIBILIZA
INFORMAÇÕES ACADÊMICAS DE INSTITUIÇÕES**

RICHARD WESLEY BARBOSA FREITAS

**Cajazeiras
2025**

RICHARD WESLEY BARBOSA FREITAS

**MURAL WEB: UM APLICATIVO MÓVEL QUE DISPONIBILIZA INFORMAÇÕES
ACADÊMICAS DE INSTITUIÇÕES**

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Francisco Paulo de Freitas Neto.

**Cajazeiras
2025**

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

F866m	<p>Freitas, Richard Wesley Barbosa. Mural web : um aplicativo móvel que disponibiliza informações acadêmicas de instituições / Richard Wesley Barbosa Freitas. – Cajazeiras-PB, 2025. 60f. : il.</p> <p>Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2025.</p> <p>Orientador(a): Prof. Me. Francisco Paulo de Freitas Neto.</p> <p>1. Desenvolvimento de sistemas. 2. Aplicativo Mural web. 3. Mural de informações acadêmicas. 4. IFPB Cajazeiras. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.</p>
IFPB/CZ	CDU: 004.4(043.2)



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

RICHARD WESLEY BARBOSA FREITAS

**MURAL WEB: UM APLICATIVO MÓVEL QUE DISPONIBILIZA
INFORMAÇÕES ACADÊMICAS DE INSTITUIÇÕES**

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Francisco Paulo de Freitas Neto

Aprovada em: **06 de Março de 2025.**

Prof. Me. Francisco Paulo de Freitas Neto - Orientador

Prof. Me. Michel da Silva - Avaliador

IFPB - Campus Cajazeiras

Profa. Dra. Eva Maria Campos Pereira

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Francisco Paulo de Freitas Neto**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 06/03/2025 15:31:23.
- **Michel da Silva**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 07/03/2025 07:39:37.
- **Eva Maria Campos Pereira**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 11/03/2025 19:26:43.

Este documento foi emitido pelo SUAP em 06/03/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 677435

Verificador: e663e42f45

Código de Autenticação:



Rua José Antônio da Silva, 300, Jardim Oásis, CAJAZEIRAS / PB, CEP 58.900-000
<http://ifpb.edu.br> - (83) 3532-4100

RESUMO

Este trabalho de conclusão de curso apresenta o desenvolvimento de uma aplicação denominada de Mural Web, que tem como objetivo melhorar a experiência dos usuários na obtenção de informações acadêmicas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), campus Cajazeiras. O aplicativo disponibiliza informações acadêmicas em forma de postagens de maneira mais eficiente e interativa para solucionar problemas de comunicação e usabilidade identificados nos meios atuais de comunicação. O trabalho descreve os objetivos gerais e específicos do projeto, bem como a análise e o levantamento de requisitos em forma de histórias de usuário. Além disso, é destacado o conjunto de tecnologias, arquiteturas, práticas, estruturas e padrões que foram seguidos ou utilizados durante o processo de desenvolvimento. O Mural Web é uma contribuição para a área de tecnologia e pode ser utilizado como referência para o desenvolvimento de aplicações similares.

Palavras-chave: Postagens. Informações acadêmicas. IFPB. Aplicação. Desenvolvimento. Interatividade. Usabilidade.

ABSTRACT

This course completion work presents the development of an application project called Mural Web, which aims to improve users' experience in obtaining academic information from the Federal Institute of Education, Science and Technology of Paraiba (IFPB), campus Cajazeiras. The application provides academic information in the form of posts in a more efficient and interactive way to solve communication and usability problems identified in current means of communication. The work describes the general and specific objectives of the project, as well as the analysis and gathering of requirements in the form of user stories. Furthermore, the set of technologies, architectures, practices, structures, and standards that will be followed or used during the development process is highlighted. Mural Web is a contribution to the technology area and can be used as a reference for the development of similar applications.

Keywords: Posts. Academic information. IFPB. Application. Development. Interactivity. Usability.

LISTA DE FIGURAS

Figura 1 – Exemplo de formulário estruturado de duas maneiras	13
Figura 2 – Exemplo de documento que representa um objeto <i>BSON</i> armazenado no <i>MongoDB</i>	20
Figura 3 – Tela inicial da aplicação	34
Figura 4 – Diagrama da Arquitetura Modular Baseada em API com Serviços Distribuídos	35
Figura 5 – Captura da tela inicial do aplicativo móvel exibindo as postagens separadas opor categorias e o menu de opções	38
Figura 6 – Captura da tela de busca do aplicativo móvel exibindo os resultados da busca e as opções de filtragem	39
Figura 7 – Captura da tela de exibição de uma postagem no aplicativo móvel .	40
Figura 8 – Captura da página de login da plataforma web	41
Figura 9 – Captura de tela da seção de postagens do painel administrativo da plataforma web	42
Figura 10 – Captura de tela da seção de categorias do painel administrativo da plataforma web	42
Figura 11 – Captura de tela da seção de usuários do painel administrativo da plataforma web	43
Figura 12 – Captura de tela do formulário de criação de postagens do painel administrativo da plataforma web	44
Figura 13 – Captura de tela do formulário de edição de postagens do painel administrativo da plataforma web	44
Figura 14 – Captura da página inicial da plataforma web	56
Figura 15 – Captura da exibição de uma postagem na plataforma web	57
Figura 16 – Captura do uso da barra de busca da plataforma web	57
Figura 17 – Captura da página de edição do perfil do usuário na plataforma web	58
Figura 18 – Captura da página de recuperação de conta do usuário na plataforma web.	58
Figura 19 – Captura de tela da área de notificações de um dispositivo móvel que exibe uma notificação de nova postagem do Mural Web.	59

LISTA DE QUADROS

Quadro 1 – User story da criação de conta do usuário	50
Quadro 2 – User story de acesso a conta do usuário	50
Quadro 3 – User story de recuperação de conta do usuário	51
Quadro 4 – User story de edição do perfil do usuário	51
Quadro 5 – User story de visualização de postagens	52
Quadro 6 – User story de visualização de links úteis	52
Quadro 7 – User story de visualização de atualizações de postagens	52
Quadro 8 – User story de busca por postagens	53
Quadro 9 – User story de alteração de permissões de usuários	53
Quadro 10 – User story de criação de postagens	54
Quadro 11 – User story de edição de postagens	54
Quadro 12 – User story de exclusão de postagens	55
Quadro 13 – User story de gerenciamento de categorias	55

LISTA DE ALGORITMOS

Algoritmo 1 – Exemplo de inserção e consulta por documentos em uma coleção no <i>MongoDB</i>	21
Algoritmo 2 – Exemplo de criação e utilização de um índice de busca por texto completo em uma coleção no <i>MongoDB</i>	24
Algoritmo 3 – Exemplo de configuração e <i>upload</i> de arquivos para um <i>bucket</i> criado no <i>Cloudflare R2</i> usando a <i>API S3</i>	25
Algoritmo 4 – Exemplo de configuração do transportador para o envio de e-mails de confirmação usando o <i>nodemailer</i> com uma caixa de e-mail gerada previamente na <i>Brevo</i>	28
Algoritmo 5 – Exemplo de configuração para receber notificações utilizando <i>Expo</i>	29
Algoritmo 6 – Exemplo de configuração do servidor para enviar notificações utilizando o <i>Expo</i>	30

LISTA DE ABREVIATURAS E SIGLAS

<i>API</i>	<i>Application Programming Interface</i>
<i>BSON</i>	<i>Binary JSON</i>
<i>IFPB</i>	<i>Instituto Federal de Educação, Ciência e Tecnologia da Paraíba</i>
<i>FCM</i>	<i>Firebase Cloud Messaging</i>
<i>GCM</i>	<i>Google Cloud Messaging</i>
<i>HTML</i>	<i>HyperText Markup Language</i>
<i>HTTP</i>	<i>Hypertext Transfer Protocol</i>
<i>JSON</i>	<i>JavaScript Object Notation</i>
<i>NoSQL</i>	<i>Not Only SQL (Structured Query Language)</i>
<i>R2</i>	<i>Cloudflare R2 Storage</i>
<i>REST</i>	<i>Representational State Transfer</i>
<i>S3</i>	<i>AWS S3 Storage</i>
<i>SMTP</i>	<i>Simple Mail Transfer Protocol</i>
<i>TA</i>	<i>Test Acceptance</i>
<i>TCC</i>	<i>Trabalho de Conclusão do Curso</i>
<i>TLS</i>	<i>Transport Layer Security</i>
<i>US</i>	<i>User Story</i>
<i>XML</i>	<i>Extensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	PROBLEMÁTICA	11
1.2	DESCRIÇÃO DA SOLUÇÃO	14
1.3	OBJETIVOS	15
1.3.1	Objetivo Geral	15
1.3.2	Objetivos Específicos	15
1.4	ATIVIDADES	15
1.5	ORGANIZAÇÃO DO DOCUMENTO	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	PROCESSO DE DESENVOLVIMENTO	17
2.2	SISTEMA DE ARMAZENAMENTO DE DADOS	18
2.2.1	Dados não relacionais (NoSQL)	18
2.2.2	<i>MongoDB</i>	19
2.2.3	Índices	21
2.2.4	Busca por Texto Completo	23
2.3	SISTEMA DE ARMAZENAMENTO DE OBJETOS	24
2.4	SISTEMA DE AUTENTICAÇÃO	26
2.5	SISTEMA DE NOTIFICAÇÕES	29
3	MURAL WEB: UM APLICATIVO MÓVEL QUE DISPONIBILIZA INFORMAÇÕES ACADÊMICAS DE INSTITUIÇÕES	31
3.1	ANÁLISE	31
3.1.1	Histórias de Usuário	31
3.1.2	Protótipos	33
3.2	PROJETO	34
3.2.1	Arquitetura do sistema	34

3.2.2	Tecnologias utilizadas	36
3.3	Funcionamento da aplicação	37
4	CONSIDERAÇÕES FINAIS	46
	REFERÊNCIAS	47
	APÊNDICE A – HISTÓRIAS DE USUÁRIO	50
	APÊNDICE B – CAPTURAS DE TELA DA APLICAÇÃO	56

1 INTRODUÇÃO

Ao longo dos anos, a demanda por produtos digitais que oferecem mais qualidade e acessibilidade aos usuários tem crescido bastante. Essa demanda é impulsionada pelos modelos de aplicação que apresentam mais eficiência e estruturação em suas interfaces disponibilizadas aos usuários. Segundo Junior (2017), para garantir uma interação de qualidade, as interfaces devem proporcionar alto desempenho, agilidade nas operações, clareza, facilidade de uso, estruturação e conseguir abranger variedades de usuários.

Ao considerar este cenário, é fundamental que o levantamento de requisitos seja bem formulado, para que o processo de desenvolvimento entregue um produto de valor, que atenda às necessidades e características do público-alvo. Em caso de não cumprimento dessas necessidades, a aplicação pode ser abandonada por alternativas mais eficientes (SILVA et al., 2020).

É importante notar que, além da interface e interação, a eficiência de uma aplicação depende de um sistema que tenha capacidade de lidar com as solicitações recebidas pelo usuário. Esse sistema é responsável por processar e responder às interações humano-computador nas aplicações (YABLONSKI, 2020). Interfaces que promovem melhores interações não apenas facilitam o acesso à informação, mas também melhoram significativamente a experiência do usuário.

Considerando a importância de produtos digitais de qualidade com interfaces intuitivas e interativas, este trabalho destaca problemas de comunicação e usabilidade presentes no site do Instituto Federal da Paraíba (IFPB) e tem como objetivo propor uma aplicação que agregue e disponibilize informações acadêmicas relevantes de forma clara, eficiente e acessível, promovendo uma experiência de uso mais satisfatória quando comparada ao site atual.

1.1 PROBLEMÁTICA

Sites institucionais, especialmente de órgãos públicos, costumam apresentar informações de forma generalizada e com pouca organização intuitiva. Essa abordagem pode dificultar a navegação, especialmente para usuários que precisam localizar informações específicas de forma ágil (MATOS et al., 2021).

Na perspectiva de Matos et al. (2021), a usabilidade é um aspecto crucial para melhorar a experiência dos usuários, considerando fatores como eficácia e satisfação ao atingir seus objetivos. A norma ISO 9241 NBR (2002) reforça essa relação ao destacar

critérios que permitem avaliar sistemas sob três dimensões, que são: a facilidade de aprendizado, a capacidade de memorização e a redução de erros durante o uso. Esses critérios são fundamentais para garantir uma interação eficaz em sistemas digitais.

O site do IFPB¹ oferece opções de acessibilidade e funcionalidades destinadas a diferentes perfis de usuários, como alunos, professores e a comunidade acadêmica. Contudo, a falta de uma interface intuitiva prejudica a localização de informações importantes, como notícias ou informações institucionais. Isso resulta em uma experiência frustrante para os usuários, que frequentemente levam mais tempo para buscar uma informação.

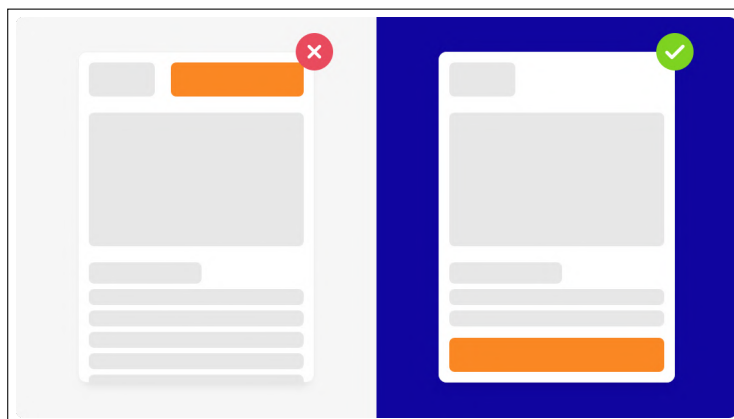
Conforme descrito por Terra et al. (2021), o ser humano tem buscado cada vez mais locais e formas de obter informações com maior rapidez, eficiência e usabilidade. Entretanto, quem utiliza o site do IFPB consegue notar que ele não atende a essas expectativas, o que pode gerar insatisfação e descontentamento entre seus usuários, baseando-se na forma como as informações do site estão disponibilizadas, bem como na quantidade de tempo que o usuário necessita para encontrar uma informação.

Baseando-se em Yablonski (2020), a lei de Hick-Hyman da psicologia aplicada ao design de experiência do usuário destaca que o tempo necessário para uma tomada de decisão aumenta proporcionalmente baseado na quantidade e na complexidade das escolhas disponíveis. A ideia é fortalecida quando é descrito que uma interface que oferece diversas opções em um mesmo local não possui eficiência ou elegância, e isso mostra que os criadores do produto não entendem as necessidades dos usuários.

A aplicação da lei de Hick-Hyman é mostrada no lado direito da Figura 1. Quando o botão laranja é movido para a parte inferior da tela, a complexidade e o tempo necessário para o usuário tomar uma decisão são reduzidos devido à aproximação do objetivo com o usuário quando comparado com a tela mostrada no lado esquerdo da figura.

¹ <https://ifpb.edu.br>

Figura 1 – Exemplo de formulário estruturado de duas maneiras



Fonte: Adaptado de (SOUSA, 2020)

Ao considerar a perspectiva de Yablonski (2020), é evidente que o site do IFPB se encaixa na descrição, pois já nos primeiros acessos ele pode deixar o usuário confuso devido a várias opções disponibilizadas com uma organização ineficiente. Este é o cenário que dificulta o acesso às informações e aumenta o tempo necessário tanto em leitura quanto na execução das operações necessárias pelas páginas.

Em relação à barra de pesquisa, que deveria minimizar as dificuldades de encontrar informações, observa-se que ao fazer uma pesquisa, o resultado obtido engloba todos os campus do IFPB, mesmo que o usuário tenha acessado um dos portais de determinado campus para fazer a busca. Esse problema demonstra uma deficiência no levantamento dos requisitos durante o processo de desenvolvimento do site, pois deveria existir uma funcionalidade de filtragem que permitisse distinguir qual campus seria utilizado para buscar as informações.

Como alternativa ao site, alguns campus adicionalmente compartilham informações em redes sociais, tornando o acesso mais fácil. Uma estratégia complementar a isso seria a utilização de chamadas nas redes sociais oficiais do IFPB, como Instagram ou Facebook, com links diretos para áreas específicas do site. Isso facilitaria o acesso do usuário e ajudaria a divulgar as informações mais relevantes de forma acessível e rápida.

No entanto, o uso de redes sociais como principal fonte de informações institucionais apresenta desafios. Apesar de sua praticidade, o excesso de publicações e a possibilidade de disseminação de informações incorretas, como destacado por Garcia (2014) e Delmazo e Valente (2018), tornam inadequado depender exclusivamente dessas plataformas. Esses resultados reforçam a necessidade de aprimorar o site

institucional para que ele cumpra seu papel como principal fonte de informações oficiais, diminuindo a dependência de redes sociais.

Ainda assim, Garcia (2014) descreve que as redes sociais são consideradas como avanços tecnológicos, que oferecem oportunidades para melhorar o engajamento e aprimorar várias áreas, existindo relatos de sucesso de docentes e tutores em tentativas de aplicar essa tecnologia no meio institucional.

Mas Garcia (2014) também descreve que essas mesmas possibilidades podem trazer consequências, pois a instituição pode perder oportunidades de melhoria em seus serviços, perder investimentos feitos nos serviços que foram criados para ter mais engajamento com os alunos, perder interação e comunicação com os alunos, relacionadas a vários fatores do ambiente das redes sociais, dentre outros problemas.

1.2 DESCRIÇÃO DA SOLUÇÃO

Para resolver o problema descrito, a ideia de implementar melhorias no site para aprimorar a experiência do usuário durante a navegação e a busca por informações seria uma alternativa. No entanto, essa proposta exige tempo e custos financeiros para a análise dos requisitos necessários e o desenvolvimento de correções em uma plataforma que já está em produção, podendo possuir tecnologias legadas, resultando em mais trabalho para inserir atualizações, ajustar mudanças e *bugs* durante o processo.

Uma abordagem alternativa que já vem sendo utilizada são as redes sociais, que não devem ser utilizadas como o principal meio de divulgação da instituição devido aos problemas já mencionados. No entanto, faria sentido utilizá-las para divulgar apenas materiais de eventos ou notícias que merecerão conhecimento do público fora da instituição, e manter outros tipos de materiais em ambientes institucionais, para que não exista dependência de meios não institucionais ou outros problemas.

Entretanto, é importante ressaltar que o objetivo deste Trabalho de Conclusão de Curso não está direcionado ao uso de redes sociais nem à melhoria do site existente. A solução proposta é o desenvolvimento de uma aplicação móvel que ofereça um ambiente amigável e acessível para a disponibilização rápida e interativa de informações acadêmicas. Além disso, a aplicação contará com um sistema de notificações para alertar sobre novas postagens, aprimorando a experiência do usuário e reduzindo acessos desnecessários. Juntamente com a aplicação móvel, o projeto contará com uma plataforma de gerenciamento da aplicação onde gestores e demais responsáveis vão publicar os conteúdos da aplicação de forma simples e sem a necessidade de possuir conhecimento técnico avançado.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

O objetivo geral deste Trabalho de Conclusão de Curso é desenvolver um sistema que melhore a experiência dos usuários na obtenção de informações acadêmicas do IFPB, utilizando uma arquitetura multiplataforma que possui módulos baseados em API com os serviços distribuídos. Dessa forma, a arquitetura propõe escalabilidade e flexibilidade, permitindo atender às demandas de crescimento e adaptação às necessidades dos usuários, enquanto visa solucionar os problemas na comunicação entre a instituição e suas principais categorias: estudantes, servidores e o público em geral.

1.3.2 Objetivos Específicos

- Proporcionar aos usuários um aplicativo *mobile* que facilite o acesso às informações acadêmicas de maneira mais eficiente e interativa.
- Disponibilizar uma plataforma *web* administrativa que permita o cadastro e a gestão das informações acadêmicas apresentadas no aplicativo *mobile*.
- Possibilitar aos usuários da aplicação *mobile* o recebimento de notificações sobre novas postagens sem a necessidade de acessar a aplicação.
- Permitir a busca e filtragem por postagens no aplicativo *mobile*, oferecendo opções de filtragem por data, ordenação e tipo de informação, além da busca textual avançada baseada na indexação de documentos.
- Garantir aos administradores total controle de acesso sobre a aplicação, possibilitando a gestão completa das informações acadêmicas e de usuários cadastrados.

1.4 ATIVIDADES

Durante o processo de desenvolvimento desse Trabalho de Conclusão de Curso, as atividades foram organizadas para serem realizadas da seguinte forma:

- **Elaboração da ideia (A1):** Nessa etapa, foram realizados estudos para elaborar uma ideia para o tema do projeto, junto da definição de uma problemática e uma solução para o tema escolhido.
- **Escolha das Tecnologias (A2):** A partir daqui, foram realizados estudos para escolher as tecnologias necessárias para o desenvolvimento do projeto.

- **Estudo do Acervo Teórico (A3):** Após escolher as tecnologias, foram conduzidas pesquisas em busca de trabalhos e teorias que estão relacionados com o tema do trabalho para que seja adquirido um embasamento teórico que sustente a proposta desse trabalho.
- **Análise e Projeto do Sistema (A4):** Durante as pesquisas da fundamentação teórica, foram definidos os quadros com as histórias de usuário, a prioridade, o nível de dificuldade, e os requisitos necessários para cada história. E ao mesmo tempo, foram criados os protótipos baseados nessas histórias, foi definido a arquitetura utilizada, e também foram feitas as descrições das tecnologias utilizadas.
- **Elaboração do Documento (A5):** Conforme as atividades acima foram sendo definidas, ocorreu a elaboração do documento de qualificação do trabalho e documentação da aplicação.
- **Atualização do Projeto do Sistema (A6):** Durante o desenvolvimento do trabalho e da aplicação, foi necessário refinar e refatorar o projeto, juntamente das histórias de usuário e requisitos.
- **Desenvolvimento da Aplicação (A7):** Ao preparar todas as tecnologias necessárias e finalizar a análise e projeto do sistema, foi iniciado o processo de desenvolvimento da aplicação acompanhado por testes automatizados e validação de dados. Durante essa etapa, ocorreram diversos ajustes e alterações no que foi planejado.
- **Atualização do Documento (A8):** Após a criação da aplicação final, foi necessário fazer alterações no documento para incluir novas funcionalidades, tecnologias e requisitos que não estavam previstos ou que precisaram serem alterados durante o processo de desenvolvimento da aplicação.

1.5 ORGANIZAÇÃO DO DOCUMENTO

As demais partes deste trabalho são destacadas em mais três capítulos. O Capítulo 2 descreve a fundamentação teórica do trabalho, onde são mostradas afirmações baseadas em trabalhos científicos, livros e documentos sobre os conceitos necessários acerca das tecnologias que serão utilizadas no desenvolvimento da aplicação. No Capítulo 3 é mostrada a análise da aplicação juntamente com o levantamento de requisitos e histórias de usuário, protótipos da aplicação, além da arquitetura e tecnologias utilizadas no projeto. Já o Capítulo 4 destaca as considerações finais desse documento, além das expectativas finais do projeto.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo destaca os conceitos e tecnologias que serão essenciais durante o processo de desenvolvimento da aplicação proposta. Serão discutidos conceitos sobre *Scrum*, Prototipagem, Bancos de Dados *NoSQL*, Bancos de Dados Orientados a Documentos, Busca por Texto Completo no *MongoDB*, Linguagem de Programação *Javascript* e alguns de seus *frameworks* e ferramentas.

2.1 PROCESSO DE DESENVOLVIMENTO

Ao criar um projeto sem uma ideia totalmente formulada, é essencial escolher uma abordagem para o desenvolvimento do projeto que permita adaptação e entregas constantes de novas funcionalidades sem distorcer a ideia inicial do projeto. Nesse caso, a utilização de métodos ágeis é indispensável, pois resultará em um projeto mais flexível, eficaz e alinhado com as expectativas do público-alvo (NUNES, 2017).

Nesse contexto, a escolha do *Scrum* como método ágil e a prototipação como técnica ágil de desenvolvimento fornecem uma estrutura ideal, visto que o *Scrum* adota um processo iterativo e incremental contínuo, enquanto a prototipagem oferece uma visão inicial do projeto antes de estruturá-lo (PRESSMAN; MAXIM, 2016).

A metodologia *Scrum* que será adotada destaca a criação de um artefato essencial chamado *backlog*, que é utilizado para organizar e priorizar as atividades do projeto a ser cumprido, listando as tarefas pendentes de acordo com a sua importância. No *backlog*, as tarefas podem ser categorizadas como histórias de usuário — descrições que refletem como o usuário deseja ver determinada funcionalidade no projeto — e como tarefas, que correspondem aos requisitos necessários para a implementação dessas histórias (COHN, 2004).

Conforme registradas no *backlog*, as pendências são gradualmente movidas para o quadro de implementação denominado *sprint*, que garante uma abordagem interativa e focada na entrega contínua de novas funcionalidades, permitindo controlar o que está sendo implementado na aplicação durante um período específico baseado nas prioridades das tarefas e histórias de usuários no *backlog* do projeto (COHN, 2004).

De acordo com Pressman e Maxim (2016), quando não existe uma ideia clara do objetivo final do projeto, a inclusão de prototipação no processo de desenvolvimento é benéfica, pois permite a visualização concreta das ideias antes da implementação, o que facilita a validação de conceitos, obtenção de *feedback* dos usuários e otimização do *design* e usabilidade. Isso garante redução de trabalhos, maximização de

desempenho e que as funcionalidades atendam melhor às demandas do público-alvo.

No projeto, o fluxo de trabalho com a metodologia *Scrum* será feito por meio de reuniões semanais para fazer planejamentos, recapitulação da *sprint* semanal concluída e avaliação de quais tarefas e histórias de usuário serão implementadas na nova *sprint* semanal. Essa abordagem flexível permitirá ajustes contínuos à medida que o projeto evolui.

2.2 SISTEMA DE ARMAZENAMENTO DE DADOS

Para armazenar dados, é necessário utilizar um sistema de gerenciamento de banco de dados externo. Para a ideia do projeto, o sistema escolhido foi o *MongoDB*, um tipo de banco de dados considerado como não relacional, ou *NoSQL*, projetado para armazenar dados em forma de documentos, ou orientado a documentos.

2.2.1 Dados não relacionais (*NoSQL*)

De acordo com o Cloud (2023), os bancos de dados *NoSQL* diferem dos bancos de dados relacionais por adotarem um formato de dados não tabular. Essa característica permite que esses bancos operem com um modelo de esquema flexível, que proporciona maior adaptabilidade e abrange uma variedade de tipos de modelos de dados não estruturados.

Os principais tipos de modelos de dados são os de documentos, colunas, grafos e chave-valor. O modelo de documento armazena dados em coleções de documentos flexíveis com organização adaptável. O modelo de coluna armazena e lê dados em linhas e os organiza em colunas com formatações diferentes de modelos relacionais, otimizando consultas analíticas e facilitando recuperações seletivas de dados específicos. O modelo de grafos representa dados em forma de grafos, destacando relações complexas por meio de nós e arestas, ideal em casos de mapeamento de relacionamentos. Já o modelo de chave-valor simplifica o armazenamento, associando valor a chaves únicas para operações rápidas de leitura e escrita (LÓSCIO et al., 2011).

Ainda segundo Lóscio et al. (2011), a escolha entre esses modelos é guiada pelas características específicas de cada aplicação, visando atender aos requisitos necessários. Baseado nisso, o *MongoDB* foi escolhido por possuir escalabilidade e alta disponibilidade (CLOUD, 2023). Essas características serão úteis ao transmitir novas postagens aos usuários através de distribuições de informações em larga escala como notificações que vários usuários poderão acessar simultaneamente.

2.2.2 *MongoDB*

Conforme afirmado por Souza e Oliveira (2019), o *MongoDB*¹ é um sistema de gerenciamento de dados com a proposta de facilitar o trabalho do utilizador, proporcionando notável desempenho e disponibilidade em consultas e operações. Sua arquitetura permite escalabilidade horizontal, distribuição geográfica e manipulação eficaz de enormes volumes de dados. Além disso, o *MongoDB* oferece flexibilidade e simplicidade na realização de consultas.

Vale destacar que, além de um serviço de gerenciamento de dados local, o *MongoDB* disponibiliza um serviço em nuvem chamado *MongoDB Atlas*². Dessa forma, esse *SaaS* remove toda a complexidade e a necessidade de configurar e escalar todo um ambiente de banco de dados em produção ao disponibilizar a aplicação globalmente. Além disso, esse serviço pode ser usado gratuitamente até um certo limite, o que pode beneficiar desenvolvedores que procuram serviços gratuitos para testar sua aplicação.

Conforme o *MongoDB* (2023), os documentos armazenados no banco de dados *MongoDB* podem estar em formatos *JSON*, *BSON* e *XML*. Esses documentos são armazenados em coleções, ou *collections*, em pares de campo-valor, com tipos e estruturas que podem variar uns dos outros.

Na Figura 2, é apresentado um documento que representa um objeto armazenado em *BSON* no *MongoDB*. Esse documento contém atributos de identificação de uma postagem, incluindo campos exibidos no formato *ObjectId()*, que é o tipo padrão para identificadores primários no *MongoDB*. Esse mesmo documento pode ser inserido e consultado no banco de dados, com a possibilidade de a consulta ser feita dentro de qualquer atributo do documento.

¹ <https://mongodb.com>

² <https://mongodb.com/atlas/database>

Figura 2 – Exemplo de documento que representa um objeto *BSON* armazenado no *MongoDB*

```
_id: ObjectId('676b07c8a1f91674edef8f0f')
title: "Cribro alii debitis caecus."
is_public: true
created_by: ObjectId('676b07c8a1f91674edef8efe')
category_id: ObjectId('676b07c8a1f91674edef8f01')
▼ content: Array (11)
  ▼ 0: Object
    type: "paragraph"
    alignment: "right"
    ▶ children: Array (14)
  ▶ 1: Object
  ▶ 2: Object
  ▶ 3: Object
  ▶ 4: Object
  ▶ 5: Object
  ▶ 6: Object
  ▶ 7: Object
  ▶ 8: Object
  ▶ 9: Object
  ▶ 10: Object
created_at: 2024-12-24T19:13:12.722+00:00
updated_at: 2024-12-24T19:13:12.722+00:00
```

Fonte: Elaborado pelo autor (2024)

A variação nas estruturas e tipos faz com que informações inseridas em um documento sejam flexíveis, com capacidade do documento perder ou receber parâmetros se for necessário. Essa forma de armazenamento de dados aumenta o desempenho das consultas ao banco, pois o sistema possibilita buscar dados baseados em qualquer parâmetro do documento, sem a necessidade de ver o documento inteiro para tomar uma decisão (CLOUD, 2023).

Conforme descrito pelo MongoDB (2023), cada coleção guarda documentos com conteúdo semelhantes mas sem a necessidade de que todos os documentos tenham os mesmos atributos presentes obrigatoriamente. E mesmo com toda essa flexibilidade, existem opções de validação do esquema que exigem mais rigidez aos campos, sendo algo opcional.

Ainda baseando no MongoDB (2023), quando existe ausência de determinado campo em um documento, não existe nulidade ou adição de atributos nulos baseados na coleção, ele é salvo da forma que está dependendo das opções de validação. O Algoritmo 1 mostra os passos para se criar uma *database* e uma coleção no *MongoDB*, além de mostrar também a inserção e consulta completa por documentos na coleção.

Algoritmo 1 – Exemplo de inserção e consulta por documentos em uma coleção no MongoDB

```

1 > use example // criando uma database no mongodb
2 > db.Example.insertMany([ // criando uma collection e inserindo nela
3   { name: 'José', description: 'José é um dos professores.' },
4   { name: 'Maria', description: 'Maria é uma das professoras.' },
5   { name: 'Francisco' }
6 ])
7 > db.Example.find() // solicita todos os documentos da collection
8 [ // resultado retornado
9   {
10    _id: ObjectId("655b735752b025b8f443b44d"),
11    name: 'José',
12    description: 'José é um dos professores.'
13  },
14  {
15    _id: ObjectId("655b735752b025b8f443b44e"),
16    name: 'Maria Silva',
17    description: 'Maria é uma das professoras.'
18  },
19  {
20    _id: ObjectId("655b735752b025b8f443b44f"),
21    name: 'Francisco'
22  }
23 ]

```

Fonte: Elaborado pelo autor (2023)

Durante o desenvolvimento da aplicação, será necessário fazer uso tanto da validação como da flexibilidade dos esquemas que foram mencionados previamente. O uso da validação dos esquemas ocorrerá quando um novo usuário tentar se cadastrar na aplicação, sendo obrigatório inserir os campos de nome, e-mail e senha para a conta ser criada, pois esses campos serão necessários para identificar e autenticar o usuário. Enquanto isso, a flexibilidade dos esquemas será utilizada ao cadastrar uma postagem, que tratará a imagem como atributo opcional, pois algumas postagens não vão precisar da imagem de exibição.

2.2.3 Índices

Dentro das coleções, existem estruturas que tornam o banco de dados mais produtivo, como é o caso dos índices, ou *indexes*, que melhoram a eficiência das operações de leitura. Conforme destacado por Clifton e Garcia-Molina (2000), a implementação de índices é fundamental quando existem consultas ao banco de dados baseadas em campos dos documentos de uma coleção, pois os índices evitam a necessidade de realizar buscas completas em um documento, otimizando as consultas. Caso o documento não tenha o campo requisitado pelo índice, ele é descartado da consulta.

No projeto, é necessário utilizar índices para efetuar operações de busca de diferentes formas. No geral, as consultas necessárias poderão envolver parâmetros únicos, múltiplos ou textuais. Para isso, a plataforma *MongoDB* disponibiliza diversos tipos de índices para atender às necessidades de indexação e consulta. Dentre eles, serão utilizados os índices de campo único, índices compostos e índices de texto.

Conforme o Junior (2019), em qualquer documento de uma coleção, um campo `_id` e um índice chamado de índice de campo único são gerados automaticamente. Este índice utiliza o identificador único do documento para fazer buscas rápidas e eficientes. Além disso, o índice de campo único pode ser criado manualmente utilizando outros campos dos documentos de uma coleção.

Já o índice composto trabalha de maneira semelhante ao índice de campo único. Ainda segundo o Junior (2019), esse índice oferece a flexibilidade de indexar vários campos em conjunto, permitindo otimizar consultas que envolvem múltiplos campos ou possibilitar buscas que envolvem critérios em diversos campos da coleção.

O índice de texto é o tipo que facilita a busca por texto completo em uma coleção. Ele é útil quando existe a necessidade de fazer consultas através de palavras-chave em títulos, descrições e outros campos, pois esse índice não possui limitações em correspondências exatas ou expressões regulares, oferecendo uma busca rápida e suporte a funcionalidades como tokenização apropriada ao idioma, pesos de termos, *case insensitive*, *stopwords* e *stemming* (CHODOROW et al., 2018).

Ainda segundo Chodorow et al. (2018), ao fazer uma busca utilizando o índice de texto, a *string* é dividida em unidades menores utilizando delimitadores que podem ser espaços em branco ou a maioria dos sinais de pontuação da *string*. Em termos mais simples, a busca não considera o texto buscado como uma sequência contínua de caracteres, mas sim como uma série de unidades. Essas unidades são os tokens resultantes da tokenização. Depois disso, serão aplicadas as outras funcionalidades que foram descritas anteriormente. Vale ressaltar que algumas dessas propriedades são opcionais e podem ou não ser utilizadas durante o processo de busca.

A funcionalidade do projeto que usará o índice de texto é a barra de pesquisar, que receberá entradas de texto do usuário para fazer a busca utilizando o índice. Vale destacar que uma coleção pode ter apenas um índice de texto, mas que o índice pode englobar vários campos do documento (FOTOPOULOS et al., 2023).

2.2.4 Busca por Texto Completo

Além do índice de texto, a barra de busca necessita de outra funcionalidade do *MongoDB*, que é chamada de busca por texto completo (*full-text search*). Segundo Chaitanya et al. (2019), a busca recebe palavras como parâmetros textuais, busca algo que corresponda aos parâmetros dentro de extensos dados de texto armazenados, e retorna resultados que podem conter uma, algumas ou todas as palavras da consulta.

Conforme descrito por Fotopoulos et al. (2023), para que o *MongoDB* suporte a busca por texto completo, é necessário criar um índice de texto no campo do tipo *string* que será pesquisado especificando o literal de *string* "text" nas propriedades desse campo. Ao efetuar a pesquisa, deve-se utilizar o parâmetro *\$text* na operação de busca por texto completo. O operador *\$text* suporta várias opções de pesquisa, como a pesquisa insensível a maiúsculas e minúsculas, pesquisa de frase, etc.

O Algoritmo 2 mostra a inserção de um novo documento na coleção criada pelo Algoritmo 1, além do passo a passo para a criação e utilização do índice de texto em uma busca por texto completo. Esse índice considera dois atributos da coleção, atribuindo pesos distintos a cada um deles.

Após a execução da busca, o resultado retorna todos os documentos que contêm variações da palavra pesquisada, independentemente de flexões de gênero (professor/professora), número (professor/professores) ou posição dentro do texto (*name* ou *description*). Esse comportamento ocorre porque o mecanismo de *full-text search* do *MongoDB* emprega técnicas como *stemming*, que reduzem as palavras às suas raízes, permitindo consultas mais flexíveis e abrangentes. Além disso, o primeiro documento retornado corresponde ao atributo com maior peso, seguido dos demais resultados.

Algoritmo 2 – Exemplo de criação e utilização de um índice de busca por texto completo em uma coleção no *MongoDB*

```

1 > db.Example.insertOne( // inserindo um novo documento
2   {name: 'Aparecida Professora'}
3 )
4 > db.Example.createIndex( // criando index full-text search
5   {name: 'text', description: 'text'},
6   {default_language: 'pt', weights: { name: 2, description: 1 }}
7 )
8 > db.Example.find( // buscando com full-text search index
9   { $text: { $search: 'professor' } }
10 )
11
12 [ // resultado retornado
13   {
14     _id: ObjectId("655b76ce52b025b8f443b450"),
15     name: 'Aparecida Professora'
16   },
17   {
18     _id: ObjectId("655b735752b025b8f443b44e"),
19     name: 'Maria',
20     description: 'Maria é uma das professoras.'
21   },
22   {
23     _id: ObjectId("655b735752b025b8f443b44d"),
24     name: 'José',
25     description: 'José é um dos professores.'
26   }
27 ]

```

Fonte: Elaborado pelo autor (2023)

2.3 SISTEMA DE ARMAZENAMENTO DE OBJETOS

Na aplicação, é necessário utilizar um sistema de armazenamento de objetos para armazenar arquivos de imagens das postagens. Para suprir essas necessidades, será utilizado o serviço de armazenamento em *bucket* chamado *R2*, que pertence à *Cloudflare*³, uma plataforma que disponibiliza diversos serviços de *Cloud Computing* para desenvolver projetos.

Conforme Zhao (2022) descreve, o *Cloudflare R2* é um serviço de armazenamento de objetos que elimina as taxas de saída de dados, permitindo a criação de arquiteturas flexíveis e multinuvem, além de reduzir custos associados ao tráfego de rede.

³ <https://cloudflare.com/developer-platform/products/r2>

Compatível com a *API S3*⁴, o *R2* oferece flexibilidade ao acessar uma ampla gama de ferramentas e bibliotecas *S3*, facilitando a integração com sistemas existentes ou migração de armazenamento sem a necessidade de fazer muitas mudanças (CLOUDFLARE, 2024). Além disso, o *R2* possui um nível gratuito que pode ser útil para desenvolvedores que procuram serviços gratuitos para armazenar objetos.

No Algoritmo 3 é apresentado o fluxo de configuração e *upload* de um arquivo para o *R2* usando a *API S3* em uma biblioteca oficial para *Javascript*. Supondo que o acesso já esteja configurado na plataforma da *Cloudflare*, é necessário inserir as credenciais de segurança de forma oculta nas variáveis de ambiente para que possam ser enviados os arquivos.

Algoritmo 3 – Exemplo de configuração e *upload* de arquivos para um *bucket* criado no *Cloudflare R2* usando a *API S3*.

```

1 import { S3 } from "@aws-sdk/client-s3";
2 const fs = require('node:fs');
3
4 const client = new S3Client({
5   region: 'us-east-1', // o R2 só aceita us-east-1 ou auto
6   endpoint: process.env.ENDPOINT, // direciona trabalho para o R2
7   credentials: { // credenciais de acesso do bucket R2
8     accessKeyId: process.env.ACCESS_KEY_ID,
9     secretAccessKey: process.env.SECRET_ACCESS_KEY,
10  }
11 });
12
13 const uploadToS3 = async (file) => {
14   const command = new PutObjectCommand({ // monta objeto para upload
15     Bucket: 'meu-bucket', // nome do bucket de destino
16     Key: file.key,
17     Body: file.data,
18   });
19
20   await s3.send(command); // envia o comando pelo cliente gerado
21 };
22
23 (async () => {
24   try {
25     const data = fs.readFileSync('imagem.png');
26     await uploadToS3({ key: 'imagem.png', data });
27   } catch (error) {
28     console.log(error);
29   }
30 })();

```

Fonte: Elaborado pelo autor (2024)

⁴ <https://developers.cloudflare.com/r2/api/s3/api>

2.4 SISTEMA DE AUTENTICAÇÃO

Aplicações que disponibilizam informações publicamente sem necessidade de autenticação possuem áreas administrativas em ambientes *desktop* ou *web*, que por razões de segurança só podem ser acessadas por utilizadores que possuem acessos administrativos e, dessa forma, requerem uma forma para se autenticarem (MABUNDA; SILVA, 2017).

Para garantir a segurança e privacidade dos usuários, a aplicação utiliza diversas técnicas para evitar a exposição dos dados e garantir sua proteção. Entre essas técnicas, destaca-se a criptografia das senhas, a criação de *tokens* de autenticação imutáveis (sem dados sensíveis) para identificação do usuário, e várias formas de validação para o acesso às áreas privadas, como o controle do propósito da autenticação e a validação da conta via e-mail. Além disso, são utilizados canais e protocolos seguros para o envio de e-mails ou notificações para dispositivos móveis dos usuários.

Diante disso, a forma de autenticação dessa aplicação se dá por gerar o primeiro usuário administrador no banco de dados, e depois esse administrador poderá criar contas para outros usuários utilizando endereços de e-mail legítimos para que possam ser enviados a senha gerada automaticamente e o link de confirmação da conta.

Para o envio de e-mails de forma segura e de uso profissional, é necessário criar uma caixa de e-mails proprietária, e para criar isso existem várias formas. A forma abordada nessa aplicação será utilizar a plataforma de serviço da *Brevo*⁵, uma plataforma de marketing que oferece soluções de relacionamento com o cliente, inclusive soluções de e-mail. Essa plataforma possui atualmente um limite de 300 e-mails gratuitos que podem ser enviados diariamente; é ideal para fazer testes ou até mesmo para aplicações de pequeno porte.

Ao gerar a caixa de e-mail, são recebidas credenciais que podem ser utilizadas em bibliotecas de mensageria, e a biblioteca utilizada para enviar e-mails profissionais no projeto foi o *nodemailer*⁶, uma das bibliotecas mais utilizadas do *Node.js* para o envio de e-mails, destacando-se pela simplicidade, eficiência e segurança na integração com sistemas que necessitam de funcionalidade de mensageria, (NODEMAILER, 2024). Entre os recursos do *nodemailer* está o suporte ao transporte *Simple Mail Transfer Protocol (SMTP)* com *TLS/STARTTLS* e envio de *HyperText Markup Language (HTML)*, que são os principais recursos que serão utilizados para criar um transportador de

⁵ <https://brevo.com>

⁶ <https://nodemailer.com>

mensagens com as credenciais da caixa de e-mails da *Brevo* para poder enviar e-mails de forma segura.

No Algoritmo 4 é apresentado um exemplo do fluxo de configuração do transportador na constante *transporter* e do envio de um e-mail pela função *sendConfirmationEmail* que recebe os dados do usuário necessários para o envio do e-mail de confirmação da conta, e utiliza o transportador configurado para realizar a ação através do host da *Brevo*, que antes se chamava *Sendinblue*.

Algoritmo 4 – Exemplo de configuração do transportador para o envio de e-mails de confirmação usando o *nodemailer* com uma caixa de e-mail gerada previamente na *Brevo*.

```

1 import nodemailer from 'nodemailer';
2
3 const transporter = nodemailer.createTransport({
4   host: 'smtp-relay.brevo.com', // host de serviço da brevo
5   port: 587, secure: false, requireTLS: true,
6   auth: { user: process.env.USER, pass: process.env.PASS },
7 });
8
9 export const sendConfirmationEmail = async (data) => {
10   return await transporter.sendMail({
11     from: 'Example <example@mail.com>', // caixa de e-mail da brevo
12     to: data.email,
13     subject: 'Teste de envio',
14     html: `
15     <!DOCTYPE html>
16     <html lang="pt-BR">
17       <head> ... </head>
18       <body>
19         <div>
20           <p>Olá, <b id="name">${data.name}</b>!</p>
21           <p>Seu cadastrado foi realizado no Mural Web como um <b id=
22             "role">${data.role}</b>.</p>
23           <p>Seu e-mail de acesso é: <b id="email">${data.email}</b>
24             </p>
25           <p>Sua senha é: <b id="password">${data.password}</b></p>
26           <p>Sua conta pode ser confirmada por esse <a href="https://
27             example.com/auth/verify-email?code=${data.token}" id="
28             verify-link">link</a>.</p>
29         </div>
30       </body>
31     </html>
32     `;
33   });
34 }
35
36 (async () => {
37   try {
38     await sendConfirmationEmail({
39       name: 'José Silva',
40       email: 'josesil@example.com',
41       role: 'MODERADOR',
42       password: 'dlKsjk2@Ajz1d547$%3',
43     });
44   } catch (error) {
45     console.log(error);
46   }
47 })();

```

2.5 SISTEMA DE NOTIFICAÇÕES

Segundo Riadh (2016), o uso de notificações em dispositivos móveis nas aplicações institucionais visa otimizar a comunicação com os estudantes no ambiente acadêmico. Através de um sistema de notificações integrado a uma aplicação móvel, sempre que uma nova postagem for adicionada, os estudantes poderão receber notificações diretamente em seus dispositivos móveis em tempo de execução, facilitando a disseminação das informações acadêmicas. Essa abordagem não só economiza tempo e esforço, mas também garante que os estudantes recebam atualizações em tempo real, melhorando a organização e a eficiência no processo educacional.

Para implementar o sistema de notificações na aplicação, será utilizado o serviço *Firebase Cloud Messaging (FCM)*. Segundo Riadh (2016), este serviço, anteriormente conhecido como *Google Cloud Messaging (GCM)*, é uma solução gratuita e eficiente para criar canais de comunicação cliente/servidor em aplicações móveis. Posteriormente, o serviço foi integrado ao *Firebase*⁷ e renomeado.

Antes de implementar as notificações na aplicação, são necessárias credenciais que serão utilizadas na construção da aplicação móvel. Essas credenciais são geradas ao criar um projeto *mobile* com notificações *push* seguindo os documentos⁸ no console do *Firebase* para poder utilizar o serviço *FCM*.

Para fazer a ponte com o *FCM* tanto o servidor como a aplicação móvel utilizarão bibliotecas compatíveis para enviar e receber notificações, dessa forma o Algoritmo 5 mostra um exemplo de configuração de recepção.

Algoritmo 5 – Exemplo de configuração para receber notificações utilizando *Expo*.

```

1 // Na aplicacao móvel é necessário permitir a recepcao de
  notificacoes e capturar o token gerado do dispositivo e armazená-
  lo em algum local para mais tarde ser utilizado como receptor.
2
3 import * as Notifications from 'expo-notifications';
4
5 Notifications.setNotificationHandler({
6   handleNotification: async () => ({
7     shouldShowAlert: true,
8     shouldPlaySound: true,
9     shouldSetBadge: true,
10  }),
11 });

```

Fonte: Elaborado pelo autor (2024)

⁷ <https://firebase.google.com/>

⁸ <https://docs.kii.com/en/samples/push-notifications/push-notifications-android-fcm/create-project>

Já o Algoritmo 6 mostra o exemplo de configuração de envio de mensagens usando o serviço de notificações⁹ do *Expo*, um construtor de aplicações móveis que utiliza *React Native* como base. Para que as notificações possam funcionar, também é necessário criar um projeto no *Expo* e gerar credenciais que servirão como identificação da aplicação móvel e serão utilizadas no servidor para criar o canal de comunicação entre o servidor e o cliente com o *FCM*.

Algoritmo 6 – Exemplo de configuração do servidor para enviar notificações utilizando o *Expo*.

```

1 // O servidor que vai enviar as notificacoes precisa ter conhecimento
  para qual aplicacao e quais dispositivos ele irá enviar as
  notificacoes.
2
3 import Expo from 'expo-server-sdk';
4
5 const expo = new Expo({ // Configura objeto expo com a credencial
6   accessToken: process.env.EXPO_ACCESS_TOKEN,
7 });
8
9 const sendNotification = async (data) => {
10   const chunks = expo.chunkPushNotifications([{
11     to: data.token, // Token do dispositivo salvo
12     title: data.title,
13     body: 'Acesse a postagem para obter mais detalhes',
14   }]);
15
16   await Promise.all([chunks.map(async (chunk) => {
17     await expo.sendPushNotificationsAsync(chunk);
18   })]);
19 }
20
21 (async () => {
22   try {
23     await sendNotification({
24       token: 'ExponentPushToken[*****]',
25       title: 'Nova postagem',
26       body: 'Acesse a postagem par aobter mais detalhes',
27     });
28   } catch (error) {
29     console.log(error);
30   }
31 })();

```

Fonte: Elaborado pelo autor (2024)

⁹ <https://docs.expo.dev/push-notifications/push-notifications-setup>

3 MURAL WEB: UM APLICATIVO MÓVEL QUE DISPONIBILIZA INFORMAÇÕES ACADÊMICAS DE INSTITUIÇÕES

Conforme foi descrito no Capítulo 1, existe a necessidade de um sistema que solucione os problemas de comunicação e usabilidade identificados no site do IFPB. Com base nisso, foi considerada a criação de uma aplicação para resolver esses problemas e proporcionar um ambiente mais interativo.

A aplicação servirá exclusivamente como um agregador de conteúdos acadêmicos, possuindo um ambiente mobile para que as postagens possam ser visualizadas, e um ambiente web para que administradores e usuários com permissões possam inserir e manipular conteúdo na plataforma.

3.1 ANÁLISE

Vale ressaltar que a elaboração deste trabalho e a criação dessa aplicação vão além de apenas construir algo sem padrões. Será necessário analisar as necessidades e requisitos que a aplicação precisa cumprir para ter usabilidade e interatividade. Para fazer isso e proporcionar um melhor desempenho, o processo levará em consideração os fundamentos destacados na seção do processo de desenvolvimento no Capítulo 2.

Portanto, baseado nesses fundamentos, a análise será feita através de histórias de usuário inseridas dentro de quadros para organizar e destacar separadamente as necessidades e requisitos necessários. Além disso, será disponibilizada uma visualização concreta de uma prévia de como será a aplicação final com protótipos.

3.1.1 Histórias de Usuário

Conforme descrito por Longo e Silva (2014), as histórias de usuário servem para descrever os requisitos da aplicação através das vontades do cliente, procurando saber quais são as necessidades dele na aplicação. Normalmente, essas histórias são escritas de forma curta e breve, contando com uma ou duas frases. Além disso, as histórias são fundamentais quando se está utilizando o método ágil *Scrum* no processo de desenvolvimento.

Em conjunto com as histórias de usuário, existem os testes de aceitação da história, que ainda segundo Longo e Silva (2014), eles servem para especificar e expressar como são os requisitos ou comportamentos que o sistema precisa atender para a conclusão daquela história. Além disso, o autor descreve que os testes fazem parte da garantia de que o sistema funcionará corretamente.

Conforme foi descrito, as histórias de usuário ficarão separadas por quadros, onde cada quadro contará com uma história de usuário, uma descrição da história, um nível de dificuldade, esforço ou estimativa para a conclusão da história (*Story Points*), uma prioridade para a execução da história e os testes de aceitação da história (*TA*).

Logo abaixo segue uma breve descrição de cada história de usuário da aplicação. Mas se o leitor tiver interesse em visualizar todos os quadros com as histórias, descrições, estimativas, prioridades e testes de aceitação, o detalhamento de cada uma pode ser encontrado no Apêndice A.

História de Usuário 01: Criar conta

Descrição: Como um administrador autenticado, eu quero criar uma conta para um usuário.

História de Usuário 02: Acessar conta

Descrição: Como usuário cadastrado, quero acessar minha conta.

História de Usuário 03: Recuperar conta

Descrição: Como usuário cadastrado, desejo recuperar o acesso à minha conta cadastrada.

História de Usuário 04: Editar perfil

Descrição: Como usuário cadastrado, desejo alterar as informações da minha conta cadastrada.

História de Usuário 05: Visualizar postagens

Descrição: Como usuário, desejo visualizar as postagens disponíveis.

História de Usuário 06: Visualizar *links* úteis

Descrição: Como usuário, desejo visualizar *links* úteis.

História de Usuário 07: Visualizar atualizações

Descrição: Como usuário, desejo visualizar atualizações/notificações para novas postagens.

História de Usuário 08: Buscar postagens

Descrição: Como usuário, desejo fazer uma busca por postagens.

História de Usuário 09: Alterar permissões de usuários

Descrição: Como usuário administrador autenticado, desejo alterar as permissões dos

usuários cadastrados.

História de Usuário 10: Criar postagem

Descrição: Como usuário administrador ou provedor de conteúdo autenticado, desejo criar uma nova postagem.

História de Usuário 11: Editar postagem

Descrição: Como usuário administrador ou provedor de conteúdo autenticado, desejo editar uma postagem criada anteriormente.

História de Usuário 12: Excluir postagem

Descrição: Como usuário administrador ou provedor de conteúdo autenticado, desejo excluir uma postagem criada anteriormente.

História de Usuário 13: Gerenciar categorias

Descrição: Como usuário administrador autenticado, desejo gerenciar categorias.

3.1.2 Protótipos

Conforme descrito no Capítulo 2 e reforçado por Pereira et al. (2017), a utilização de um protótipo durante o processo de desenvolvimento é fundamental, pois é através dele que se materializa de forma concreta a ideia inicial de como ficará o *design* do projeto que será desenvolvido. Dessa forma, o protótipo deste projeto foi criado seguindo a estrutura das histórias e requisitos que foram definidos durante o processo de escrita e levantamento de requisitos das histórias de usuário.

Para criar o protótipo, foi utilizada a plataforma chamada *Figma*¹. Essa plataforma possui diversas ferramentas que podem ser utilizadas na criação de interfaces para sistemas de diferentes formas, de maneira prática e intuitiva. A plataforma também é bastante conhecida e utilizada pelos designers de interfaces.

Na Figura 3, é exibida a tela inicial da aplicação, onde podem ser vistos os tipos de postagens definidos e algumas funcionalidades do sistema. Caso o leitor tenha interesse em visualizar todas as telas da aplicação, elas podem ser encontradas no Apêndice B. Vale ressaltar que dentro deste documento estarão apenas as telas e funcionalidades de maior relevância para o trabalho, mas as demais telas poderão ser vistas acessando este endereço que direciona ao protótipo desenvolvido, onde o leitor pode visualizar o protótipo completo com todas as partes mais importantes em destaque da aplicação.

¹ <https://figma.com>

Figura 3 – Tela inicial da aplicação



Fonte: Elaborado pelo autor (2023)

3.2 PROJETO

3.2.1 Arquitetura do sistema

A arquitetura adotada nesta aplicação é a Arquitetura Modular Baseada em *API* com Serviços Distribuídos, que se destaca pela sua capacidade de escalar e se adaptar às crescentes demandas de sistemas complexos. Essa abordagem permite que a *API* funcione como o núcleo central, gerenciando eficientemente as interações entre as aplicações web e mobile, enquanto mantém um bom desacoplamento entre os componentes do sistema.

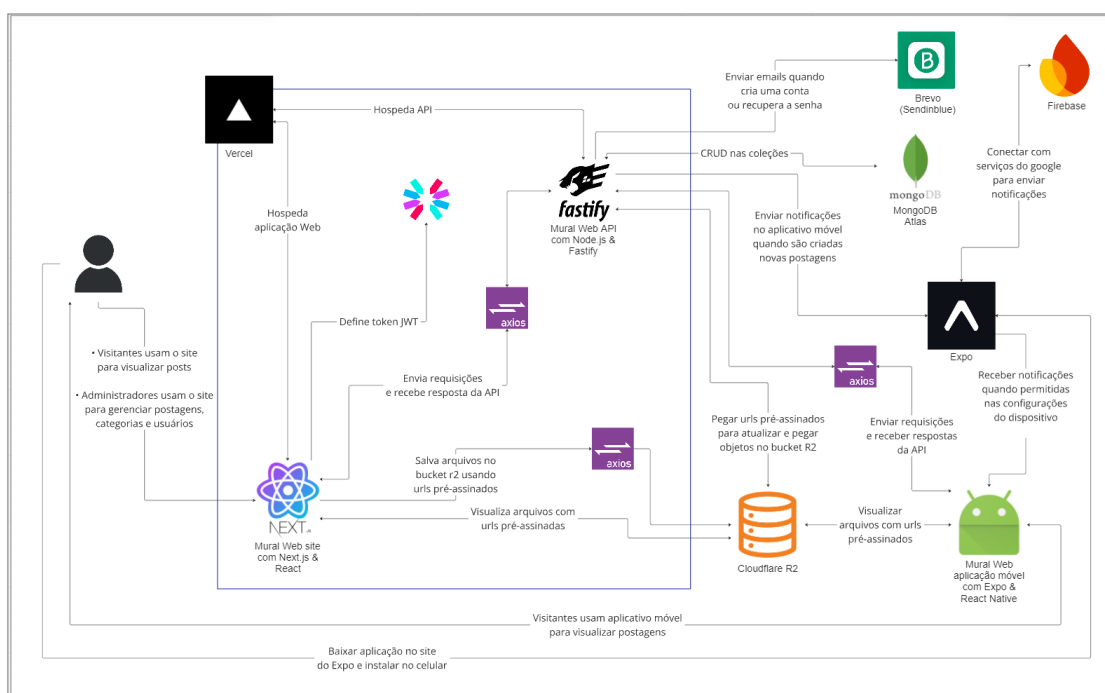
O motivo da escolha dessa abordagem é superar as limitações das arquiteturas centralizadas em termos de escalabilidade, manutenibilidade, flexibilidade e um gerenciamento eficiente de dados, (SHARMA et al., 2024). Além disso, também assegura que a aplicação possa evoluir continuamente e se adaptar às necessidades dos usuários sem impactar as operações já existentes (JAYALATH; FERNANDO, 2007).

Na *API* é onde estão localizados controladores do fluxo de requisições e respostas utilizando protocolos *REST (Representational State Transfer)* que possuem as lógicas que interagem com os serviços de manipulação de banco de dados, geração de links pré-assinados para leitura e escrita de objetos em *buckets*, envio de e-mails e envio de notificações para dispositivos móveis.

Já na aplicação *Web* e *Mobile* se localizam as interfaces onde os usuários irão visualizar de forma amigável os dados consumidos pela *API* e realizar interações. As interfaces do sistema serão produzidas utilizando as tecnologias *Expo* com *React-Native* e *Next.js* com *React* durante o processo de desenvolvimento, sempre priorizando uma boa interação e boas práticas de programação.

A aplicação possui uma grande flexibilidade em modificar seus serviços utilizados, com apenas poucas linhas de alteração ou apenas com a mudança de credenciais, a aplicação pode mudar de qualquer um dos serviços utilizados para outras opções disponíveis no mercado, caso necessário. Além disso, a aplicação também pode ser escalada ou mantida com facilidade, conforme Sharma et al. (2024) descreve sobre esta arquitetura. A figura 4 destaca a representação do modelo de arquitetura utilizado.

Figura 4 – Diagrama da Arquitetura Modular Baseada em API com Serviços Distribuídos



3.2.2 Tecnologias utilizadas

Nesta seção serão abordados os detalhes sobre as tecnologias que serão utilizadas na construção e desenvolvimento deste projeto.

O *Node.js*² é a tecnologia com motor *v8* para executar *Javascript* fora de ambientes de navegador e será utilizada em todos os ambientes da aplicação. O *Node.js* é uma tecnologia fundamental para o projeto, que permite que todas as tecnologias e módulos descritos funcionem de maneira correta. A versão do *Node.js* que está sendo utilizada atualmente é a *v20.16.0*.

O projeto está sendo gerenciado em forma de mono-repositório, que significa que o repositório contém vários projetos juntos que podem compartilhar pacotes entre si. Esse mono-repositório é gerenciado pela ferramenta de construção chamada *Turborepo*³ utilizada para ter um gerenciamento mais eficiente do repositório, junto com o gerenciador de pacotes *Pnpm*⁴, utilizado para otimizar o disco da máquina de desenvolvimento. O *Turborepo* está sendo utilizado apenas para desenvolvimento e a versão utilizada atualmente é a *v2.3.3*; já a versão utilizada do *Pnpm* é a *v9.15.1*.

No lado servidor que serve a *API* do sistema, a principal tecnologia que está sendo utilizada é o *Fastify*⁵, que serve para criar um servidor *HTTP* e expor as rotas da *API*. Além do *Fastify*, existem várias outras tecnologias utilizadas no servidor, entre elas estão as tecnologias de mais relevância citadas no Capítulo 2. A versão do *Fastify* que está sendo utilizada atualmente é a *v5.0.0*.

Para o lado do *website* o *framework* que está sendo utilizado é o *Next.js*⁶ em conjunto com a tecnologia *React*⁷. O *React* é uma biblioteca *JavaScript* de código aberto para construir interfaces de usuário interativas e reativas baseadas em componentes reutilizáveis. O *React* trabalha no lado do cliente, permitindo que os desenvolvedores criem aplicativos para web rápidos e interativos. Já o *Next.js* conta com recursos adicionais para o *React*, como a criação de aplicativos web com renderizações do lado servidor, que possibilita ao desenvolvedor criar aplicações web *full-stack* com tudo o que ele precisa. Além dessas tecnologias principais, existem outras para estilização, interfaces, requisições e entre outras que estão sendo utilizadas para montar o escopo da aplicação web. A versão utilizada do *Next.js* é a *v14.2.21*, já a versão utilizada do *React* é a *v18.3.1*.

² <https://nodejs.org/>

³ <https://turbo.build>

⁴ (<https://pnpm.io>

⁵ <https://fastify.dev>

⁶ <https://nextjs.org/>

⁷ <https://react.dev/>

E para o lado *mobile* está sendo utilizado o framework *Expo*⁸ que utiliza a tecnologia *React-Native*⁹ para criar aplicações móveis. O *React-Native* é utilizado para desenvolver aplicativos nativos utilizando o *Javascript*, baseados em *React*. Ele permite que os desenvolvedores compartilhem uma base de código entre plataformas *iOS* e *Android*, o que facilita o desenvolvimento multiplataforma. No *React-Native*, assim como no *React*, podem ser criadas interfaces de usuário nativas com componentes personalizados. Já o *Expo* propõe um ferramental que abstrai toda a complicação da configuração e da construção de aplicações utilizando *React-Native*, além de possuir vários outros recursos adicionais. Além dessas tecnologias, foram utilizadas outras para trabalhar com a estilização, interfaces e requisições, e isso monta todo o escopo da aplicação *mobile*. A versão utilizada do *React-Native* é a v0.76.5, enquanto a versão utilizada do *Expo* é a v52.0.23.

Durante o desenvolvimento do projeto, para facilitar futuras manutenções e aplicar técnicas de código limpo, foi utilizada a biblioteca *Eslint*¹⁰ para análises de estética e identificação de problemas no código-fonte baseados em regras customizáveis, e também a biblioteca *Prettier*¹¹ para padronizar o código-fonte com estilos personalizados de formatação.

Além disso, para reduzir falhas e problemas com as compilações, foram aplicadas técnicas de testes *end-to-end* dentro do projeto da *API* e da plataforma web com o uso das bibliotecas *Vitest*¹² para testes do lado servidor no *API* e *Cypress*¹³ para testes do lado cliente no *website*.

3.3 FUNCIONAMENTO DA APLICAÇÃO

O aplicativo móvel já foi completamente construído e está disponível para ser baixado no seguinte endereço. Ele foi projetado para funcionar nas versões mais atuais dos sistemas *Android* sem travamentos ou uso excessivo de recursos do dispositivo.

A plataforma web também já está totalmente implantada em produção no ambiente de hospedagem da *Vercel*¹⁴, que foi destacada na arquitetura do sistema. A plataforma foi desenvolvida para funcionar de maneira fluida em diversos navegadores e dispositivos, podendo ser acessada através deste endereço.

⁸ <https://expo.dev>

⁹ <https://reactnative.dev>

¹⁰ <https://eslint.org/>

¹¹ <https://prettier.io/>

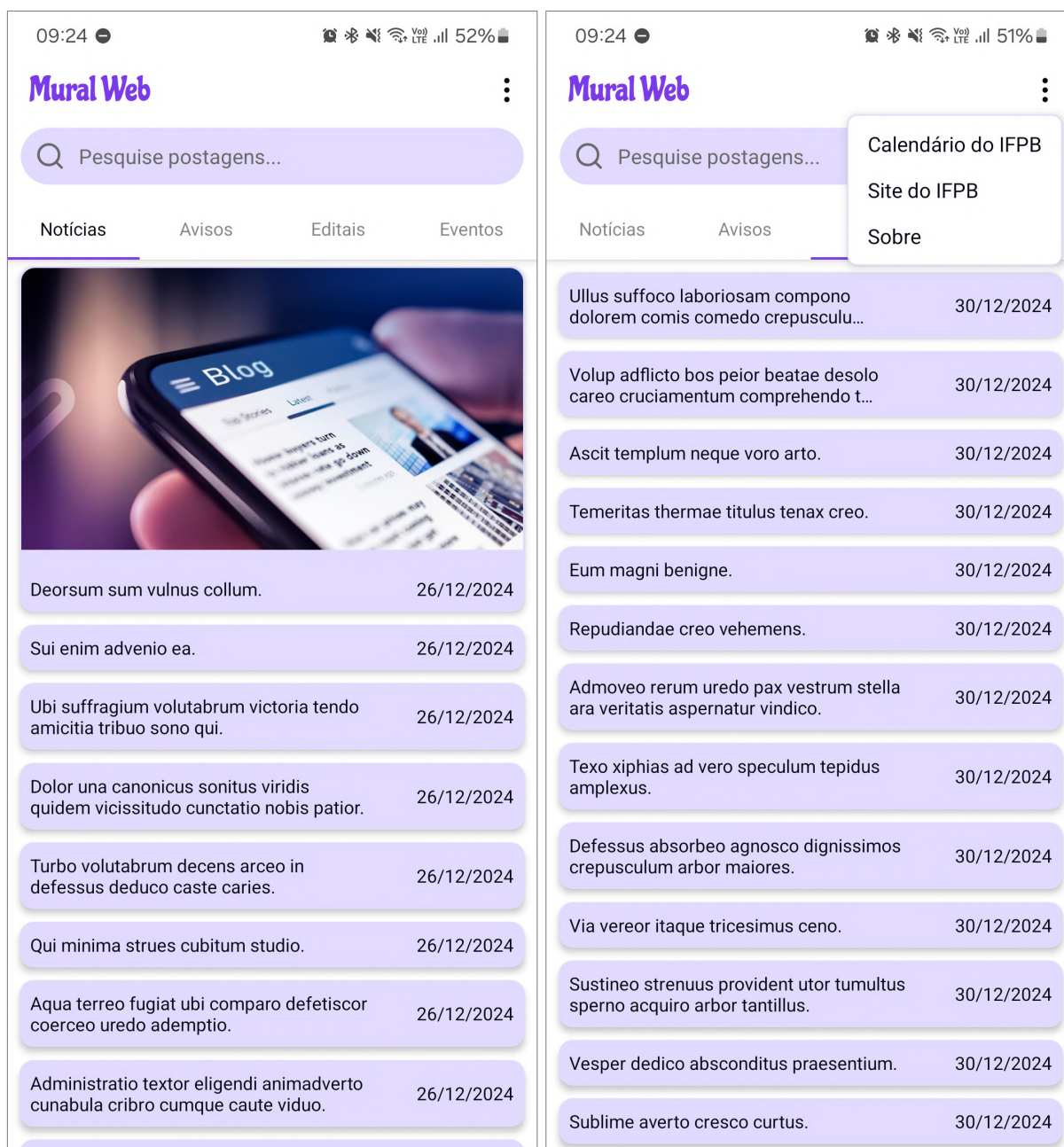
¹² <https://vitest.dev/>

¹³ <https://cypress.io>

¹⁴ <https://vercel.com>

A Figura 5 apresenta a tela inicial do aplicativo móvel onde são mostradas em uma interface amigável as postagens separadas por categorias e filtradas da mais recente à mais antiga, além da barra de busca por postagens e o menu de opções da tela inicial.

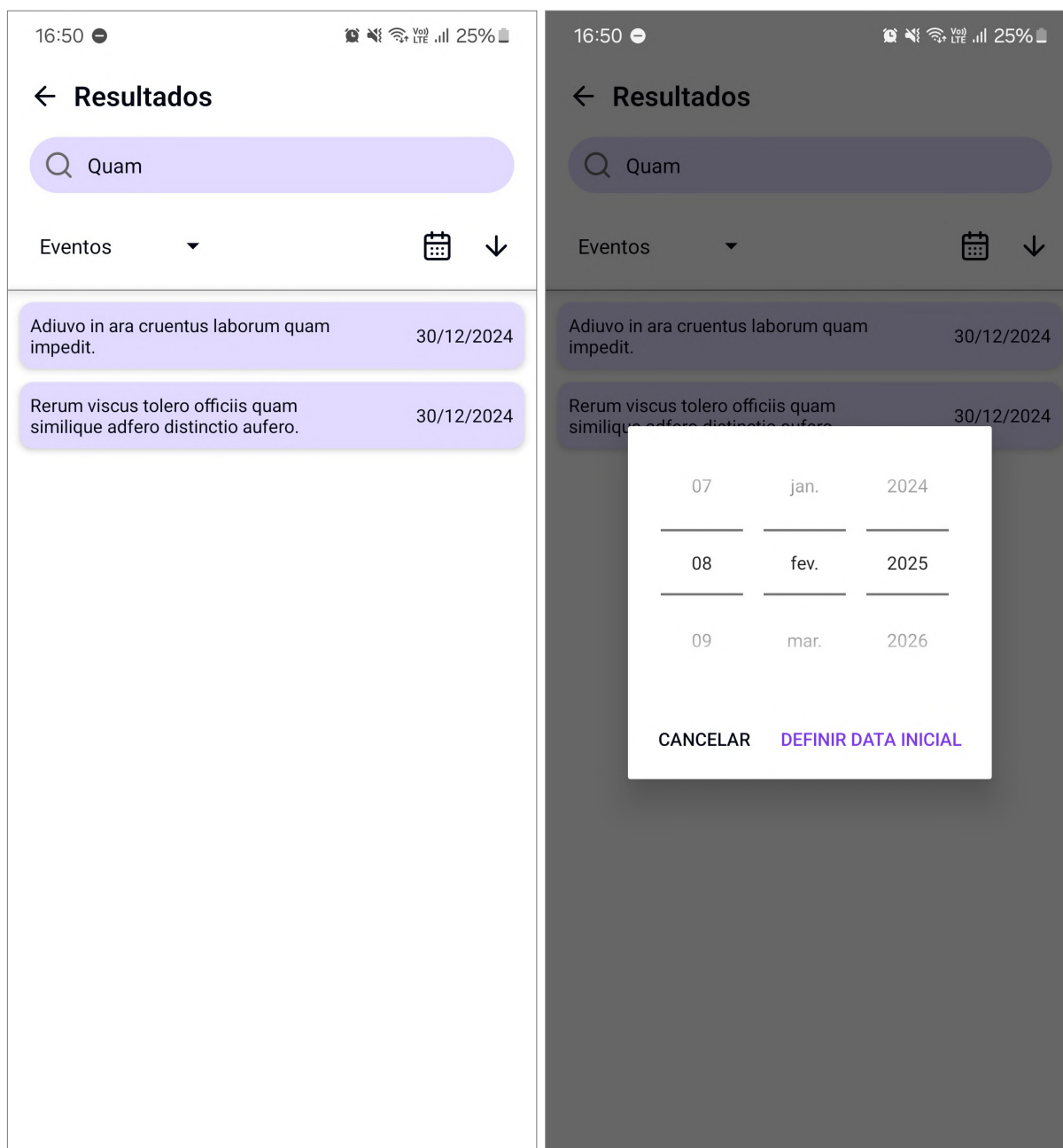
Figura 5 – Captura da tela inicial do aplicativo móvel exibindo as postagens separadas por categorias e o menu de opções



Fonte: Elaborado pelo autor (2025)

No aplicativo móvel foi implementada a barra de busca para encontrar postagens mais facilmente. A Figura 6 destaca um exemplo de utilização da barra de busca com os resultados obtidos através da busca pelo título das postagens, além das opções de filtragem por categorias, limites de datas e ordenação crescente e decrescente.

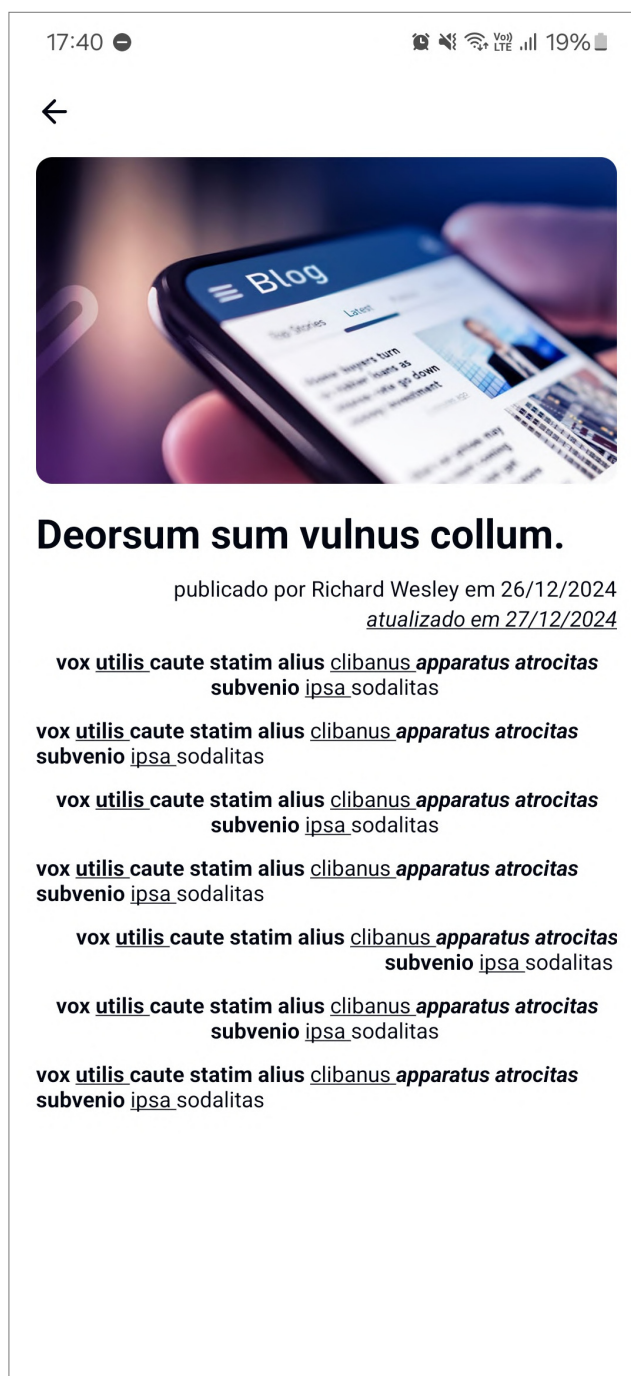
Figura 6 – Captura da tela de busca do aplicativo móvel exibindo os resultados da busca e as opções de filtragem



Fonte: Elaborado pelo autor (2025)

A Figura 7 demonstra um exemplo de como é exibida uma postagem ao acessá-la a partir da tela inicial ou dos resultados de busca e filtragem da aplicação. A postagem pode exibir imagem de destaque, título, autor, data de criação, data de última atualização e o corpo da postagem que pode incluir formatações de texto e *links* úteis.

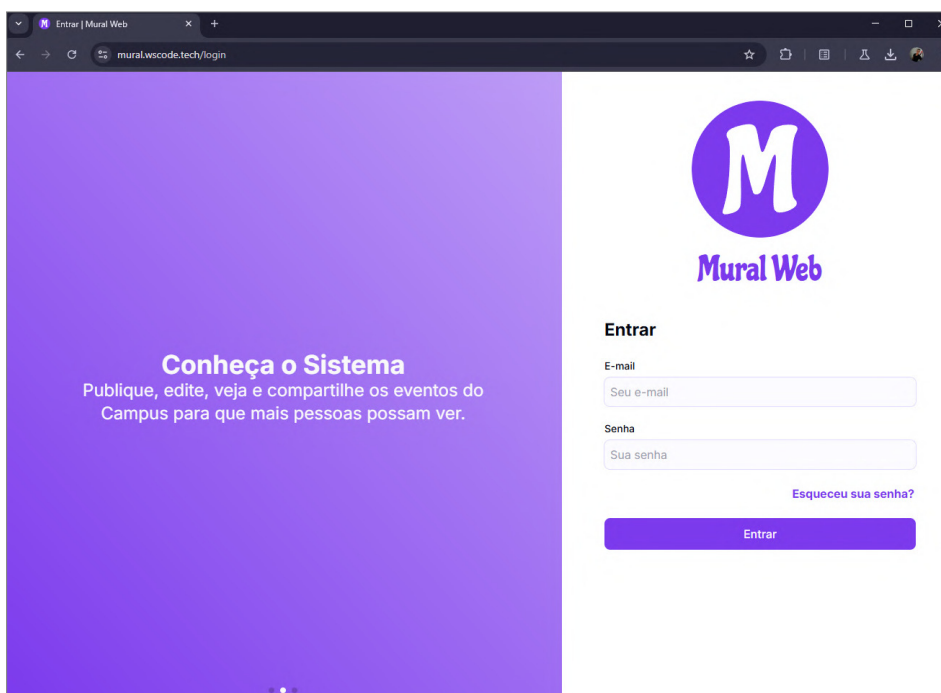
Figura 7 – Captura da tela de exibição de uma postagem no aplicativo móvel



Fonte: Elaborado pelo autor (2025)

Para acessar o painel administrativo, o usuário precisa entrar na sua conta na plataforma web. A Figura 8 demonstra como é a página de acesso à conta do usuário na plataforma web utilizando o e-mail e a senha de uma conta que foi criada previamente por um administrador da plataforma.

Figura 8 – Captura da página de login da plataforma web



Fonte: Elaborado pelo autor (2025)

Dentro do painel administrativo existem três seções que separam as funcionalidades principais do painel, e cada uma seção possui a sua barra de busca, exibição de itens, paginação e ações por elemento, os quais incluem: criar, visualizar, editar, deletar, alterar disponibilidade e cargos. Essas seções principais são: as postagens que podem ser vistas na Figura 9; as categorias que podem ser vistas na Figura 10; e os usuários administradores/moderadores que podem ser vistos na Figura 11.

Figura 9 – Captura de tela da seção de postagens do painel administrativo da plataforma web

Postagem	Autor	Categoria	Status	Criado em	Atualizado em
<input type="checkbox"/> test 6797dfb239338b30e9d701f1	Richard Wesley 676d3b4d8e9cec620438a41f	Avisos 676d3b4e8e9cec620438a420	publica	27/01/2025	27/01/2025
<input type="checkbox"/> teste 6797dfa238338b30e9d701f0	Richard Wesley 676d3b4d8e9cec620438a41f	Avisos 676d3b4e8e9cec620438a420	publica	27/01/2025	27/01/2025
<input type="checkbox"/> testing post 16/01, update 27/01 6789e594c38c3059c2cbafca	Richard Wesley 676d3b4d8e9cec620438a41f	Avisos 676d3b4e8e9cec620438a420	publica	16/01/2025	27/01/2025
<input type="checkbox"/> Feliz natal 67749c67a5a327c086058d2c	Richard Wesley 676d3b4d8e9cec620438a41f	Avisos 676d3b4e8e9cec620438a420	publica	31/12/2024	31/12/2024
<input type="checkbox"/> Novas formas de publicar 67743dc8f55eaadd523a59c5	Richard Wesley 676d3b4d8e9cec620438a41f	Avisos 676d3b4e8e9cec620438a420	publica	31/12/2024	31/12/2024
<input type="checkbox"/> Testando uma postagem 67743d8f876327aaf4a9e03	Richard Wesley 676d3b4d8e9cec620438a41f	Avisos 676d3b4e8e9cec620438a420	publica	31/12/2024	31/12/2024
<input type="checkbox"/> Sonitus barba tricesimus libero alioqui tabella. 67730fefe8b82045307b5751	Richard Wesley 676d3b4d8e9cec620438a41f	Avisos 676d3b4e8e9cec620438a420	publica	30/12/2024	30/12/2024
<input type="checkbox"/> Defaeco distinctio laboriosam suadeo. 67730fefe8b82045307b576b	Richard Wesley 676d3b4d8e9cec620438a41f	Eventos 676d3b4e8e9cec620438a422	publica	30/12/2024	30/12/2024
<input type="checkbox"/> Cultura abbas totidem utrimque	Richard Wesley	Eventos	publica		

Mostrando 10 de 244 itens

Linhas por página: 10

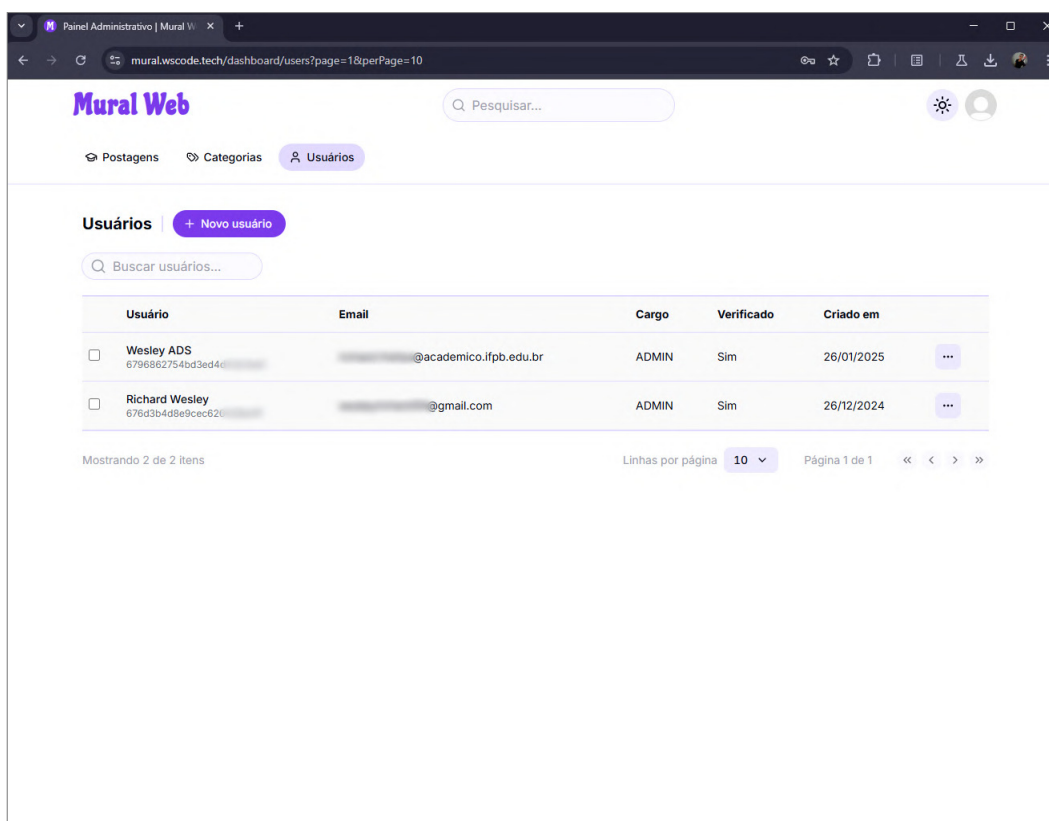
Página 1 de 25

Fonte: Elaborado pelo autor (2025)

Figura 10 – Captura de tela da seção de categorias do painel administrativo da plataforma web

Categoria	Slug	Postagens	Criado em	Ações
<input type="checkbox"/> Notícias 676d3b4e8e9cec620438a423	noticias	60	26/12/2024	...
<input type="checkbox"/> Avisos 676d3b4e8e9cec620438a420	aviso	64	26/12/2024	...
<input type="checkbox"/> Editais 676d3b4e8e9cec620438a421	editais	60	26/12/2024	...
<input type="checkbox"/> Eventos 676d3b4e8e9cec620438a422	eventos	60	26/12/2024	...

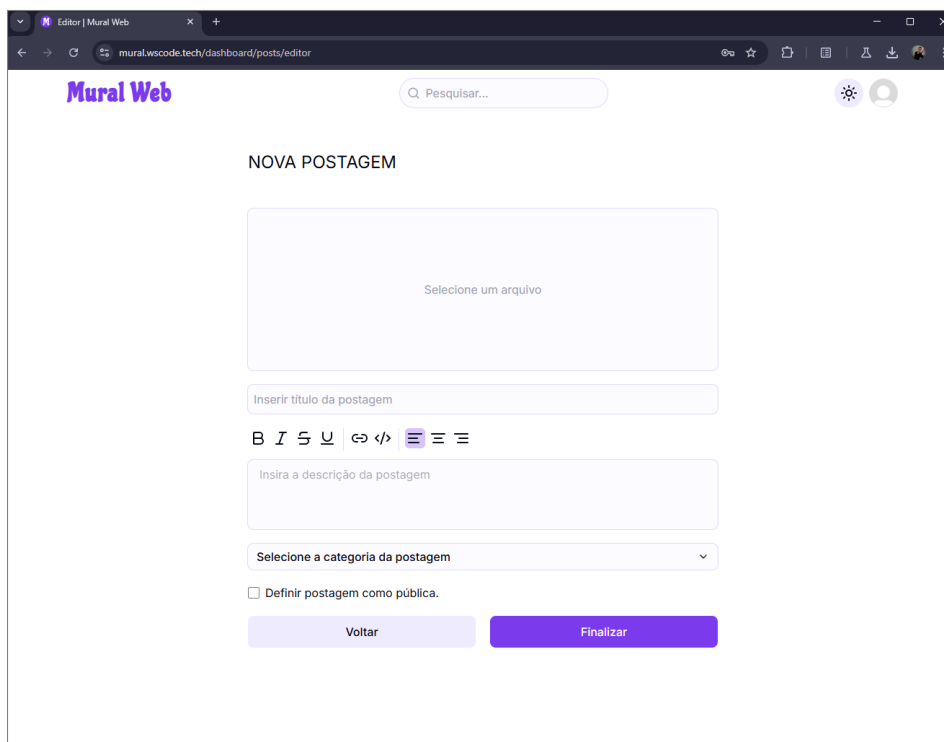
Fonte: Elaborado pelo autor (2025)

Figura 11 – Captura de tela da seção de usuários do painel administrativo da plataforma web

Fonte: Elaborado pelo autor (2025)

Para criar ou editar uma postagem, são necessários diversos atributos, como título, categoria, arquivos de imagem da postagem, edição de texto do corpo da descrição e definição de disponibilidade. A Figura 12 mostra como é a página de criação de postagens, enquanto a Figura 13 mostra como é que fica o formulário preenchido com os dados de uma postagem já cadastrada que pode ser editada.

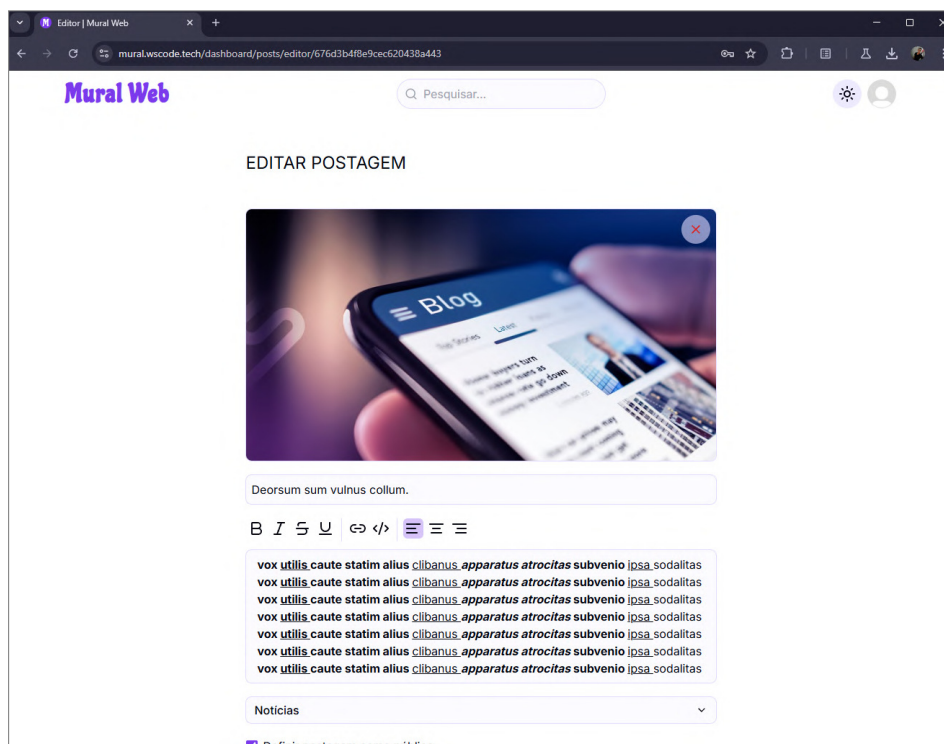
Figura 12 – Captura de tela do formulário de criação de postagens do painel administrativo da plataforma web



The screenshot shows a web browser window with the URL `mural.wscodetech.com/dashboard/posts/editor`. The page title is "Mural Web" and the main heading is "NOVA POSTAGEM". The form includes a large text area for image selection with the placeholder "Selecione um arquivo". Below this is a text input for "Inserir título da postagem". A rich text editor toolbar with icons for bold, italic, underline, link, unlink, code, and list is present. The main text area contains the placeholder "Insira a descrição da postagem". A dropdown menu for "Selecione a categoria da postagem" is set to "Notícias". A checkbox for "Definir postagem como pública." is unchecked. At the bottom are "Voltar" and "Finalizar" buttons.

Fonte: Elaborado pelo autor (2025)

Figura 13 – Captura de tela do formulário de edição de postagens do painel administrativo da plataforma web



The screenshot shows a web browser window with the URL `mural.wscodetech.com/dashboard/posts/editor/676d3b4f8e9cec620438a443`. The page title is "Mural Web" and the main heading is "EDITAR POSTAGEM". A preview image of a smartphone displaying a blog post is shown at the top. Below the image is a text input containing the placeholder "Deorsum sum vulnus collum.". A rich text editor toolbar is present. The main text area contains the placeholder "vox utilis, caute statim alius cilbanus, apparatus atrocitas subvenio ipsa sodalitas". A dropdown menu for "Selecione a categoria da postagem" is set to "Notícias". A checkbox for "Definir postagem como pública." is checked.

Fonte: Elaborado pelo autor (2025)

O processo de cadastro ou edição de categorias ou usuários é mais simples e menos complexo, não sendo necessário criar uma nova página para essa funcionalidade. Para criar/editar uma categoria, só é necessário o nome dela, e já para criar um usuário, é necessário o nome do usuário, o e-mail que receberá o acesso à plataforma e o cargo do usuário, que pode ser alterado a qualquer momento. Essas funcionalidades descritas só podem ser feitas por um administrador da plataforma. Demais cargos não possuem permissão para isso.

Dentre outras funcionalidades, existe também a área de recuperação da conta perdida que pode ser acessada a partir da página inicial da plataforma web, a área do perfil do usuário que pode ser introduzida através do menu principal disponível em todas as páginas após um usuário estar conectado à sua conta, o que possibilita a mudança do seu nome, e-mail ou senha cadastrados.

Além dessas funcionalidades descritas acima, a plataforma web ainda proporciona, sem o uso de credenciais, a possibilidade de visualizar uma página inicial com as postagens publicadas, uma aba de busca por postagens e a visualização dessas postagens.

E além disso, a aplicação móvel ainda conta com o sistema de notificações, que, ao permiti-las, o usuário está apto a ser notificado à medida que novas postagens vão sendo publicadas, sem a necessidade de ficar acessando o aplicativo ou a plataforma para ver se existe algo novo.

Algumas capturas de telas dessas funcionalidades não apresentadas no texto são mostradas no Apêndice B.

4 CONSIDERAÇÕES FINAIS

A análise realizada ao longo deste trabalho evidenciou deficiências na comunicação e na usabilidade no site da instituição, apontando para a necessidade de soluções que melhorem o acesso às informações acadêmicas. O desenvolvimento da aplicação Mural Web surge como uma alternativa viável, implementada com as melhores práticas para promover a interatividade, eficiência e usabilidade na disseminação dessas informações.

A implementação do Mural Web na instituição requer uma avaliação cuidadosa dos recursos tecnológicos necessários, incluindo a estimativa de custos associados ao uso dos serviços de envio de e-mails, banco de dados, armazenamento de objetos, hospedagem, e inclusive eventuais manutenções que são necessárias para a utilização da aplicação em larga escala. É essencial que a gestão do IFPB considere a relação custo-benefício dessa aplicação, levando em conta o impacto positivo esperado na comunicação entre a instituição e sua comunidade acadêmica.

Para trabalhos futuros, é recomendado explorar a integração do Mural Web com outras plataformas e sistemas existentes no IFPB, permitindo uma troca de dados mais fluida e abrangente. Além disso, futuras versões da aplicação podem incluir funcionalidades como um sistema de *feedback* contínuo para avaliação das postagens, e também um sistema de recomendações de conteúdos, que pode aumentar ainda mais o engajamento dos usuários. A manutenção e atualização regular das tecnologias utilizadas também serão cruciais para reciclar e garantir a longevidade e eficácia da aplicação.

Essas ações contribuirão não apenas para a melhoria da comunicação acadêmica, mas também para o aprimoramento contínuo da experiência do usuário, estabelecendo um modelo que pode ser replicado em outras instituições de ensino, possibilitando um maior alcance e impacto dentro da comunidade acadêmica.

A tecnologia possui um papel significativo na transformação da educação, e a evolução contínua do Mural Web tem o potencial de impactar positivamente a comunidade acadêmica, contribuindo para melhorias dentro dos processos de ensino e aprendizagem.

REFERÊNCIAS

CHAITANYA, B. S. S. K.; REDDY, D. A. K.; CHANDRA, B. P. S. E.; KRISHNA, A. B.; MENON, R. R. K. Full-text search using database index. In: **2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)**. [S.l.: s.n.], 2019. p. 1–5.

CHODOROW, K.; BRADSHAW, S.; BRAZIL, E.; SAFARI, a. O. M. C. **MongoDB: The Definitive Guide, 3rd Edition**. O'Reilly Media, Incorporated, 2018. ISBN 9781491954454. Disponível em: <<https://books.google.com.br/books?id=VS-2tAEACAAJ>>.

CLIFTON, C.; GARCIE-MOLINA, H. The design of a document database. In: **Proceedings of the ACM conference on Document processing systems**. [S.l.: s.n.], 2000. p. 125–134.

CLOUD, G. **O que é NoSQL e como ela funciona? | Google Cloud**. 2023. Disponível em: <<https://cloud.google.com/discover/what-is-nosql>>. Acesso em: 16 de outubro de 2023.

CLOUDFLARE. **Cloudflare R2**. 2024. Disponível em: <<https://www.cloudflare.com/developer-platform/products/r2>>. Acesso em: 09 de janeiro de 2024.

COHN, M. **User Stories Applied: For Agile Software Development**. Addison-Wesley, 2004. (Addison-Wesley signature series). ISBN 9780321205681. Disponível em: <<https://books.google.com.br/books?id=SvlwuX4SVigC>>.

DELMAZO, C.; VALENTE, J. C. L. Fake news nas redes sociais online: propagação e reações à desinformação em busca de cliques. **Media & Jornalismo**, v. 18, n. 32, p. 155–169, Mai. 2018. Disponível em: <https://impactum-journals.uc.pt/mj/article/view/2183-5462_32_11>.

FOTOPOULOS, G.; KOLOVEAS, P.; RAFTOPOULOU, P.; TRYFONOPOULOS, C. Comparing data store performance for full-text search: to sql or to nosql? In: . [S.l.: s.n.], 2023.

GARCIA, L. S. **A preferência dos estudantes de nível superior na adoção de redes sociais em comparação com os canais institucionais**. 2014. Disponível em: <<http://hdl.handle.net/10438/11649>>.

JAYALATH, C.; FERNANDO, R. A modular architecture for secure and reliable distributed communication. In: IEEE. **The Second International Conference on Availability, Reliability and Security (ARES'07)**. [S.l.], 2007. p. 621–628.

JUNIOR, J. R. F. **Índices em MongoDB**. 2019. Disponível em: <<https://www.linkedin.com/pulse/índices-em-mongodb-jose-r-f-junior/?originalSubdomain=pt>>. Acesso em: 20 de novembro de 2023.

JUNIOR, M. M. V. Interação homem computador: Análise da interface e usabilidade do sicoobnet, o internet banking do banco cooperativo do brasil sa. 2017. Disponível em: <<http://hdl.handle.net/123456789/401>>.

LONGO, H. E. R.; SILVA, M. P. A utilização de histórias de usuários no levantamento de requisitos Ágeis para o desenvolvimento de software. 2014. Disponível em: <<https://doi.org/10.47916/ijkem-vol3n6-2014-1>>.

LÓSCIO, B. F.; OLIVEIRA, H. d.; PONTES, J. d. S. Nosql no desenvolvimento de aplicações web colaborativas. **VIII Simpósio Brasileiro de Sistemas Colaborativos**, sn, v. 10, n. 1, p. 11, 2011.

MABUNDA, V. dos S.; SILVA, F. M. da. Sistema de autenticação centralizada e single-sign on unificado baseado no central authentication service para serviços web no isutc. 2017.

MATOS, C. M. d.; SILVA, J. M. C.; MATIAS, M. Análise de usabilidade nas homepages das instituições de ensino superior de montes claros/mg. **Informação & Informação**, v. 26, n. 3, p. 651–673, out. 2021. Disponível em: <<https://ojs.uel.br/revistas/uel/index.php/informacao/article/view/40642>>.

MONGODB. **Document Database - NoSQL | MongoDB**. 2023. Disponível em: <<https://www.mongodb.com/document-databases>>. Acesso em: 15 de outubro de 2023.

NBR. **NBR 9241-11 - Requisitos Ergonômicos para trabalho de escritórios com computadores**. 2002. Disponível em: <http://www.inf.ufsc.br/~jedla/ine5624/_Walter/Normas/Parte%2011/iso9241-11F2.pdf>. Acesso em: 01 de fevereiro de 2025.

NODEMAILER. **Nodemailer**. 2024. Disponível em: <<https://www.nodemailer.com>>. Acesso em: 10 de janeiro de 2024.

NUNES, R. D. A implantação das metodologias ágeis de desenvolvimento de software scrum e extreme programming(xp): uma alternativa para pequenas empresas do setor de tecnologia da informação. **ForScience**, v. 4, n. 2, fev. 2017. Disponível em: <<https://forscience.ifmg.edu.br/index.php/forscience/article/view/117>>.

PEREIRA, D.; LANUTTI, J.; PASCHOARELLI, L. C.; PINHEIRO, O. Comparison of traditional manual prototyping techniques and their importance to design. **DAT Journal**, v. 2, n. 2, Dec. 2017. Disponível em: <<https://datjournal.emnuvens.com.br/dat/article/view/62>>.

PRESSMAN, R.; MAXIM, B. **Engenharia De Software: UMA ABORDAGEM PROFISSIONAL**. MCGRAW HILL - ARTMED, 2016. ISBN 9788580555332. Disponível em: <<https://books.google.com.br/books?id=smNFvgAACAAJ>>.

RIADH, M. H. Notification system to students using an android application. **Int. J. Comput. Appl**, v. 140, n. 1, p. 22–27, 2016.

SHARMA, P.; REIMAN, A. P.; ANDERSON, A. A.; POUDEL, S.; ALLWARDT, C. H.; FISHER, A. R.; SLAY, T. E.; MUKHERJEE, M.; DUBEY, A.; OGLE, J. P.; MELTON, R. B. Gridapps-d distributed app architecture and api for modular and distributed grid operations. **IEEE Access**, v. 12, p. 39862–39875, 2024.

SILVA, D. K. M. da; SOBRAL, N. V.; SOARES, L. V. H.; DIGITAIS, M. Acessibilidade em sites de universidades federais de pernambuco: avaliação à luz do modelo de

acessibilidade em governo eletrônico—emag accesibility on pernambuco federal universities sites: evaluation under the electronic government accessibility model-emag. 2020. Disponível em: <<https://doi.org/10.52614/2317-0115.2016.244884>>.

SOUSA, P. de. **5 leis que todos os UX Designers devem saber - Virgu Portugal**. 2020. Disponível em: <<https://virgu.com/blog/5-leis-que-todos-os-ux-designers-devem-saber/>>. Acesso em: 07 de novembro de 2023.

SOUZA, E. C.; OLIVEIRA, M. R. d. Comparativo entre os bancos de dados mysql e mongodb: quando o mongodb é indicado para o desenvolvimento de uma aplicação. **Revista Interface Tecnológica**, v. 16, n. 2, p. 38–48, dez. 2019. Disponível em: <<https://revista.fatectq.edu.br/interfacetecnologica/article/view/664>>.

TERRA, C.; DREYER, B.; RAPOSO, J. **Comunicação organizacional: Práticas, desafios e perspectivas digitais**. [S.l.]: Summus Editorial, 2021. 12 p. ISBN 9786555490381.

YABLONSKI, J. **Leis da Psicologia Aplicadas a UX: Usando psicologia para projetar produtos e serviços melhores**. [S.l.]: Novatec Editora, 2020. 35 p. ISBN 9786586057263.

ZHAO, H. **Managing Vendor Lock-in in Serverless Edge-to-Cloud Computing from the Client Side**. Dissertação (Mestrado) — Technical University of Berlin, dez. 2022. Disponível em: <<https://inria.hal.science/hal-03946722>>.

APÊNDICE A – HISTÓRIAS DE USUÁRIO

Quadro 1 – User story da criação de conta do usuário

US 01 - Criar conta	
Como um administrador autenticado, eu quero criar uma conta para um usuário.	
Story Points	8 Prioridade 1
TA1.1	O administrador deve conseguir acessar a área de criação de conta na página do painel.
TA1.2	O administrador deve inserir um endereço de e-mail válido para o usuário.
TA1.3	O administrador deve inserir o nome do usuário a ser criado.
TA1.4	O sistema deve criar uma senha com pelo menos 6 caracteres para o usuário.
TA1.5	O sistema deve verificar se o endereço de e-mail não foi registrado anteriormente.
TA1.6	O sistema deve fornecer mensagens de erro claras se os detalhes de registro das credenciais forem inválidos.
TA1.7	O sistema deve enviar um link de confirmação de conta para o e-mail do usuário com seu acesso.
TA1.8	O usuário deve conseguir acessar sua conta após a confirmação da conta.

Fonte: Elaborado pelo autor (2024)

Quadro 2 – User story de acesso a conta do usuário

US 02 - Acessar conta	
Como usuário cadastrado, quero acessar minha conta.	
Story Points	5 Prioridade 1
TA2.1	O usuário deve conseguir se autenticar na sua conta.
TA2.2	O usuário deve inserir seu endereço de e-mail e senha registrados.
TA2.3	O sistema deve validar o formato do endereço de e-mail e da senha.
TA2.4	O sistema deve autenticar o usuário se as credenciais forem válidas.
TA2.5	O usuário deve conseguir acessar o sistema se o login for bem sucedido.
TA2.6	O sistema deve manter o usuário autenticado e conectado por um certo período de tempo.
TA2.7	O sistema deve fornecer mensagens de erro claras se as credenciais de login forem inválidas ou incorretas.

Fonte: Elaborado pelo autor (2023)

Quadro 3 – User story de recuperação de conta do usuário

US 03 - Recuperar conta			
Como usuário cadastrado, desejo recuperar o acesso à minha conta cadastrada.			
Story Points	5	Prioridade	2
TA3.1	O usuário deve ser capaz de acessar a área de recuperação da conta a partir da área de login.		
TA3.2	O usuário deve inserir um e-mail válido.		
TA3.3	O sistema deve verificar se o e-mail inserido é válido e se pertence a alguma conta cadastrada.		
TA3.4	O sistema deve enviar um link de recuperação de senha para o e-mail inserido se o endereço de e-mail existir e pertencer a alguma conta.		
TA3.5	O link enviado deve abrir uma página onde o usuário possa redefinir sua senha.		
TA3.6	O usuário deve inserir uma senha que contenha ao menos 6 caracteres.		
TA3.7	O usuário deve ser capaz de acessar sua conta novamente ao recuperar a senha perdida.		
TA3.8	O sistema deve fornecer mensagens de erro claras caso o e-mail inserido não atender aos requisitos.		

Fonte: Elaborado pelo autor (2023)

Quadro 4 – User story de edição do perfil do usuário

US 04 - Editar perfil			
Como usuário cadastrado, desejo alterar as informações da minha conta.			
Story Points	13	Prioridade	2
TA4.1	O usuário deve poder alterar a foto do perfil da conta na área de perfil.		
TA4.2	O usuário deve poder editar seu nome de conta, e-mail e senha na área de perfil.		
TA4.3	O usuário deve inserir informações válidas ao editar o nome, e-mail ou senha.		
TA4.4	O sistema deve verificar se ao inserir um novo endereço de e-mail, ele não foi registrado anteriormente.		
TA4.5	O sistema deve verificar se ao digitar uma nova senha, ela tenha pelo menos 6 caracteres.		
TA4.6	O sistema deve confirmar a senha do usuário se a senha atual for alterada.		
TA4.7	O sistema deve enviar um link de confirmação para o novo e-mail se o endereço de e-mail atual for alterado.		
TA4.8	O sistema deve fornecer mensagens de erro claras se as credenciais inseridas forem inválidas.		
TA4.9	O usuário deve poder visualizar novas informações na conta ao salvar alterações.		
TA4.10	O usuário deve conseguir sair da conta através do menu.		

Fonte: Elaborado pelo autor (2023)

Quadro 5 – User story de visualização de postagens

US 05 - Visualizar postagens			
Como usuário, desejo visualizar as postagens disponíveis.			
Story Points	8	Prioridade	1
TA5.1	O usuário deve ser capaz de visualizar todas as postagens disponíveis na área principal do sistema.		
TA5.2	O usuário deve ser capaz de expandir todas as informações de uma postagem clicando ou pressionando-a.		
TA5.3	O usuário administrador deve conseguir visualizar todas as postagens na página do painel, incluindo postagens privadas.		
TA5.4	O sistema deve separar as postagens por categorias.		

Fonte: Elaborado pelo autor (2023)

Quadro 6 – User story de visualização de links úteis

US 06 - Visualizar links úteis			
Como usuário, desejo visualizar links úteis.			
Story Points	1	Prioridade	3
TA6.1	O usuário deve ser capaz de visualizar os links úteis na área de mais opções.		
TA6.2	O sistema deve redirecionar o usuário abrindo as devidas páginas no navegador da web ao pressionar ou clicar em qualquer link útil.		

Fonte: Elaborado pelo autor (2023)

Quadro 7 – User story de visualização de atualizações de postagens

US 07 - Visualizar atualizações			
Como usuário, desejo visualizar atualizações/notificações para novas postagens.			
Story Points	13	Prioridade	3
TA7.1	O sistema deve enviar notificações ao usuário sempre que houver novas postagens.		
TA7.2	O usuário deve poder visualizar novas postagens como notificações.		

Fonte: Elaborado pelo autor (2023)

Quadro 8 – User story de busca por postagens

US 08 - Buscar postagens			
Como usuário, desejo fazer uma busca por postagens.			
Story Points	8	Prioridade	2
TA8.1	O usuário deve ser capaz de pesquisar postagens pela área da barra de pesquisa.		
TA8.2	O usuário deve ser capaz de inserir textos para serem pesquisados na barra de busca.		
TA8.3	O sistema deve exibir o resultado da busca baseado no texto inserido pelo usuário.		
TA8.4	O usuário deve ser capaz de filtrar os resultados da pesquisa por períodos de tempo, ordem ou categoria específicos.		
TA8.5	O usuário deve conseguir acessar e visualizar as informações das postagens ao pressionar ou clicar nelas.		
TA8.6	O sistema deve possibilitar o usuário fazer novas buscas na área dos resultados da busca.		

Fonte: Elaborado pelo autor (2023)

Quadro 9 – User story de alteração de permissões de usuários

US 09 - Alterar permissões de usuários			
Como usuário administrador autenticado, desejo alterar as permissões dos usuários cadastrados.			
Story Points	5	Prioridade	2
TA9.1	O usuário administrador deve ser capaz de alterar as permissões dos usuários acessando a área administrativa de usuários.		
TA9.2	O sistema deve disponibilizar uma lista com todos os usuários cadastrados exibindo o e-mail e as permissões deles.		
TA9.3	O sistema deve confirmar se o administrador realmente deseja alterar a permissão do usuário.		
TA9.4	O sistema deve impossibilitar alterar a permissão do próprio usuário administrador.		
TA9.5	O usuário administrador deve ser capaz de excluir outro usuário na área administrativa.		
TA9.6	O sistema deve fornecer mensagens claras sobre os erros ou alterações.		

Fonte: Elaborado pelo autor (2023)

Quadro 10 – User story de criação de postagens

US 10 - Criar postagem			
Como usuário administrador ou provedor de conteúdo autenticado, desejo criar uma nova postagem.			
Story Points	5	Prioridade	1
TA10.1	O usuário administrador ou provedor de conteúdo deve poder acessar a área de criação de postagens na área principal do sistema.		
TA10.2	O usuário deve poder adicionar uma imagem à postagem, caso seja necessário.		
TA10.3	O usuário deve inserir um título, conteúdo, categoria e visibilidade válidos para a nova postagem.		
TA10.4	O sistema deve exibir mensagens claras de sucesso ou erro ao registrar a postagem.		
TA10.5	O sistema deve incorporar a data de criação na nova postagem.		
TA10.6	O sistema deve notificar e disponibilizar a postagem criada aos usuários caso seja uma postagem pública.		

Fonte: Elaborado pelo autor (2023)

Quadro 11 – User story de edição de postagens

US 11 - Editar postagem			
Como usuário administrador ou provedor de conteúdo autenticado, desejo editar uma postagem criada anteriormente			
Story Points	8	Prioridade	1
TA11.1	O usuário ou provedor de conteúdo deve ser capaz de acessar área de edição de uma postagem a partir da área principal do sistema.		
TA11.2	O usuário deve ser capaz de editar o título, conteúdo, categoria, visibilidade ou imagem de uma postagem cadastrada.		
TA11.3	O sistema deve exibir mensagens de sucesso ou erro claras ao tentar editar uma postagem.		
TA11.4	O sistema deve restringir o usuário provedor de conteúdo a editar somente as postagens criadas por ele.		
TA11.5	O sistema deve incorporar a data de atualização na postagem editada.		
TA11.6	O sistema deve notificar e disponibilizar a postagem atualizada aos usuários caso seja atualizada como uma postagem pública e não tenha sido uma postagem pública anteriormente.		

Fonte: Elaborado pelo autor (2023)

Quadro 12 – User story de exclusão de postagens

US 12 - Excluir postagem			
Como usuário administrador ou provedor de conteúdo autenticado, desejo excluir uma postagem criada anteriormente.			
Story Points	5	Prioridade	1
TA12.1	O usuário deve ser capaz de excluir uma postagem a partir da área principal do sistema.		
TA12.2	O sistema deve confirmar se o usuário realmente deseja excluir a postagem.		
TA12.3	O sistema deve exibir mensagens de sucesso ou erro claras ao tentar excluir uma postagem.		
TA12.4	O sistema deve restringir o usuário provedor de conteúdo a excluir somente as postagens criadas por ele.		

Fonte: Elaborado pelo autor (2023)

Quadro 13 – User story de gerenciamento de categorias

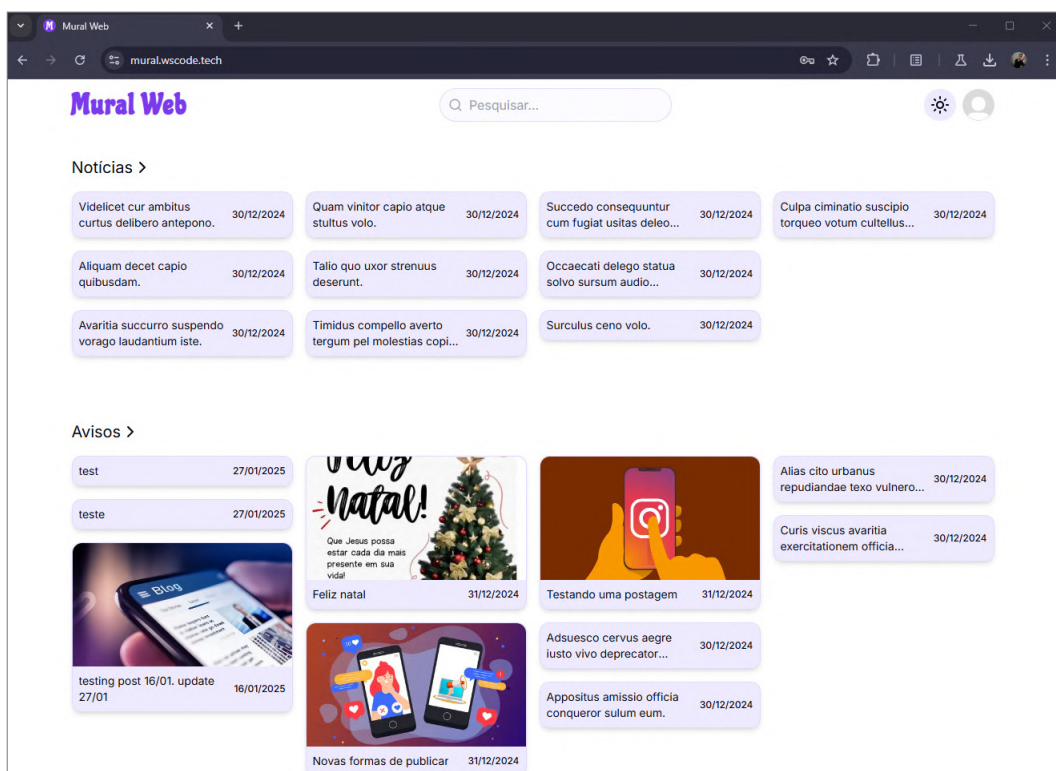
US 13 - Gerenciar categorias			
Como usuário administrador autenticado, desejo gerenciar categorias.			
Story Points	5	Prioridade	1
TA13.1	O administrador deve ser capaz de criar novas categorias.		
TA13.2	O administrador deve ser capaz de atualizar as categorias criadas.		
TA13.3	O administrador deve ser capaz de excluir categorias criadas.		
TA13.4	O sistema deve verificar se o novo nome da categoria/atualização não foi registrado anteriormente.		

Fonte: Elaborado pelo autor (2024)

APÊNDICE B – CAPTURAS DE TELA DA APLICAÇÃO

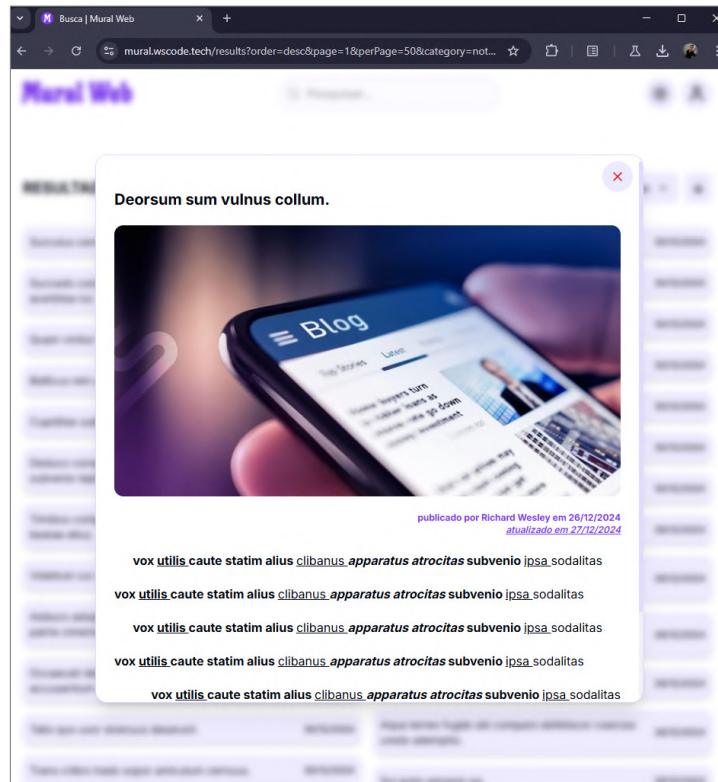
A Figura 14 mostra a tela de entrada do sistema no site onde visitantes podem ver postagens, ou administradores e provedor de conteúdos podem se autenticar para acessar o perfil ou o painel administrativo, além da possibilidade de acessar a área de recuperação de conta. A Figura 15 exibe um exemplo de postagem que pode ser acessada na página inicial ou busca por postagens.

Figura 14 – Captura da página inicial da plataforma web



Fonte: Elaborado pelo autor (2025)

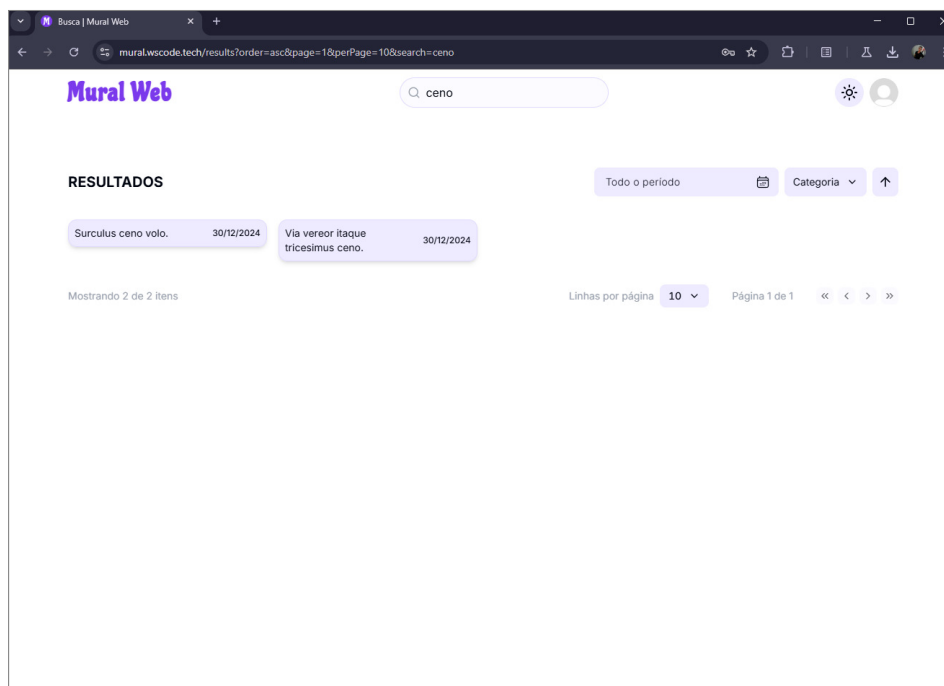
Figura 15 – Captura da exibição de uma postagem na plataforma web



Fonte: Elaborado pelo autor (2025)

A Figura 16 destaca um exemplo de utilização da busca na plataforma web com o resultado já obtido, além das opções de filtragem disponíveis para as postagens encontradas na busca.

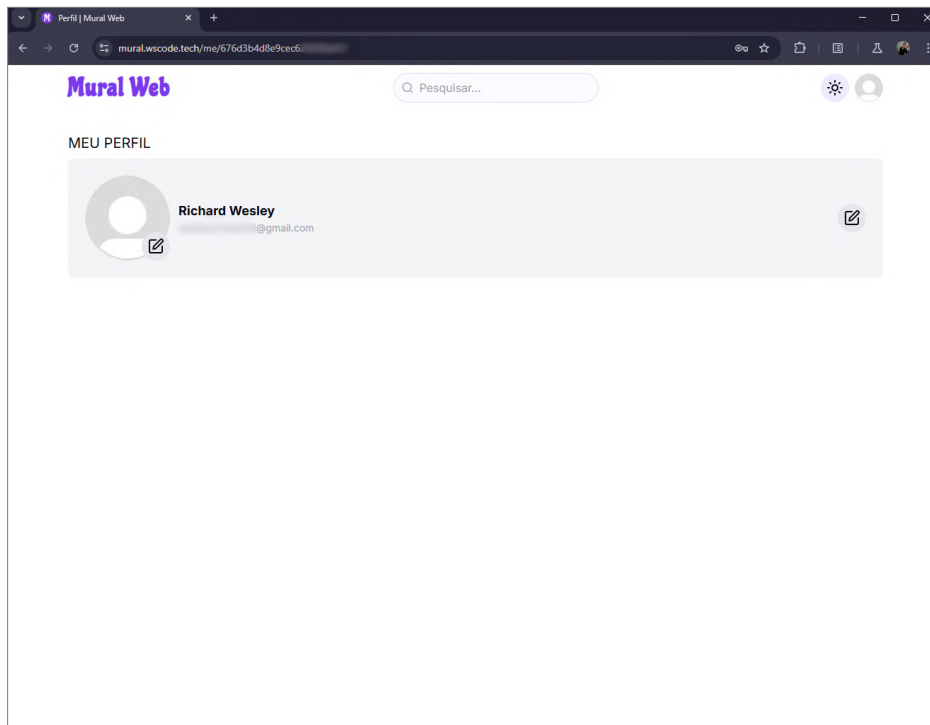
Figura 16 – Captura do uso da barra de busca da plataforma web



Fonte: Elaborado pelo autor (2025)

A Figura 17 mostra a página do perfil do usuário na plataforma web. Nesta área encontram-se os dados do perfil do usuário e opções para alterar os seus dados.

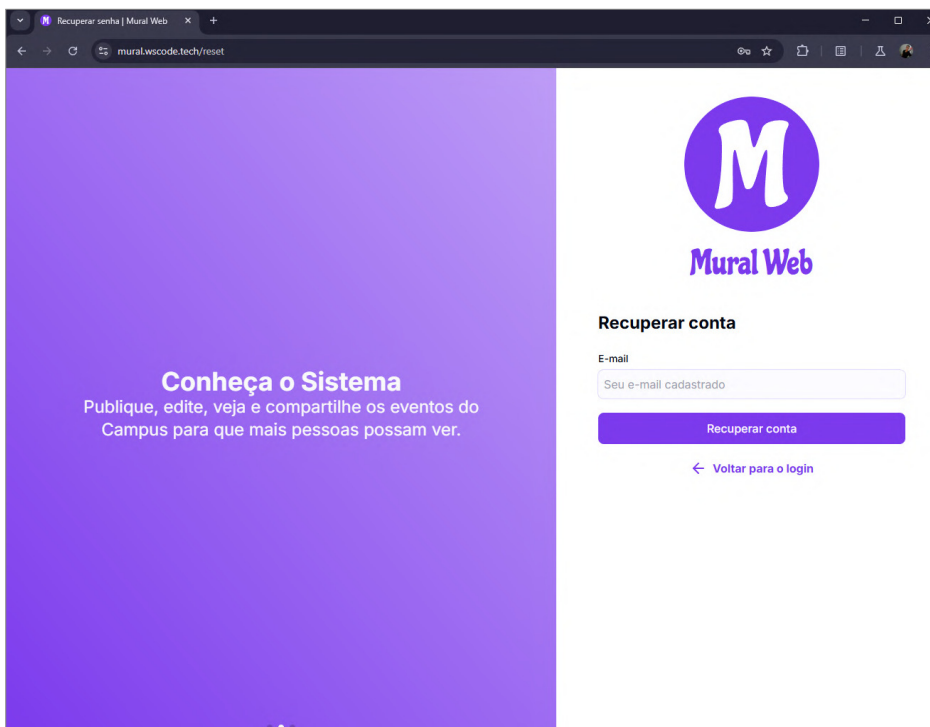
Figura 17 – Captura da página de edição do perfil do usuário na plataforma web



Fonte: Elaborado pelo autor (2025)

A Figura 18 mostra a página de recuperação de conta na plataforma web onde o usuário insere o e-mail da conta cadastrada e recebe um link de recuperação.

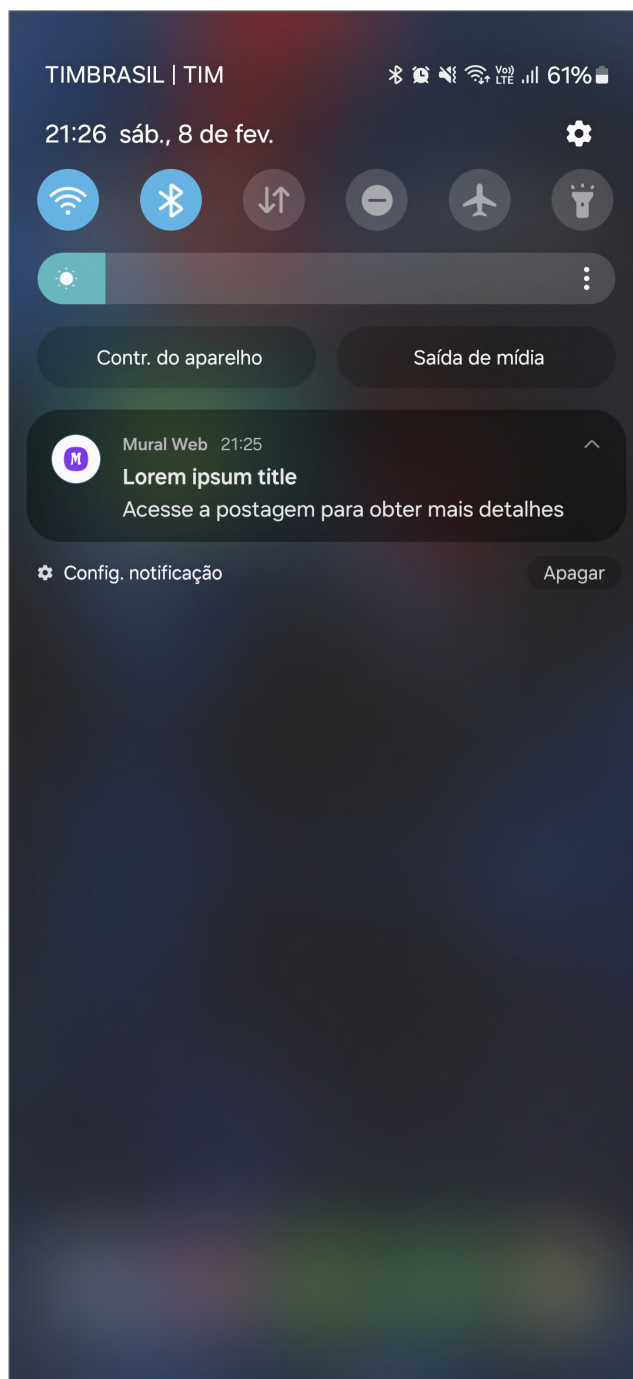
Figura 18 – Captura da página de recuperação de conta do usuário na plataforma web.




Fonte: Elaborado pelo autor (2025)

A Figura 19 exibe um exemplo de notificação recebida no dispositivo móvel sobre uma nova postagem publicada no Mural Web.

Figura 19 – Captura de tela da área de notificações de um dispositivo móvel que exibe uma notificação de nova postagem do Mural Web.



Fonte: Elaborado pelo autor (2025)

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

Documento Digitalizado Ostensivo (Público)

Trabalho de Conclusão de Curso

Assunto:	Trabalho de Conclusão de Curso
Assinado por:	Richard Wesley
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Richard Wesley Barbosa Freitas, DISCENTE (202122010005) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS, em 14/03/2025 08:12:36.

Este documento foi armazenado no SUAP em 14/03/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1418728

Código de Autenticação: b0b5094f00

