



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

EDUARDO VIANA DE ABREU

**DESENVOLVIMENTO DE UMA PLATAFORMA WEB PARA COMUNICAÇÃO E
ORGANIZAÇÃO NO AMBIENTE UNIVERSITÁRIO**

FORTALEZA

2026

EDUARDO VIANA DE ABREU

DESENVOLVIMENTO DE UMA PLATAFORMA WEB PARA COMUNICAÇÃO E
ORGANIZAÇÃO NO AMBIENTE UNIVERSITÁRIO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Ângelo Roncalli
Alencar Brayner

FORTALEZA

2026

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

A145d Abreu, Eduardo Viana de.
Desenvolvimento de uma plataforma web para comunicação e organização no ambiente universitário /
Eduardo Viana de Abreu. – 2026.
58 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia,
Curso de Engenharia de Computação, Fortaleza, 2026.
Orientação: Prof. Dr. Ângelo Roncalli Alencar Brayner.

1. Plataforma web. 2. Comunicação universitária. 3. Gestão de eventos. 4. Desempenho de aplicações
web. 5. Arquitetura de software. I. Título.

CDD 621.39

EDUARDO VIANA DE ABREU

DESENVOLVIMENTO DE UMA PLATAFORMA WEB PARA COMUNICAÇÃO E
ORGANIZAÇÃO NO AMBIENTE UNIVERSITÁRIO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em: 21 de Janeiro de 2026

BANCA EXAMINADORA

Prof. Dr. Ângelo Roncalli Alencar
Brayner (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Me. Ricardo Jardel Nunes da Silveira
Universidade Federal da Ceará (UFC)

Prof. Dr. Jose Wellington Franco da Silva
Universidade Federal da Ceará (UFC)

RESUMO

Este trabalho apresenta o desenvolvimento e a análise de desempenho do UFC Hub, uma plataforma web projetada para centralizar a comunicação e a organização de atividades na Universidade Federal do Ceará. A motivação surge da identificação de limitações em plataformas institucionais existentes que, embora robustas para fins administrativos, carecem de recursos adequados para interação social entre discentes, formação de grupos de estudo e organização de eventos de iniciativa estudantil, especialmente em dispositivos móveis. O UFC Hub foi desenvolvido com base em TypeScript e Next.js, disponibilizado como solução de código aberto e implantado em serviços de nuvem em plano gratuito para fins de prototipação. A plataforma integra funcionalidades como criação e gerenciamento de grupos privados com chat, organização e participação em eventos acadêmicos, blog comunitário, calendário pessoal e perfis de usuário. Os resultados demonstram que o UFC Hub obteve pontuações elevadas nas auditorias realizadas, superando, em diversos indicadores, plataformas institucionais como o SIGAA e o portal UFC Notícias. Além disso, a análise da interface indicou atendimento aos requisitos de usabilidade e acessibilidade, com responsividade adequada entre computadores e dispositivos móveis e suporte a temas visuais diferentes. Conclui-se que a arquitetura e as tecnologias adotadas foram eficazes para atender aos requisitos estabelecidos de desempenho, usabilidade e acessibilidade, oferecendo uma solução complementar às plataformas oficiais da universidade.

Palavras-chave: Plataforma web. Comunicação universitária. Gestão de eventos. Desempenho de aplicações web. Arquitetura de software

ABSTRACT

This work presents the development and performance analysis of UFC Hub, a web platform designed to centralize communication and activity organization at the Universidade Federal do Ceará. The motivation stems from limitations identified in existing institutional platforms which, although robust for administrative purposes, provide insufficient support for student social interaction, study group formation, and student-led event organization, especially on mobile devices. UFC Hub was developed using TypeScript and Next.js, released as an open-source solution, and deployed on cloud services under a free-tier plan for prototyping purposes. The platform integrates features such as the creation and management of private groups with chat, academic event organization and participation, a community blog, a personal calendar, and user profiles. Results show that UFC Hub achieved high scores in the conducted audits, outperforming, across several indicators, institutional platforms such as SIGAA and the UFC News portal. In addition, the interface analysis indicated compliance with usability and accessibility requirements, with adequate responsiveness across desktop and mobile devices and support for different visual themes. Overall, the adopted architecture and technologies proved effective in meeting the established performance, usability, and accessibility requirements, providing a complementary solution to the university's official platforms.

Keywords: Web development. University communication. Event management. Web application performance. Software architecture.

LISTA DE FIGURAS

Figura 1 – Diagrama de Casos de Uso do UFC Hub	22
Figura 2 – Diagrama Entidade-Relacionamento do sistema	25
Figura 3 – Top 10 linguagens de programação no GitHub 2023 - 2025	26
Figura 4 – Funcionamento SSR e SSG	27
Figura 5 – O que é <i>Client-side Rendering</i>	28
Figura 6 – Fluxo de desenvolvimento e migração com Prisma ORM	32
Figura 7 – Estrutura de um JWT	33
Figura 8 – Fluxo de autenticação com JWT	34
Figura 9 – Média de todas as páginas de usuário autenticado - UFC Hub	44
Figura 10 – Média de todas as páginas de usuário visitante (Lighthouse) - UFC Hub	47
Figura 11 – Média de todas as páginas de usuário visitante (PageSpeed Insights) - UFC Hub	48
Figura 12 – Visualização Desktop e Mobile da Home	49
Figura 13 – Visualização da barra de navegação - Mobile	49
Figura 14 – Visualização Desktop e Mobile da Interface de Grupos	50
Figura 15 – Comparativo entre temas: claro, escuro e alto contraste - Mobile	50
Figura 16 – Comparativo de pontuação média por categoria - Desktop	53
Figura 17 – Comparativo de pontuação média por categoria - Mobile	53

LISTA DE TABELAS

Tabela 1 – Classificação das métricas <i>desktop</i>	40
Tabela 2 – Classificação das métricas <i>mobile</i>	40
Tabela 3 – Métricas de desempenho da tela de listagem de eventos - Lighthouse	43
Tabela 4 – Pontuações do Lighthouse para a tela de listagem de eventos	43
Tabela 5 – Métricas de desempenho da tela de listagem de grupos	44
Tabela 6 – Pontuações do Lighthouse para a tela de grupos	44
Tabela 7 – Métricas de desempenho da página Home - Lighthouse	45
Tabela 8 – Pontuações do Lighthouse para a página Home	45
Tabela 9 – Métricas de desempenho da página Home - PageSpeed Insights	46
Tabela 10 – Pontuações do PageSpeed Insights para a página Home	47
Tabela 11 – Médias das métricas de desempenho no comparativo entre sites (Lighthouse).	52
Tabela 12 – Médias das métricas de desempenho no comparativo entre sites (PageSpeed Insights).	52

LISTA DE QUADROS

Quadro 1 – Comparativo funcional entre os trabalhos relacionados	18
Quadro 2 – Requisitos Funcionais do UFC Hub	20
Quadro 3 – Requisitos Não Funcionais do UFC Hub	21

LISTA DE ABREVIATURAS E SIGLAS

ACID	Atomicidade, Consistência, Isolamento e Durabilidade
API	<i>Application Programming Interface</i>
CLS	<i>Cumulative Layout Shift</i>
CSR	<i>Client-side Rendering</i>
CSS	<i>Cascading Style Sheets</i>
DER	Diagrama Entidade-Relacionamento
FCP	<i>First Contentful Paint</i>
JWT	JSON Web Token
LCP	<i>Largest Contentful Paint</i>
NoSQL	<i>Not Only SQL</i>
ORM	<i>Object-Relational Mapping</i>
PDF	<i>Portable Document Format</i>
PET	Programa de Educação Tutorial
PSI	PageSpeed Insights
RTT	<i>Round-Trip Time</i>
SEO	<i>Search Engine Optimization</i>
SGBD	Sistema Gerenciador de Banco de Dados
SI	<i>Speed Index</i>
SIGAA	Sistema Integrado de Gestão de Atividades Acadêmicas
SQL	<i>Structured Query Language</i>
SSG	<i>Static Site Generation</i>
SSR	<i>Server-side Rendering</i>
TBT	<i>Total Blocking Time</i>
TCC	Trabalho de Conclusão de Curso
TTFB	<i>Time to First Byte</i>
UFC	Universidade Federal do Ceará
UFPB	Universidade Federal da Paraíba
UFSC	Universidade Federal de Santa Catarina
UML	<i>Unified Modeling Language</i>
UUID	<i>Universally unique identifier</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Objetivos	13
1.2	Objetivos Específicos	13
1.3	Estrutura do Trabalho	13
2	TRABALHOS RELACIONADOS	15
2.1	SIGAA	15
2.2	Soluções para comunicação e gestão de eventos	15
2.2.1	<i>Acadêmico Móvel</i>	16
2.2.2	<i>Scholar</i>	16
2.2.3	<i>Sistema de Gerenciamento de Eventos para Grupos PET</i>	16
2.2.4	<i>IFES Meet</i>	16
2.2.5	<i>Monography</i>	17
2.3	Comparativo entre os trabalhos	17
3	PLATAFORMA UFC HUB	19
3.1	Escopo	19
3.1.1	<i>Requisitos Funcionais</i>	19
3.1.2	<i>Requisitos Não Funcionais</i>	20
3.1.3	<i>Casos de uso</i>	21
3.1.4	<i>Limitações do Escopo</i>	23
3.2	Banco de Dados	23
3.3	Pilha de Tecnologias	25
3.3.1	Frontend	26
3.3.1.1	<i>Next.js</i>	26
3.3.1.2	<i>Comparativo entre Next.js e React</i>	27
3.3.1.3	<i>Tailwind CSS</i>	28
3.3.1.4	<i>shadcn/ui</i>	29
3.3.2	Backend	29
3.3.2.1	<i>Server Actions</i>	30
3.3.2.2	<i>Consulta de Dados</i>	30
3.3.2.3	<i>Prisma ORM</i>	31

3.3.2.4	<i>Autenticação e Segurança</i>	32
3.4	Implantação e Infraestrutura	34
4	DESEMPENHO	36
4.1	Métricas de Desempenho	36
4.1.1	<i>Time to First Byte (TTFB)</i>	36
4.1.2	<i>Total Transfer Size</i>	37
4.1.3	<i>First Contentful Paint (FCP)</i>	37
4.1.4	<i>Total Blocking Time (TBT)</i>	37
4.1.5	<i>Speed Index</i>	37
4.1.6	<i>Largest Contentful Paint (LCP)</i>	38
4.1.7	<i>Cumulative Layout Shift (CLS)</i>	38
4.2	Ferramentas de Avaliação	38
4.2.1	<i>Lighthouse</i>	38
4.2.2	<i>PageSpeed Insights</i>	39
4.2.3	<i>Protocolo de Coleta de Dados</i>	39
4.2.4	<i>Cálculo dos Escores</i>	40
4.3	Simulação de Dados	41
4.4	Resultados UFC Hub	42
4.4.1	<i>Cenário 1: Usuário Autenticado</i>	42
4.4.2	<i>Cenário 2: Usuário Visitante</i>	45
4.4.3	<i>Responsividade e Acessibilidade</i>	48
4.5	Comparativo com Outros Sites	51
4.5.1	<i>Análise Geral de Desempenho</i>	51
4.6	Conclusão dos resultados	54
5	CONCLUSÃO	55
5.1	Limitações	55
5.2	Trabalhos Futuros	56
	REFERÊNCIAS	57

1 INTRODUÇÃO

A comunicação e a organização de atividades no ambiente universitário tornaram-se demandas centrais, sobretudo pelo aumento do volume de informações e pela multiplicidade de iniciativas estudantis. Com a evolução crescente dos meios de comunicação, a comunicação organizacional deixa de ser apenas um fluxo informativo e passa a assumir caráter mais interativo (KUNSCH, 2007). Nesse cenário, estudantes e docentes precisam de canais que facilitem a divulgação de ações, a coordenação de grupos, o compartilhamento de materiais e o acompanhamento de eventos e prazos acadêmicos.

No entanto, plataformas institucionais costumam concentrar-se em rotinas administrativas e acadêmicas formais, oferecendo suporte limitado para a interação cotidiana entre estudantes e para a organização de eventos estudantis. No contexto da Universidade Federal do Ceará, observa-se que o SIGAA cumpre funções essenciais de gestão acadêmica, mas não prioriza recursos voltados à comunicação e à formação de comunidades, o que pode reduzir o engajamento e dificultar a coordenação de atividades extracurriculares.

Além disso, a experiência de uso em dispositivos móveis no SIGAA é frequentemente prejudicada pela falta de responsividade adequada, o que compromete a navegação em telas menores. Na prática, conteúdos e elementos de interface tendem a não se ajustar ao tamanho do dispositivo, exigindo que o usuário aumente o *zoom* e faça deslocamentos constantes para acessar menus, ler informações e concluir tarefas. Esse tipo de limitação impacta diretamente o uso cotidiano, especialmente considerando que boa parte do acesso a serviços digitais ocorre por *smartphones*.

Dessa forma, este trabalho propõe o desenvolvimento do UFC Hub com o objetivo de complementar os sistemas institucionais ao oferecer recursos centrados na interação e na colaboração discente. A proposta consiste em integrar, em um único ambiente, funcionalidades como grupos de estudo, organização e divulgação de eventos, publicações informativas e calendário. Além da implementação da solução, o trabalho realiza uma análise metódica do sistema, apresentando seus requisitos, decisões arquiteturais e uma avaliação de qualidade, a fim de verificar sua adequação ao contexto acadêmico e sua contribuição para a comunidade.

1.1 Objetivos

O objetivo geral deste trabalho é desenvolver e analisar uma plataforma *web* para comunicação e organização no ambiente universitário da Universidade Federal do Ceará, que centralize funcionalidades de interação social, gestão de eventos e grupos de estudo.

1.2 Objetivos Específicos

A partir da análise dos trabalhos relacionados e das limitações identificadas no sistema institucional vigente, os objetivos específicos deste trabalho são:

- Centralizar, em uma única plataforma, funcionalidades de comunicação e organização entre estudantes universitários, incluindo grupos de estudo, eventos acadêmicos e compartilhamento de conteúdos.
- Projetar e implementar a plataforma UFC Hub utilizando tecnologias *web* modernas, com arquitetura baseada em Next.js, adotando desenvolvimento em código aberto e implantação em nuvem.
- Desenvolver uma interface responsiva e acessível, adaptada para computadores e dispositivos móveis, mitigando as limitações de usabilidade identificadas nos sistemas institucionais existentes.
- Priorizar o desempenho como requisito da plataforma, adotando técnicas de otimização e validando os resultados por meio de métricas da indústria, com comparativo frente às plataformas institucionais existentes.

1.3 Estrutura do Trabalho

A estrutura de texto deste trabalho está organizada da seguinte forma:

- Capítulo 2 examina os trabalhos relacionados, iniciando pela análise do SIGAA como sistema institucional vigente e prosseguindo com a revisão de cinco trabalhos acadêmicos que propõem soluções para comunicação e gestão de eventos no contexto universitário. Ao final, apresenta-se um quadro comparativo que explicita as distinções entre essas soluções.
- Capítulo 3 descreve a plataforma UFC Hub, contemplando as etapas de definição de escopo, levantamento de requisitos funcionais e não funcionais, modelagem do banco de dados e definição da arquitetura de software. São detalhadas as escolhas tecnológicas adotadas e mecanismos de autenticação e segurança, bem como a infraestrutura de implantação.

- Capítulo 4 apresenta a avaliação de desempenho da plataforma, descrevendo as métricas utilizadas, as ferramentas de avaliação e o protocolo de coleta de dados. Os resultados são apresentados para usuários autenticados e visitantes, seguidos de uma análise qualitativa de responsividade e acessibilidade. Por fim, realiza-se um comparativo com outras plataformas institucionais.
- Capítulo 5 apresenta as conclusões do trabalho, discutindo as contribuições alcançadas, as limitações identificadas e as sugestões para trabalhos futuros.

2 TRABALHOS RELACIONADOS

Este capítulo contextualiza o desenvolvimento do UFC Hub através da análise do sistema institucional vigente e de outros trabalhos acadêmicos voltados à comunicação no ambiente universitário. Inicialmente, examina-se o Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA), sistema oficial da Universidade Federal do Ceará (UFC), evidenciando seu foco administrativo e as lacunas referentes à interação social. Na sequência, são apresentados cinco trabalhos acadêmicos que propõem soluções para comunicação e gestão de eventos. Por fim, um quadro comparativo explicita as distinções entre essas soluções, ressaltando a proposta de unificar funcionalidades de gestão acadêmica e convivência estudantil em uma plataforma centralizada.

2.1 SIGAA

O SIGAA constitui a principal plataforma da UFC, desempenhando papel central na administração universitária. Sua concepção é predominantemente administrativa e pedagógica, provendo suporte a processos como matrícula em disciplinas, lançamento de notas, controle de frequência e emissão de documentos oficiais. Adicionalmente, o sistema atua como canal oficial para comunicados institucionais e docentes.

Embora robusto em aspectos administrativos, o SIGAA apresenta limitações quanto à integração social e comunicação discente. O fluxo de informações é majoritariamente unidirecional (da instituição para o aluno), sem mecanismos eficazes para a interação direta entre estudantes. O sistema carece de funções nativas para a formação de grupos de estudo ou organização de eventos estudantis. Ademais, a interface, projetada prioritariamente para computadores e gestão de dados, compromete a experiência do usuário em dispositivos móveis devido à baixa adaptabilidade a telas menores. Tais restrições evidenciam a necessidade de soluções complementares focadas na colaboração e na convivência acadêmica.

2.2 Soluções para comunicação e gestão de eventos

Para estabelecer um comparativo com o UFC Hub, analisaram-se cinco trabalhos acadêmicos que abordam problemas de comunicação e organização universitária. A seleção baseou-se na relação direta com os eixos funcionais da aplicação proposta, abrangendo gestão de eventos, redes sociais e acessibilidade à informação.

2.2.1 Acadêmico Móvel

O trabalho de Lima (2018), desenvolvido na UFC, apresenta o *Acadêmico Móvel*, aplicação que adapta conteúdos do portal universitário para dispositivos móveis. A solução viabiliza a consulta de notas, notícias e materiais didáticos de maneira mais acessível. Um diferencial reside no compartilhamento de arquivos entre dispositivos via redes locais, dispensando conexão com a internet. Entretanto, suas funcionalidades restringem-se à consulta de informações, impossibilitando a criação de conteúdo pelo usuário. Outro limitante é a exclusividade para o sistema *Android*.

2.2.2 Scholar

Na Universidade Federal de Santa Catarina (UFSC), Leal e Leal (2019) desenvolveram o *Scholar*, aplicativo voltado ao apoio e organização pessoal do estudante. A ferramenta centraliza informações como horários de aula, cardápio do restaurante universitário e controle de frequência. O ponto forte é a integração com os sistemas da universidade, facilitando o cotidiano discente. Contudo, o aplicativo possui limitações quanto à organização de eventos e carece de uma versão web completa, restringindo o acesso aos dispositivos móveis.

2.2.3 Sistema de Gerenciamento de Eventos para Grupos PET

O trabalho de Silva (2022) descreve um sistema para gestão de eventos de grupos Programa de Educação Tutorial (PET). O objetivo consiste na automação de tarefas como cadastro de eventos, controle de presença, envio de certificados em PDF e autenticação destes via UUID. A solução mostra-se eficiente para a burocracia interna dos grupos, todavia, é restrita por não oferecer recursos para comunicação contínua ou interação social com membros externos.

2.2.4 IFES Meet

De maneira análoga, Onorio e Barreto (2025) apresentam o *IFES Meet*, plataforma web para gestão de eventos institucionais. O sistema permite cadastrar eventos, gerenciar inscrições e emitir certificados. Embora disponha de diversas funções administrativas, o foco recai na organização formal, inexistindo ferramentas para interação entre participantes ou manutenção de contato após o término do evento.

2.2.5 *Monography*

Costa e Oliveira (2022) desenvolveram o *Monography*, rede social destinada a estudantes em fase de Trabalho de Conclusão de Curso (TCC) na Universidade Federal da Paraíba (UFPB). A plataforma provê um ambiente para compartilhamento de referências e ideias de pesquisa, utilizando elementos típicos de redes sociais, como perfis e interações. O *Monography* inova ao introduzir o aspecto social no meio acadêmico, entretanto, atende exclusivamente a uma fase específica do curso, não se configurando como ferramenta de uso contínuo para alunos de outros semestres.

2.3 Comparativo entre os trabalhos

A análise da literatura permite identificar dois perfis predominantes de soluções: sistemas focados estritamente na gestão administrativa de eventos e aplicativos voltados para consulta pessoal ou redes sociais de escopo específico. O UFC Hub busca unificar esses aspectos, integrando recursos de organização e interação em um único ambiente digital.

Embora se reconheçam os avanços dessas soluções, é possível observar que ainda há margem para aprimoramento, uma vez que a maioria das soluções analisadas não apresenta responsividade plena e consistente entre computadores e dispositivos móveis. Além disso, sistemas como o SIGAA e o IFES Meet tendem a priorizar a gestão institucional e fluxos administrativos, dedicando menor atenção a mecanismos voltados à comunicação cotidiana e à articulação autônoma entre estudantes e demais membros da comunidade universitária.

Para explicitar as distinções e possíveis contribuições da aplicação proposta, elaborou-se o Quadro 1, evidenciando os diferentes enfoques de cada sistema. As categorias foram definidas a partir dos eixos recorrentes nos trabalhos relacionados, contemplando organização de eventos, interação social e utilidade acadêmica.

- **Gerenciamento de Eventos:** engloba funcionalidades de criação, divulgação, participação e controle de eventos acadêmicos, incluindo mecanismos administrativos de inscrições, presença e certificação.
- **Social / Interação:** reúne recursos voltados à comunicação direta entre usuários, tais como troca de mensagens, publicações em *feed* ou *blog*, criação de grupos, comentários e reações, promovendo a convivência no ambiente universitário.
- **Utilidade / Organização:** contempla recursos de apoio à organização pessoal e acesso mul-

tiplataforma, incluindo agenda ou calendário acadêmico, disponibilidade em dispositivos móveis e existência de versão web responsiva.

No Quadro 1, o valor “*Sim*” indica suporte pleno à funcionalidade; “*Não*” indica ausência; e “*Parcial*” refere-se a implementações limitadas, restritas a determinados contextos ou com escopo reduzido comparado às demais soluções.

Quadro 1 – Comparativo funcional entre os trabalhos relacionados

Gerenciamento de Eventos					
Funcionalidade	Acadêmico	Scholar	PET	IFES Meet	Monography
Criar Evento	Não	Parcial	Sim	Sim	Não
Participar de Evento	Não	Parcial	Sim	Sim	Não
Controle de Presença	Não	Não	Sim	Sim	Não
Geração de Certificados	Não	Não	Sim	Sim	Não
Social / Interação					
Chat / Mensagens	Não	Não	Não	Não	Sim
Feed / Blog	Não	Não	Não	Não	Sim
Curtidas / Comentários	Não	Não	Não	Não	Sim
Perfil de Usuário	Não	Não	Não	Não	Sim
Criação de Grupos	Não	Não	Não	Não	Não
Interação Aluno–Aluno	Não	Não	Não	Não	Parcial
Utilidade / Organização					
Consulta Acadêmica	Sim	Sim	Não	Não	Não
Agenda / Calendário	Não	Sim	Não	Não	Não
Visualização Mobile	Sim	Sim	Não	Sim	Não
Responsividade Web	Não	Não	Não	Sim	Não

Fonte: Elaborado pelo autor.

Portanto, a revisão conduzida neste capítulo evidencia que as soluções relacionadas atendem, em geral, a problemas específicos, contemplando parcialmente dimensões como gerenciamento de eventos, interação social ou utilidades acadêmicas. Contudo, conforme sintetizado no Quadro 1, permanecem limitações recorrentes, especialmente quanto à integração entre comunicação e organização em um único ambiente e quanto à oferta de uma experiência consistente em diferentes dispositivos.

Dessa forma, identifica-se uma lacuna no contexto da UFC para uma solução que dê suporte à coordenação de iniciativas, à divulgação de informações e à interação entre participantes, com atenção à usabilidade e à acessibilidade multiplataforma. No próximo capítulo, apresenta-se a proposta do UFC Hub e detalham-se seus fundamentos, requisitos e decisões de projeto, discutindo como tais escolhas respondem às necessidades e limitações apontadas nesta análise.

3 PLATAFORMA UFC HUB

A plataforma proposta foi projetada para centralizar a comunicação e a organização no ambiente universitário. Neste capítulo, apresenta-se a proposta de desenvolvimento do sistema, contemplando as etapas de concepção, definição de escopo, levantamento de requisitos e definição da arquitetura de *software*. Por fim, descreve-se como as escolhas tecnológicas adotadas, em conjunto, visam mitigar o déficit de interação e acessibilidade observado nas plataformas oficiais da UFC.

3.1 Escopo

Entender o conceito de escopo e explicitar os requisitos é uma etapa fundamental no desenvolvimento de *software*, pois estabelece as bases de funcionamento do sistema e alinha expectativas quanto às funcionalidades a serem entregues e às restrições que limitam a solução. Segundo Pressman e Maxim (2016), o escopo do *software* descreve as funções e as características que devem ser fornecidas aos usuários finais, incluindo desempenho, restrições e confiabilidade. No UFC Hub, buscou-se identificar carências de interação e organização da comunidade acadêmica e traduzi-las em requisitos funcionais e não funcionais, considerando também diferentes perfis de acesso.

3.1.1 Requisitos Funcionais

Os requisitos funcionais descrevem os comportamentos e funções que o sistema deve prover aos usuários. Conforme Sommerville (2016), a especificação dos requisitos funcionais de um sistema deve buscar ser completa e consistente, definindo todos os serviços necessários e evitando conflitos, embora na prática isso seja desafiador em sistemas complexos.

Com base no comparativo apresentado no Quadro 1, selecionaram-se as funcionalidades recorrentes e mais relevantes para suprir, de forma integrada, esse espaço de interação e organização identificado entre as soluções analisadas. Nesta seleção, optou-se por não priorizar funções de caráter predominantemente burocrático, como emissão de certificados e consultas acadêmicas, pois elas não constituem o foco central desta proposta.

Para organizar e documentar essas funcionalidades de forma clara e reduzir ambiguidades para a equipe de desenvolvimento, elaborou-se o Quadro 2, que categoriza as demandas por módulos da aplicação e explicita o comportamento esperado para usuários autenticados e

visitantes.

Quadro 2 – Requisitos Funcionais do UFC Hub

ID	Módulo	Descrição
RF01	Autenticação	O sistema deve permitir o cadastro de usuário, mediante fornecimento de e-mail, senha, nome e curso.
RF02	Autenticação	O sistema deve permitir a autenticação (<i>login</i>) por e-mail e senha.
RF03	Grupos	O sistema deve permitir que usuários autenticados criem grupos privados, informando nome e descrição.
RF04	Grupos	O sistema deve permitir que o autor do grupo realize edição e exclusão do grupo, bem como adicione membros e atribua/atualize o papel de membros no contexto do grupo.
RF05	Grupos	O sistema deve permitir que o administrador do grupo gerencie membros, incluindo operações de inclusão e remoção.
RF06	Grupos	O sistema deve permitir que membros de um grupo troquem mensagens por meio de <i>chat</i> associado ao respectivo grupo.
RF07	Eventos	O sistema deve permitir que usuários autenticados criem eventos e, quando autores, realizem edição e exclusão, definindo data, local e descrição.
RF08	Eventos	O sistema deve permitir que usuários autenticados interajam com eventos futuros (data posterior ao horário atual), contemplando curtir, comentar e confirmar presença.
RF09	Eventos	O sistema deve permitir que visitantes visualizem eventos individualmente (via URL compartilhável), com acesso apenas de leitura e sem mecanismos de interação.
RF10	Calendário	O sistema deve disponibilizar ao usuário autenticado um calendário para visualização de eventos confirmados, com modos semanal e mensal.
RF11	Blog	O sistema deve permitir que usuários autenticados criem postagens no blog e, quando autores, realizem edição e exclusão.
RF12	Blog	O sistema deve permitir que usuários autenticados interajam com postagens de terceiros, contemplando curtir e comentar.
RF13	Blog	O sistema deve permitir que visitantes visualizem postagens individualmente (via URL compartilhável), com acesso apenas de leitura e sem mecanismos de interação.
RF14	Perfil	O sistema deve permitir a consulta de perfis, exibindo: nome, curso, eventos criados, grupos em comum, postagens no blog e quantitativo de eventos já participados.
RF15	Home	O sistema deve apresentar, na página inicial, eventos e postagens “em alta” e, para usuários autenticados, os grupos com maior nível de interação do usuário.
RF16	Home	O sistema deve permitir acesso de visitantes à página inicial, com visualização de conteúdos públicos, porém sem exibição de grupos.

Fonte: Elaborado pelo autor.

3.1.2 Requisitos Não Funcionais

Os requisitos não funcionais definem atributos de qualidade, restrições e premissas técnicas que o sistema deve respeitar para garantir uso seguro e satisfatório. Conforme comentado por Sommerville (2016), tais requisitos são frequentemente críticos, pois seu descumprimento pode comprometer a utilidade do sistema como um todo.

No UFC Hub, foram priorizados mecanismos de autenticação e restrições de acesso a visitantes, um modelo de acesso baseado em autoria para ações de criação e modificação de conteúdo, desempenho com ênfase em baixa latência nas interações, capacidade compatível com a base potencial total de usuários da comunidade da UFC, bem como usabilidade e acessibilidade em diferentes dispositivos. Esses requisitos são apresentados de forma consolidada no Quadro 3.

Quadro 3 – Requisitos Não Funcionais do UFC Hub

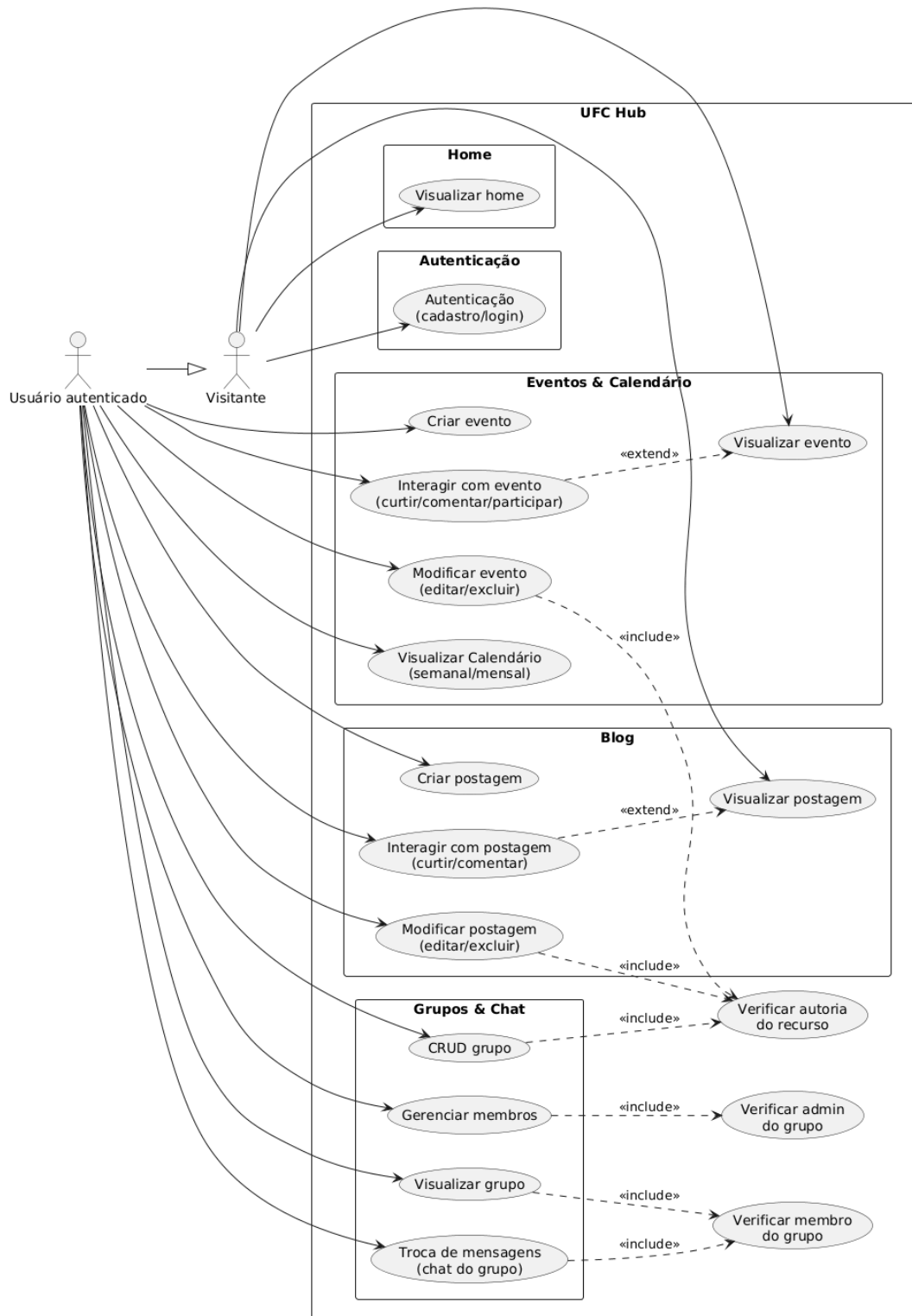
ID	Categoria	Descrição
RNF01	Segurança	O sistema deve exigir autenticação para acesso a funcionalidades restritas e não deve permitir que visitantes acessem páginas e recursos destinados a usuários autenticados, garantindo a visitantes apenas a visualização de conteúdo público.
RNF02	Modelo de Acesso	O sistema deve restringir operações de criação, edição e exclusão ao respectivo autor do conteúdo, assegurando que apenas o responsável pela publicação possa modificá-la ou removê-la.
RNF03	Desempenho	O sistema deve apresentar baixa latência e fluidez na navegação e nas operações interativas, incluindo o módulo de <i>chat</i> de grupos e a renderização de páginas.
RNF04	Capacidade	O sistema deve ser projetado para atender a uma base potencial de aproximadamente 40.000 usuários da comunidade da UFC (UFC, 2019; PROGEP, 2019).
RNF05	Usabilidade	A interface deve ser responsiva e acessível, adaptando-se a <i>desktops</i> , <i>tablets</i> e <i>smartphones</i> , e oferecendo recursos como tema claro, escuro e alto contraste.

Fonte: Elaborado pelo autor.

3.1.3 Casos de uso

Para ilustrar as interações funcionais descritas, a Figura 1 apresenta o Diagrama de Casos de Uso. Este diagrama UML (FOWLER, 2004) representa, de forma simplificada, as ações que os atores (Usuário Autenticado e Visitante) podem realizar no sistema. O Visitante possui acesso de leitura aos conteúdos públicos, enquanto o Usuário Autenticado herda essas permissões e, adicionalmente, realiza operações de criação, modificação e interação. Casos de uso são representados por ovais (por exemplo, "Autenticação (cadastro/login)"). Relações de inclusão («include») indicam passos obrigatórios, como verificações de autenticação, autoria e associação ao grupo. Já relações de extensão («extend») indicam comportamentos opcionais, como interações que estendem a visualização de eventos e postagens.

Figura 1 – Diagrama de Casos de Uso do UFC Hub



Fonte: Elaborado pelo autor.

3.1.4 Limitações do Escopo

O escopo do projeto define não apenas o que será desenvolvido, mas também o que está fora dos limites da implementação atual. Estabelecer essas fronteiras é essencial para garantir a viabilidade do cronograma e mantê-lo dentro do orçamento. Como esta proposta tem uma equipe pequena e prazo curto para implementar o projeto, algumas funcionalidades mais demoradas e burocráticas foram removidas. Desta forma, os seguintes itens foram considerados **fora do escopo**:

- Validação automática de vínculo com a UFC através de integração com o SIGAA ou sistemas governamentais.
- Sistema de notificações *push* ou via e-mail para os usuários.
- Hospedagem de arquivos de mídia (imagens, PDFs) diretamente no servidor da aplicação.

3.2 Banco de Dados

O processo de definição e construção da camada de persistência de dados do UFC Hub foi conduzido segundo a metodologia de *design* de banco de dados proposta por Ramakrishnan e Gehrke (2000). Esta abordagem divide-se em três etapas fundamentais: Análise de Requisitos, *Design* Conceitual e *Design* Lógico. A seguir, detalha-se a aplicação de cada etapa no contexto da plataforma.

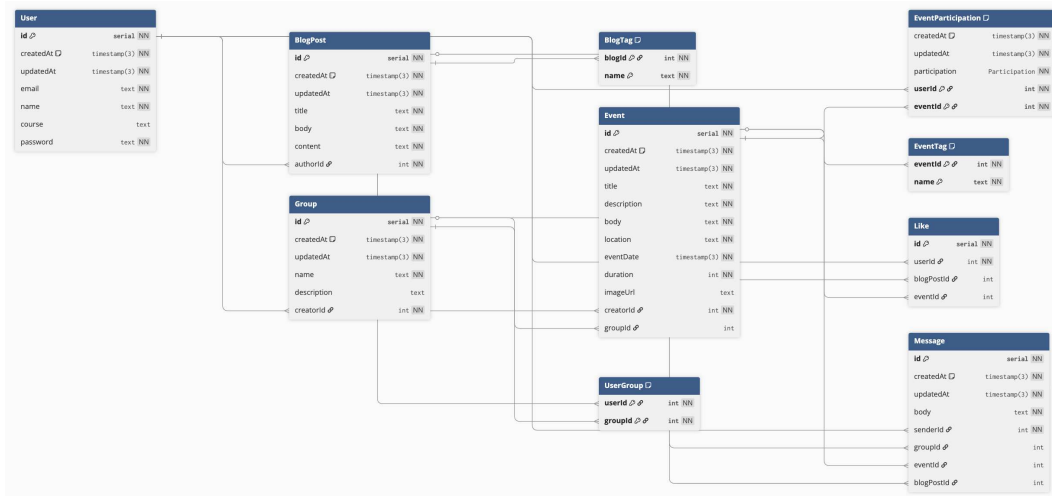
- **Análise de requisitos:** Nesta fase inicial, foram examinados os requisitos funcionais e não funcionais do sistema para determinar quais dados deveriam ser armazenados e como seriam acessados. A partir dos módulos de autenticação, grupos, eventos e blog, identificou-se a necessidade de gerenciar informações estruturadas e fortemente relacionadas, bem como aplicar restrições de acesso. Além disso, o sistema deve sustentar baixa latência nas operações interativas e no *chat*, atendendo a uma base potencial na ordem de dezenas de milhares de usuários (aproximadamente 40.000 (UFC, 2019; PROGEP, 2019)), sem exigir, a priori, estratégias de escalabilidade distribuída.
- **Design conceitual:** Elaborou-se o esquema conceitual de alto nível para representar as entidades do domínio e suas relações. A Figura 2 exemplifica um Diagrama Entidade-Relacionamento (DER), onde permite traduzir as necessidades dos usuários e pode ser implementado em um Sistema Gerenciador de Banco de Dados (SGBD). Destacam-se abaixo as principais entidades definidas do modelo:

- **User (Usuário):** Centraliza os dados de autenticação (e-mail, senha) e perfil (nome, curso), sendo a entidade ativa que interage com as demais.
 - **Group (Grupo):** Representa os espaços de comunicação privada. A relação entre *User* e *Group* foi modelada como muitos-para-muitos (N:N), permitindo que um aluno participe de diversos grupos e um grupo possua múltiplos membros.
 - **Message (Mensagem):** Entidade dependente vinculada a um grupo e a um usuário, viabilizando o histórico do *chat*.
 - **Event (Evento):** Armazena dados temporais e descritivos. Possui relacionamentos para controle de autoria e de participação, alimentando a funcionalidade de Calendário.
 - **BlogPost (Postagem no Blog):** Entidade responsável pelo conteúdo comunitário, relacionando-se com usuários através da autoria e das interações.
- **Design lógico:** Nessa etapa, deve-se escolher um SGBD para implementar o banco de dados, convertendo os conceitos abordados anteriormente na solução lógica a ser efetivada. Dessa forma, selecionou-se o PostgreSQL, um SGBD relacional de código aberto¹, amplamente reconhecido por sua robustez, confiabilidade e conformidade com os padrões da linguagem *Structured Query Language* (SQL) (PostgreSQL Global Development Group, 2025). A natureza *open source* do PostgreSQL oferece vantagens significativas para projetos acadêmicos, como a isenção de custos de licenciamento e o suporte de uma comunidade ativa e global, garantindo a longevidade e a manutenibilidade do sistema. Além disso, o SGBD destaca-se por sua conformidade com as propriedades ACID, assegurando a integridade das transações mesmo em situações de falha.

O armazenamento dos dados segue, portanto, o modelo relacional. Essa escolha justifica-se pela natureza estruturada das informações manipuladas e pela necessidade de relacionamentos fortes, como a integridade referencial. Embora bancos *Not Only SQL* (NoSQL) ofereçam flexibilidade de esquema e escalam melhor em sistemas distribuídos, eles frequentemente priorizam disponibilidade e tolerância a partição em detrimento da consistência imediata (Teorema CAP), o que pode não ser ideal para cenários que exigem consistência rigorosa. Dado que o escopo do UFC Hub é restrito à comunidade universitária e não prevê, inicialmente, a necessidade de escalabilidade massiva distribuída, optou-se por manter a arquitetura simplificada com um banco relacional único, reduzindo a complexidade operacional.

¹ <<https://github.com/resources/articles/what-is-open-source-software>>

Figura 2 – Diagrama Entidade-Relacionamento do sistema



Fonte: Elaborado pelo autor.

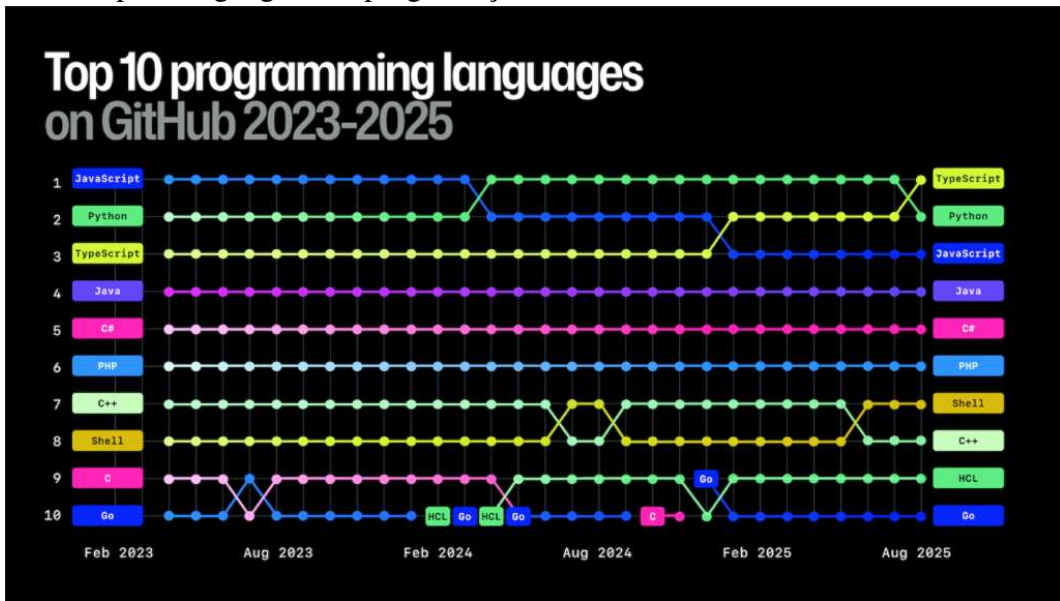
3.3 Pilha de Tecnologias

Considerando os requisitos funcionais e não funcionais definidos, adotou-se uma arquitetura capaz de contemplar as funcionalidades centrais do UFC Hub. Conforme o escopo arquitetural estabelecido, a escolha por uma plataforma *web* busca maximizar a acessibilidade em diferentes dispositivos (computadores e *smartphones*), além de permitir o compartilhamento de links públicos para visitantes, que podem visualizar eventos e postagens sem realizar interações. Nesse contexto, a linguagem TypeScript foi selecionada por atender às demandas do projeto e por ser a tecnologia com maior familiaridade da equipe, favorecendo produtividade e consistência entre as camadas do sistema.

Criada pela Microsoft em 2012, TypeScript é uma linguagem de código aberto que estende o JavaScript ao introduzir tipagem estática e recursos adicionais de verificação, contribuindo para detectar inconsistências ainda em tempo de desenvolvimento e aprimorar a documentação do código (Microsoft, 2025). Além disso, por integrar-se ao ecossistema do *JavaScript*, permite o uso de bibliotecas e *frameworks* consolidados para aplicações *web*, como Next.js, React, Angular e Express.js, viabilizando a adoção de uma *stack* coesa no *frontend* e no *backend*. Como indicativo de adoção na indústria e na comunidade, relatórios do GitHub mostram o TypeScript entre as linguagens de maior utilização no período recente² (Figura 3), o que favorece disponibilidade de ferramentas, bibliotecas e manutenção evolutiva do sistema.

² <<https://octoverse.github.com/#top-languages-over-the-years>>

Figura 3 – Top 10 linguagens de programação no GitHub 2023 - 2025



Fonte: (GitHub, 2025)

3.3.1 Frontend

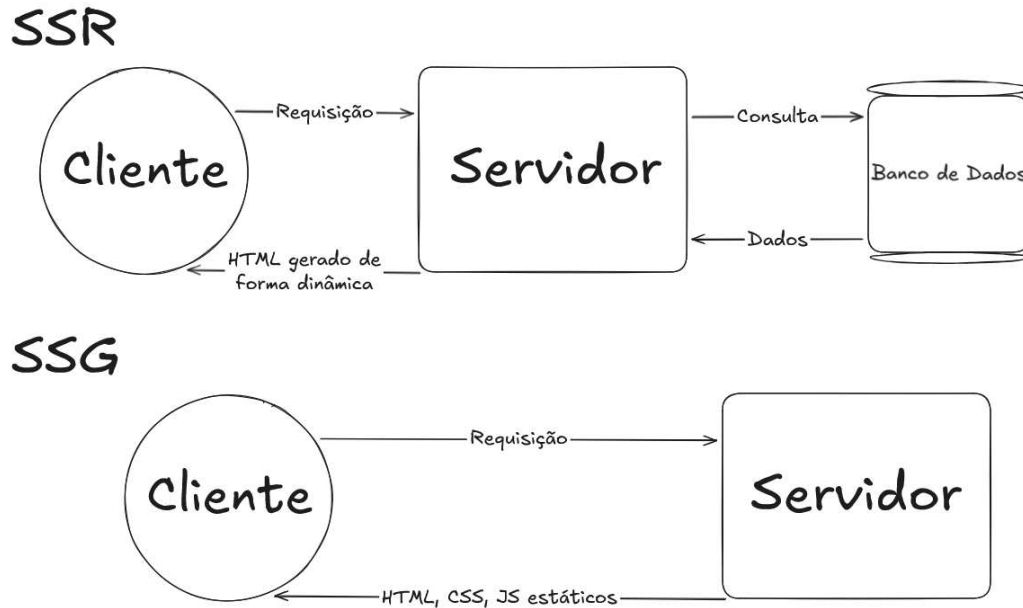
Com o objetivo de garantir uma interface responsiva e consistente em diferentes dispositivos, bem como facilitar a manutenção e a evolução do produto ao longo do tempo, adotaram-se *frameworks* que, em conjunto, buscam equilibrar produtividade de desenvolvimento, padronização visual e extensibilidade da solução.

3.3.1.1 Next.js

O Next.js é um *framework* construído sobre o React, voltado à construção de aplicações *web full stack* com suporte nativo a diferentes estratégias de renderização e organização de rotas. O *framework* permite empregar *Server-side Rendering* (SSR), em que o HTML é gerado a cada requisição, e *Static Site Generation* (SSG), em que o HTML é gerado no *build* e reutilizado em produção, favorecendo o *caching* e contribuindo para reduzir a latência percebida pelo usuário (Figura 4). Além disso, quando aplicável, o Next.js realiza otimizações automáticas de pré-renderização, viabilizando uma aplicação híbrida que combina páginas estáticas e páginas renderizadas sob demanda (Vercel, 2025a).

No UFC Hub, essa flexibilidade é particularmente relevante, pois permite equilibrar páginas com forte necessidade de conteúdo atualizado e autenticado (por exemplo, consulta a grupos e eventos) e páginas de conteúdo público, com maior potencial de *cache* (por exemplo, detalhes de um evento individual ou postagens do blog). Isso resulta em benefícios diretos para

Figura 4 – Funcionamento SSR e SSG



Fonte: Elaborado pelo autor.

o usuário, como menor tempo de carregamento e menor latência percebida, além de uma melhor experiência de uso mesmo em condições de conectividade limitada, especialmente em redes móveis mais lentas.

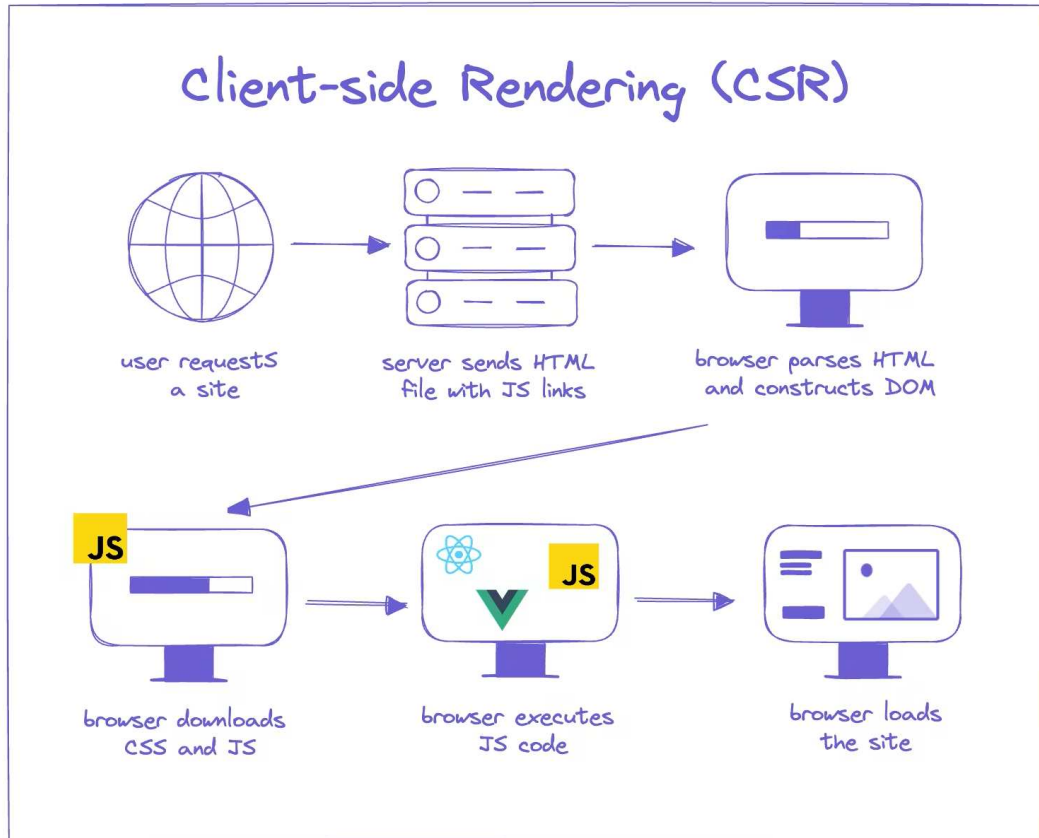
3.3.1.2 Comparativo entre Next.js e React

No contexto do desenvolvimento de plataformas *web* modernas, é relevante comparar o Next.js ao React, uma das tecnologias mais utilizadas pela comunidade de programação³. Embora o Next.js utilize React como base, o React, por si só, é uma biblioteca focada na camada de interface, ela permite construir telas a partir de componentes reutilizáveis e declarativos e atualizar a interface de forma eficiente conforme o estado da aplicação muda (Meta Platforms, 2025).

Diferente do Next.js, o React utiliza por padrão o *Client-side Rendering* (CSR), onde a renderização do conteúdo do site ocorre no cliente. Como mostrado na Figura 5, ao ser solicitado pelo cliente, o servidor responde com um HTML mínimo inicial, contendo o *JavaScript* com o código para buscar dados e renderizar o conteúdo dinamicamente. Isso traz benefícios como atualização de conteúdo sem recarregar a página e impressão maior de fluidez para o usuário, sendo muito recomendado para aplicações com alto grau de interatividade (Amazon,

³ <<https://survey.stackoverflow.co/2025/technology#most-popular-technologies-language>>

Figura 5 – O que é *Client-side Rendering*



Fonte: (prismic, 2024)

2023).

Porém, em termos de desempenho, estudos comparativos recentes indicam que aplicações equivalentes implementadas em Next.js podem apresentar melhora em métricas de carregamento sob conectividades mais lentas, quando comparadas a versões equivalentes em React com foco em CSR (PATI; ZAKI, 2025). Contudo, a análise detalhada de desempenho do UFC Hub (métricas, metodologia de testes e resultados) será apresentada de forma específica no próximo capítulo.

3.3.1.3 Tailwind CSS

Para a camada de estilos, utilizou-se o Tailwind CSS, um *framework* que permite estilizar a interface por meio de classes prontas aplicadas diretamente aos componentes. Essa abordagem reduz a necessidade de escrever *Cascading Style Sheets* (CSS) manualmente, tornando o desenvolvimento mais ágil, pois facilita a padronização visual, simplifica a codificação para múltiplos tamanhos de telas e facilita a alteração de temas de cores completos via CSS (Tailwind Labs, 2025).

Para o usuário, os benefícios se refletem em uma interface mais consistente e previsível, com melhor responsividade em diferentes tamanhos de tela e menor ocorrência de inconsistências de layout. Ademais, como abordado nos requisitos não funcionais do UFC Hub, a padronização de estilos facilita a implementação de recursos de acessibilidade, como modo escuro e alto contraste.

3.3.1.4 *shadcn/ui*

Como biblioteca de componentes para o *frontend*, adotou-se o *shadcn/ui*, que disponibiliza componentes em React na forma de código incorporável ao projeto. Essa abordagem permite adaptar a estrutura e o comportamento dos componentes às necessidades específicas da aplicação (Shadcn, 2025). Além disso, o *shadcn/ui* contempla padrões recorrentes de interface, como botões, formulários e diálogos, reduzindo o esforço de implementação e acelerando a entrega de novas telas com comportamento consistente.

Adicionalmente, os componentes do *shadcn/ui* são compatíveis com a camada de estilos baseada em Tailwind CSS, o que favorece a personalização de forma unificada e coerente com o restante da interface do UFC Hub. Por priorizar boas práticas de acessibilidade, o conjunto de componentes também contribui para uma experiência mais estável e natural.

3.3.2 *Backend*

No desenvolvimento do *backend*, optou-se pelo uso do Next.js, pois o *framework* oferece uma arquitetura que permite integrar *frontend* e *backend* de forma nativa na mesma aplicação. Essa abordagem viabiliza a unificação do desenvolvimento em TypeScript, mantendo uma única linguagem, um único projeto e um único repositório para toda a base de código, o que tende a facilitar a manutenção, reduzir inconsistências entre camadas e simplificar o fluxo de desenvolvimento. Além disso, o Next.js apresenta a vantagem de possibilitar o *deploy* da aplicação completa na plataforma Vercel, incluindo opções de hospedagem gratuita e mecanismos de CI/CD⁴, o que contribui para diminuir custos e agilizar a publicação do sistema.

⁴ <<https://github.com/resources/articles/ci-cd>>

3.3.2.1 *Server Actions*

As *Server Actions* são um recurso do Next.js que permite definir e executar funções diretamente no servidor e acioná-las a partir do *frontend*, como em submissões de formulários ou eventos de componentes. Na prática, elas reduzem a necessidade de criar rotas de *Application Programming Interface* (API) dedicadas para operações simples de escrita, pois a aplicação pode chamar uma função do servidor de forma integrada ao fluxo do React. A chamada pode ocorrer, por exemplo, por meio do atributo `action` de um formulário. Dessa forma, tarefas como inserir registros no banco de dados, validar credenciais, validar entradas do usuário e aplicar regras de negócio são realizadas no servidor, evitando expor detalhes de implementação ao cliente.

No contexto deste trabalho, o uso de *Server Actions* contribui para simplificar a arquitetura de comunicação entre *frontend* e *backend*, uma vez que elimina a necessidade de implementar e expor APIs específicas apenas para atender ao próprio *frontend*. Com isso, a lógica de negócio permanece concentrada no servidor e pode ser acionada de forma direta pelos componentes da aplicação, reduzindo código repetitivo, pontos de integração e esforço de manutenção. Além disso, essa abordagem permite manter a tipagem em TypeScript de ponta a ponta, o que aumenta a consistência entre as camadas, diminui a ocorrência de erros de integração e facilita a evolução do sistema ao longo do desenvolvimento.

3.3.2.2 *Consulta de Dados*

No Next.js, a consulta de dados pode ser realizada diretamente nos *Server Components*, que permitem acessar fontes de dados sem expor credenciais ou detalhes de implementação ao cliente. Nesse modelo, a própria função do componente pode ser assíncrona, possibilitando executar consultas e obter os resultados antes da geração do HTML enviado ao navegador.

Como consequência, os dados já chegam prontos para renderização, reduzindo a necessidade de se utilizar *fetch* no lado do cliente e melhorando a *performance* percebida, especialmente no carregamento inicial. Além disso, ao concentrar a lógica de acesso no servidor, torna-se mais simples aplicar validações, autenticação e autorização no momento da consulta, garantindo que apenas informações permitidas sejam retornadas. Em caso de falhas, essa abordagem também facilita o tratamento de erros, permitindo redirecionar o usuário para o fluxo de login ou apresentar mensagens adequadas.

3.3.2.3 Prisma ORM

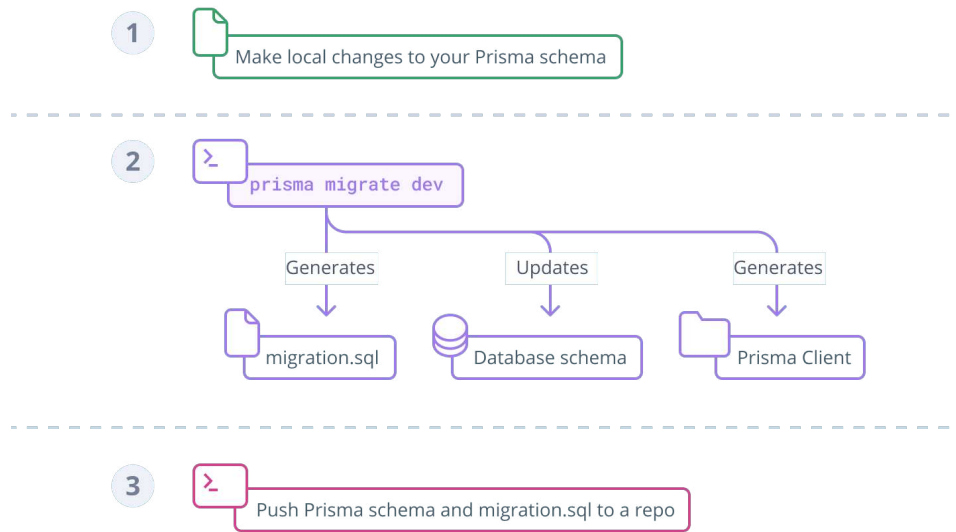
No desenvolvimento de aplicações modernas orientadas a dados, a interação entre a lógica de aplicação e o SGBD representa um componente crítico da arquitetura. Para abstrair a complexidade das consultas SQL puras e aproximar o modelo de dados do paradigma da linguagem de programação utilizada, emprega-se a técnica de *Object-Relational Mapping* (ORM). Os ORMs atuam como uma camada intermediária que traduz as tabelas do banco de dados em objetos ou estruturas manipuláveis via código, aumentando a produtividade, a legibilidade e a segurança contra vulnerabilidades comuns, como *SQL Injection* (SMITH, 2021).

Nesse sentido, associar o uso direto de SQL à lógica da aplicação pode resultar em riscos como consultas ineficientes e dificuldade de manutenção. Para mitigar esses problemas, recomenda-se o isolamento do acesso aos dados em classes específicas que espelhem a estrutura das tabelas do banco de dados, atuando como um *Gateway* entre o sistema e o armazenamento (FOWLER *et al.*, 2002). O Prisma implementa essa diretriz arquitetural no UFC Hub ao abstrair a complexidade da sintaxe SQL e fornecer uma interface clara para cada entidade. Assim, garante-se uma separação efetiva entre a lógica de domínio, responsável pelas regras de grupos e eventos, e a camada de persistência, facilitando a compreensão do código.

Além da organização, esta ferramenta destaca-se no ecossistema TypeScript por oferecer segurança de tipos, garantindo que as consultas ao banco de dados sejam validadas em tempo de compilação. Essa característica é fundamental para a manutenção da integridade dos dados, especialmente para esta proposta em que as entidades têm muitos relacionamentos (Prisma Data, 2025a).

O fluxo de operação do Prisma, conforme ilustrado na Figura 6, fundamenta-se na utilização do arquivo de esquema como a fonte central para a modelagem de dados. O processo inicia-se com a definição ou alteração declarativa das entidades e seus relacionamentos no arquivo `schema.prisma` (1). Após as modificações locais, o desenvolvedor executa o comando de migração (2), que desempenha duas funções essenciais. Primeiramente, ele gera e aplica *scripts* SQL para sincronizar a estrutura do banco de dados com o novo esquema definido. Na sequência, o comando regenera os tipos e classes do Prisma Client, criando uma biblioteca de acesso a dados personalizada e fortemente tipada. Esse mecanismo assegura que as interfaces utilizadas nas consultas estejam sempre consistentes com a versão mais recente do banco de dados, viabilizando a detecção estática de erros e a integração eficiente com as ferramentas de edição de código.

Figura 6 – Fluxo de desenvolvimento e migração com Prisma ORM



Fonte: (Prisma Data, 2025a)

Por fim, o Prisma simplifica a gestão dos relacionamentos complexos exigidos pelos requisitos funcionais do UFC Hub, como a associação de usuários a grupos privados, a confirmação de presença em eventos e a autoria de postagens no blog comunitário. Conseqüentemente, o ORM não apenas agiliza o desenvolvimento das funcionalidades de criação e consulta, mas também assegura que a estrutura de dados permaneça consistente, alinhando-se aos requisitos não funcionais de segurança e de modelo de acesso.

3.3.2.4 Autenticação e Segurança

Em conformidade com os requisitos de segurança e com o modelo de controle de acesso do sistema, o UFC Hub adota um mecanismo de autenticação baseado em *JSON Web Token (JWT)*, associado ao uso de *cookies* HTTP, mantendo o processamento de operações sensíveis no servidor. Nesse modelo, o JWT armazena informações que identificam o usuário autenticado e é persistido no cliente por meio de *cookies*. O *cookie* consiste em uma estrutura de dados do tipo chave–valor utilizada para manter estado entre requisições, assim, a cada solicitação subsequente, o cliente encaminha automaticamente os *cookies* ao servidor, possibilitando a validação da sessão e a verificação de permissões de acesso (STUTTARD; PINTO, 2011).

O JWT é um token compacto composto por três partes, *Header*, *Payload* e *Signature*, conforme ilustrado na Figura 7. Na implementação, o token é assinado com algoritmo HS256, empregando uma chave secreta definida em variável de ambiente do projeto, essa prática asse-

gura a integridade do token, impedindo que um atacante modifique o *payload* sem invalidar a assinatura. Por se tratar de um token assinado e não criptografado, o *payload* deve conter apenas informações mínimas de identificação (nesta proposta, *id* e e-mail) e metadados temporais, evitando-se dados sensíveis (PEYROTT, 2024).

Figura 7 – Estrutura de um JWT

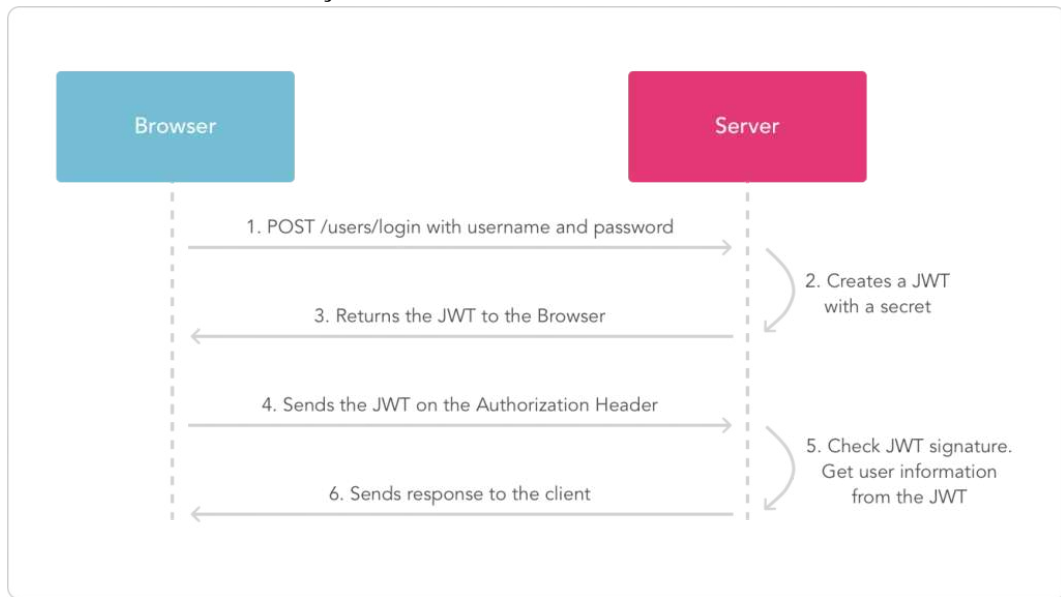


Fonte: (FusionAuth, 2025)

O fluxo de autenticação segue o modelo apresentado na Figura 8. Inicialmente, o usuário realiza cadastro ou autenticação, submetendo suas credenciais ao servidor por um canal seguro. No processo de cadastro, a senha é submetida a um algoritmo de *hashing* com a biblioteca *bcrypt*, de forma que a credencial não seja armazenada diretamente em texto no banco de dados. O *hashing* é um algoritmo projetado para ser computacionalmente custoso, o que eleva o tempo necessário para cada tentativa de senha e, conseqüentemente, dificulta ataques de força bruta em larga escala. Além disso, o algoritmo incorpora o uso de *salt* criptográfico, combinado à senha antes da aplicação do *hash*. Essa prática garante que senhas idênticas produzam valores de *hash* distintos, reduzindo a eficácia de técnicas de quebra de senhas baseadas em valores pré-computados, mesmo em cenários de vazamento da base de dados (FERGUSON *et al.*, 2010).

Além disso, antes de qualquer processamento, os dados de entrada são validados pelo *backend*, assegurando formato, consistência e aderência a regras de negócio (por exemplo, validação de e-mail, presença de campos obrigatórios e restrições de tamanho e tipo). Essa etapa é fundamental para rejeitar entradas inesperadas ou malformadas e, assim, reduzir a superfície de ataque associada à manipulação de parâmetros e à exploração de falhas de validação (STUTTARD; PINTO, 2011). Do ponto de vista do usuário, isso se traduz em maior proteção de dados pessoais e de credenciais, maior confiabilidade nas operações realizadas e redução de

Figura 8 – Fluxo de autenticação com JWT



Fonte: (Medium, 2017)

comportamentos inesperados na interface, fortalecendo a segurança e a integridade do serviço como um todo.

Uma vez autenticadas as credenciais, o servidor emite o JWT com tempo de vida limitado (expiração em 60 minutos) e o persiste no cookie de sessão. Em requisições subsequentes a rotas e ações protegidas, o navegador envia automaticamente o cookie, pois já foi salvo no navegador do cliente. O servidor recupera o token e realiza a verificação da assinatura e da validade temporal. Tokens expirados ou inválidos resultam em negação de acesso, preservando o modelo de acesso do sistema.

Em conjunto, tais mecanismos sustentam os requisitos de segurança e autorização do UFC Hub, garantindo controle de acesso consistente entre visitantes e usuários autenticados e preservando a integridade do estado de autenticação ao longo das interações na aplicação.

3.4 Implantação e Infraestrutura

A arquitetura adotada para o UFC Hub viabiliza a hospedagem da aplicação e da camada de persistência em serviços de nuvem com planos gratuitos, adequados para o escopo de prototipação e validação funcional. Essa abordagem permite escalabilidade e facilidade de manutenção sem a necessidade de gerenciamento direto de servidores físicos, embora apresente limitações técnicas inerentes aos planos não comerciais, conforme será detalhado. O ciclo de vida da aplicação inicia-se no gerenciamento de configuração e versionamento de código, hospedado

publicamente no GitHub <<https://github.com/eduardoabreu09/ufc-hub>>. A escolha por um repositório público promove a transparência do desenvolvimento e permite a reprodutibilidade do projeto acadêmico.

Para a hospedagem da aplicação *full stack*, utilizou-se a plataforma Vercel <<https://ufc-hub.vercel.app>>, que oferece infraestrutura otimizada para o *framework* Next.js e integração nativa com repositórios do GitHub. O projeto foi implantado na região de São Paulo, garantindo baixa latência para os usuários finais localizados no Brasil. No entanto, a utilização do plano gratuito impõe limitações técnicas, como o *timeout* de execução de funções restrito a 10 segundos por requisição e um limite de largura de banda de 100 GB mensais (Vercel, 2025b).

A camada de persistência de dados foi hospedada através da infraestrutura gerenciada pelo Prisma, localizada na região *US East*. Embora a aplicação principal resida no Brasil, a alocação do banco de dados na América do Norte é uma característica comum em planos gratuitos de serviços gerenciados globais. Essa distribuição geográfica introduz uma latência de rede adicional (*Round-Trip Time* (RTT)) nas operações de banco de dados, contudo, a análise quantitativa dessa latência e seu efeito no desempenho geral do sistema será apresentada no próximo capítulo. O plano gratuito do provedor estabelece quotas de armazenamento de 500 MB e limite de no máximo 100.000 operações por mês, adequadas para o escopo de prototipação e validação acadêmica do UFC Hub, mas que exigiriam migração para planos empresariais ou para outro serviço em um cenário de implantação oficial (Prisma Data, 2025b).

Em conclusão, uma infraestrutura sustentada exclusivamente por planos gratuitos impõe restrições operacionais que inviabilizam, na prática, a escala prevista de aproximadamente 40.000 usuários (conforme definido em 3.1.2). Considerando a existência de uma base ativa, o volume de operações mensais na camada de persistência tende a se tornar o principal gargalo, limitando a vazão do sistema e, conseqüentemente, impossibilitando o uso após o limite ser atingido.

Dessa forma, a arquitetura adotada é adequada para validar, no contexto acadêmico, a proposta e o funcionamento do UFC Hub. Entretanto, uma implantação efetiva em ambiente de produção demandaria a migração para serviços comerciais ou para infraestrutura própria, de modo a garantir recursos compatíveis de processamento, armazenamento e capacidade de acesso ao banco de dados para atender, com estabilidade, a comunidade universitária.

4 DESEMPENHO

No contexto atual da *web*, o desempenho deixou de ser apenas um requisito técnico e passou a constituir um componente central da experiência do usuário. Em particular, latências associadas ao estabelecimento de conexão, ao tempo de resposta do servidor e ao processamento do conteúdo no navegador tendem a se acumular, degradando a percepção de fluidez e qualidade do sistema (GRIGORIK, 2013).

Ademais, decisões de *design* e de implementação devem conciliar riqueza visual, complexidade de execução e tempo de carregamento, uma vez que atrasos e instabilidade são percebidos mesmo quando a aplicação se mantém funcionalmente correta (HOGAN, 2014). Essa discussão é especialmente relevante para o UFC Hub, cujo público-alvo inclui estudantes que acessam serviços acadêmicos com frequência por meio de redes móveis, frequentemente caracterizadas por maior latência e variabilidade de conexão.

Neste capítulo, avaliou-se o desempenho do UFC Hub com base em métricas consolidadas na indústria para tempo de carregamento, estabilidade visual, acessibilidade e responsividade. A aplicação foi desenvolvida com o *framework* Next.js, que disponibiliza estratégias de renderização no servidor (SSR) e de geração estática (SSG), além de mecanismos de *cache*. Em conjunto, essas estratégias buscam aprimorar a entrega do conteúdo inicial e a interatividade, reduzindo custos de renderização no cliente, o que fundamenta a avaliação sistemática apresentada neste capítulo.

4.1 Métricas de Desempenho

Para quantificar e analisar o desempenho da plataforma UFC Hub, foram selecionadas métricas essenciais que refletem tanto a eficiência técnica da infraestrutura quanto a qualidade da experiência do usuário. A seleção destas métricas fundamenta-se nas diretrizes estabelecidas em (EDGAR, 2024) e nas recomendações atuais do Google para otimização web, conhecidas como *Core Web Vitals*¹ (Google Developers, 2025a).

4.1.1 *Time to First Byte (TTFB)*

O *Time to First Byte* (TTFB) mensura o intervalo de tempo entre o envio da requisição inicial pelo navegador ao servidor e o retorno do primeiro byte de dados do servidor ao navegador.

¹ <<https://web.dev/articles/vitals#core-web-vitals>>

Segundo Edgar (2024), esta métrica engloba etapas críticas como a resolução de DNS e o estabelecimento da conexão TCP. Um TTFB elevado indica latência na rede ou ineficiência no processamento do servidor. No contexto do UFC Hub, esta métrica é vital para avaliar a *performance* das *Server Actions* do Next.js e a eficiência das consultas ao banco de dados PostgreSQL.

4.1.2 Total Transfer Size

O *Total Transfer Size* refere-se ao volume total de dados transferidos do servidor para o navegador para carregar a página. Esta medida é diretamente impactada pela quantidade de requisições e pela eficiência da compressão de recursos (como imagens, *scripts* e folhas de estilo). A redução do tamanho de transferência é crucial para dispositivos móveis ou redes com largura de banda limitada.

4.1.3 First Contentful Paint (FCP)

O *First Contentful Paint* (FCP) marca o momento em que o navegador renderiza qualquer parte do conteúdo da tela, como texto e imagens. O FCP é um indicador importante para o usuário de que a página está carregando e é funcional. Um FCP rápido reduz a taxa de abandono, fornecendo *feedback* visual imediato.

4.1.4 Total Blocking Time (TBT)

O *Total Blocking Time* (TBT) quantifica a falta de interatividade de uma página. Ele mede o tempo total em que a página fica sem resposta para as entradas do usuário, como cliques ou toques na tela. No cálculo do TBT, considera-se como tarefa bloqueante toda execução que excede 50 milissegundos, somando-se apenas o tempo que ultrapassa esse limiar. Um TBT baixo é essencial para garantir que a interface responda fluidamente ao carregamento inicial.

4.1.5 Speed Index

O *Speed Index* (SI) mede a rapidez com que o conteúdo de uma página é visualmente preenchido durante o carregamento. Diferentemente de métricas baseadas em marcos temporais únicos, o SI é calculado com base na progressão visual ao longo do tempo. Esta métrica oferece uma visão abrangente da experiência de carregamento percebida pelo usuário.

4.1.6 *Largest Contentful Paint (LCP)*

O *Largest Contentful Paint (LCP)*, uma das métricas principais dos *Core Web Vitals*, reporta o tempo necessário para renderizar a maior imagem ou bloco de texto visível na tela do usuário, funcionando como um indicador de que o conteúdo principal foi carregado e, portanto, de percepção de velocidade. Diferentemente do FCP, que marca o primeiro conteúdo visível, o LCP se aproxima da percepção de utilidade da página.

4.1.7 *Cumulative Layout Shift (CLS)*

O *Cumulative Layout Shift (CLS)* avalia a estabilidade visual da página. Esta métrica quantifica a soma de todas as mudanças de layout inesperadas que ocorrem durante a vida útil da página. Mudanças visuais bruscas podem causar cliques acidentais e frustração do usuário.

4.2 Ferramentas de Avaliação

Para a coleta das métricas de desempenho detalhadas na Seção 4.1, foram utilizadas as ferramentas Lighthouse e PageSpeed Insights, ambas desenvolvidas pelo Google e amplamente reconhecidas na comunidade de desenvolvimento web.

4.2.1 *Lighthouse*

O Lighthouse é uma ferramenta de código aberto projetada para auditar a qualidade de páginas web, abrangendo aspectos como desempenho, acessibilidade, boas práticas e otimização para motores de busca (Google Developers, 2025a). A ferramenta funciona por meio de um processo de auditoria que simula o carregamento da página em condições controladas.

O Lighthouse executa uma série de testes automatizados e fornece um relatório detalhado com pontuações e sugestões de melhoria. No âmbito deste trabalho, os testes foram realizados localmente para garantir maior reprodutibilidade. As configurações utilizadas para as coletas foram:

- **Hardware:** MacBook Air com processador Apple M4.
- **Navegador:** Google Chrome em modo anônimo, para assegurar a ausência de interferências de *cache* local ou extensões.

4.2.2 *PageSpeed Insights*

O PageSpeed Insights é um serviço que fornece relatórios sobre a experiência do usuário em dispositivos móveis e *desktop*, oferecendo sugestões de otimização (Google Developers, 2025c). A principal distinção técnica entre o PSI e o Lighthouse reside na natureza dos dados fornecidos: enquanto o Lighthouse foca primordialmente em dados locais, o PSI integra dados de campo.

Os dados de laboratório são fundamentais para a depuração de problemas de desempenho, pois são coletados em um ambiente controlado e reproduzível. Em contrapartida, os dados de campo refletem a experiência real dos usuários finais, capturando variações de hardware, latência de rede e condições de conectividade reais. Para o UFC Hub, o uso dessas ferramentas permitiu uma análise técnica rigorosa, validando as escolhas arquiteturais sob a ótica da eficiência e da experiência do usuário.

4.2.3 *Protocolo de Coleta de Dados*

Foram definidos dois cenários de avaliação: usuário autenticado e visitante. No cenário autenticado, selecionaram-se páginas representativas do uso da aplicação e executaram-se as medições com o Lighthouse. Devido às restrições do processo de autenticação, os ensaios desse cenário foram conduzidos exclusivamente em ambiente local, no qual foi possível realizar o *login* e instrumentar as páginas com consistência.

Já nas páginas acessíveis sem autenticação, as medições foram conduzidas adicionalmente com o PageSpeed Insights. Essa estratégia permitiu comparar os resultados obtidos em ambiente local com aqueles estimados em um contexto de avaliação externa, ampliando a robustez da análise.

Em cada página selecionada, foram efetuadas cinco execuções consecutivas. A adoção de múltiplas iterações visa reduzir a variabilidade inerente a testes de desempenho em aplicações *web*, possibilitando a mitigação de valores atípicos ocasionados por flutuações transitórias de latência de rede ou variações no processamento do dispositivo de teste.

Visando à análise e síntese dos dados coletados, calcularam-se, para cada métrica, medidas de tendência central e de dispersão, conforme descrito a seguir:

- **Média aritmética:** Empregada para estimar o valor central esperado do desempenho.
- **Mediana:** Utilizada para caracterizar a tendência central com menor sensibilidade a

valores extremos.

- **Desvio padrão:** Calculado para quantificar a dispersão entre as execuções e, consequentemente, a estabilidade dos resultados.
- **Mínimo e máximo:** Adotados para delimitar o intervalo de variação observado nos experimentos.

Dessa forma, a interpretação dos dados baseou-se na elaboração das Tabelas 1 e 2, as quais classificam cada métrica reportada pelo Lighthouse em três faixas de desempenho: *rápido*, *moderado* e *lento*.

Tabela 1 – Classificação das métricas *desktop*

Métrica	Rápido	Moderado	Lento	Unidade
FCP	≤ 900	900–1600	> 1600	ms
LCP	≤ 1200	1200–2400	> 2400	ms
SI	≤ 1300	1300–2300	> 2300	ms
TBT	≤ 150	150–350	> 350	ms
CLS	≤ 0,1	0,1–0,25	> 0,25	–

Fonte: (Google Developers, 2019)

Tabela 2 – Classificação das métricas *mobile*

Métrica	Rápido	Moderado	Lento	Unidade
FCP	≤ 1800	1800–3000	> 3000	ms
LCP	≤ 2500	2500–4000	> 4000	ms
SI	≤ 3400	3400–5800	> 5800	ms
TBT	≤ 200	200–600	> 600	ms
CLS	≤ 0,1	0,1–0,25	> 0,25	–

Fonte: (Google Developers, 2019)

Para fins de verificação e reprodução dos resultados, todos os dados coletados, bem como os arquivos de auditoria gerados durante os experimentos, estão disponíveis no repositório <<https://github.com/eduardoabreu09/tcc-results>>. Para inspeção e reprodução das análises, esses relatórios podem ser visualizados no Lighthouse Report Viewer, disponível em <<https://googlechrome.github.io/lighthouse/viewer>>.

4.2.4 Cálculo dos Escores

Destaca-se que o escore de desempenho do Lighthouse é uma composição ponderada dessas métricas, atribuindo maior influência a TBT, LCP e CLS. Assim, mesmo quando métricas como FCP e SI permanecem em níveis adequados, aumentos em LCP ou TBT tendem a impactar

de forma mais sensível a pontuação final, sobretudo no cenário *mobile* (Google Developers, 2019).

Em relação à categoria de Acessibilidade, o cálculo ocorre por meio de uma média ponderada de diversas auditorias individuais. Cada elemento avaliado é aprovado ou reprovado integralmente, e o peso de cada auditoria depende de seu impacto na experiência do usuário, logo, os testes que verificam itens críticos para tecnologias assistivas exercem maior influência na nota (Google Developers, 2025b).

Já a categoria de Boas Práticas utiliza uma lógica de média aritmética simples. Todas as auditorias desse grupo possuem peso igual e verificam a conformidade com padrões modernos da *web*, tais como o uso de HTTPS, a ausência de APIs antigas e a segurança de recursos externos. Assim, a pontuação final reflete diretamente a proporção de regras de segurança e modernização de código atendidas pela página (Google Developers, 2019).

Por fim, a categoria de *Search Engine Optimization* (SEO) avalia aspectos técnicos básicos para indexação e rastreamento pelos motores de busca. A pontuação também é calculada via média aritmética simples, e o escore final indica o cumprimento de requisitos fundamentais, como a presença de títulos e descrições meta (Google Developers, 2019).

No âmbito desta proposta, os escores mais relevantes para a avaliação são os de Desempenho e Acessibilidade, por refletirem diretamente requisitos não funcionais definidos anteriormente, relacionados à responsividade do carregamento, à estabilidade da interface e ao suporte a usuários com diferentes necessidades de acesso.

4.3 Simulação de Dados

Com o objetivo de aproximar o ambiente de testes de um cenário de uso consolidado, isto é, após um período de implantação com acúmulo significativo de registros, foram gerados dados artificiais para compor a base de consultas e recursos associados aos usuários. Assim, no momento de execução das auditorias do UFC Hub, o banco de dados encontrava-se populado conforme a estrutura descrita a seguir:

- **Usuários:** 41.529 usuários, com nomes e e-mails gerados aleatoriamente.
- **Eventos:** 2.000 eventos, com textos e datas gerados aleatoriamente, cada evento contendo de 3 a 50 comentários e de 5 a 100 participações.
- **Blog:** 1.801 postagens no blog, cada postagem contendo de 2 a 50 comentários.
- **Grupos:** 1.004 grupos, cada grupo contendo de 5 a 200 mensagens e de 5 a 100 membros.

- **Mensagens:** 200.284 mensagens no total, considerando a soma das mensagens em grupos e dos comentários em eventos e no blog.

Adicionalmente, para os ensaios no cenário de usuário autenticado, o usuário utilizado nos testes foi alocado em 33 grupos selecionados aleatoriamente, de modo a assegurar a presença de conteúdo e interações relevantes nas páginas dependentes de autenticação.

4.4 Resultados UFC Hub

Esta seção apresenta os resultados obtidos na avaliação de desempenho da plataforma UFC Hub. A coleta de dados foi realizada seguindo o protocolo metodológico definido, simulando as condições de acesso em ambientes *desktop* e *mobile*. Os resultados estão organizados em três eixos de análise: o *Cenário 1: Usuário Autenticado*, que avalia as áreas restritas; o *Cenário 2: Usuário Visitante*, que analisa as páginas públicas; e, por fim, uma análise qualitativa da responsividade e acessibilidade da interface gráfica.

4.4.1 Cenário 1: Usuário Autenticado

Neste cenário, analisaram-se as páginas de Eventos, Grupos e Blog, as quais dependem de autenticação para acesso e apresentam conteúdo predominantemente dinâmico, condicionado ao perfil do usuário autenticado. Assim, as medições refletem o desempenho do UFC Hub em rotas representativas do uso real após o *login*.

Conforme a Tabela 3, a tela de listagem de eventos apresentou, em *desktop*, indicadores de carregamento e estabilidade visual compatíveis com uma experiência rápida para o cliente. Em dispositivos móveis, apesar do aumento esperado nos tempos de renderização, os resultados permaneceram, em geral, abaixo dos limiares recomendados.

Esse comportamento é coerente com as pontuações apresentadas na Tabela 4. No computador, a categoria de desempenho atingiu 100, ao passo que, no *mobile*, registrou-se média de 95. A pontuação máxima na categoria de desempenho não foi alcançada em razão de o LCP situar-se na faixa *moderada*, com mediana de 3071 ms. Como essa métrica exerce influência significativa no score final, sua elevação reduz a possibilidade de atingir a pontuação máxima no cenário móvel. Assim, o tempo de carregamento do maior elemento visível configura-se como um ponto de atenção nesse cenário.

A página de grupos também foi avaliada por apresentar conteúdo integralmente

Tabela 3 – Métricas de desempenho da tela de listagem de eventos - Lighthouse

Plataforma	Métrica	Média	Mediana	Desvio Padrão	Mínimo	Máximo	Unidade
Desktop	TTFB	61	62	1,95	58	63	ms
Desktop	FCP	366	365	2,05	364	369	ms
Desktop	TBT	0	0	0,00	0	0	ms
Desktop	LCP	748	797	73,86	642	803	ms
Desktop	CLS	0,00	0,00	0,00	0,00	0,00	-
Desktop	SI	584	688	152,63	380	702	ms
Desktop	Total Transfer Size	626,34	626,40	0,14	626,18	626,50	KB
Mobile	TTFB	63	63	4,60	57	69	ms
Mobile	FCP	1090	1146	126,99	863	1151	ms
Mobile	TBT	7	3	8,65	0	19	ms
Mobile	LCP	2687	3071	860,01	1149	3074	ms
Mobile	CLS	0,00	0,00	0,00	0,00	0,00	-
Mobile	SI	1164	1146	102,77	1049	1332	ms
Mobile	Total Transfer Size	439,91	439,95	0,12	439,71	439,98	KB

Fonte: Elaborado pelo autor

Tabela 4 – Pontuações do Lighthouse para a tela de listagem de eventos

Plataforma	Categoria	Média	Mediana	Desvio Padrão	Mínimo	Máximo
Desktop	Desempenho	100	100	0,00	100	100
Desktop	Acessibilidade	100	100	0,00	100	100
Desktop	Boas Práticas	100	100	0,00	100	100
Desktop	SEO	100	100	0,00	100	100
Mobile	Desempenho	95	94	2,68	94	100
Mobile	Acessibilidade	100	100	0,00	100	100
Mobile	Boas Práticas	100	100	0,00	100	100
Mobile	SEO	100	100	0,00	100	100

Fonte: Elaborado pelo autor

dinâmico, dependente do usuário autenticado. De acordo com a Tabela 5, em *desktop* observou-se carregamento consistente, com tempos reduzidos nas etapas iniciais de renderização e ausência de instabilidade de layout, o que indica maior previsibilidade na apresentação do conteúdo. No *mobile*, de forma similar ao caso anterior, o LCP médio situou-se muito próximo ao limiar superior de desempenho *rápido*, indicando renderização adequada do maior elemento visível, porém com margem reduzida para absorver oscilações de rede ou de processamento.

Com o intuito de sintetizar os resultados do cenário autenticado, elaborou-se o gráfico apresentado na Figura 9, o qual consolida os escores de desempenho das páginas avaliadas (Eventos, Grupos e Blog). Para compor essa visualização, calcularam-se as médias dos escores por página e, posteriormente, agregaram-se os valores por plataforma, possibilitando a comparação do desempenho do UFC Hub entre *desktop* e *mobile*.

Tabela 5 – Métricas de desempenho da tela de listagem de grupos

Plataforma	Métrica	Média	Mediana	Desvio Padrão	Mínimo	Máximo	Unidade
Desktop	TTFB	61	61	1,92	58	63	ms
Desktop	FCP	330	329	1,65	328	332	ms
Desktop	TBT	0	0	0,00	0	0	ms
Desktop	LCP	445	445	1,53	444	447	ms
Desktop	CLS	0,00	0,00	0,00	0,00	0,00	-
Desktop	SI	330	329	1,65	328	332	ms
Desktop	Total Transfer Size	574,04	574,06	0,15	573,83	574,19	KB
Mobile	TTFB	64	64	3,35	60	68	ms
Mobile	FCP	1004	1004	2,27	1002	1008	ms
Mobile	TBT	4	4	2,70	2	9	ms
Mobile	LCP	2503	2504	8,90	2488	2511	ms
Mobile	CLS	0,00	0,00	0,00	0,00	0,00	-
Mobile	SI	1009	1004	9,22	1002	1025	ms
Mobile	Total Transfer Size	383,30	383,34	0,13	383,09	383,42	KB

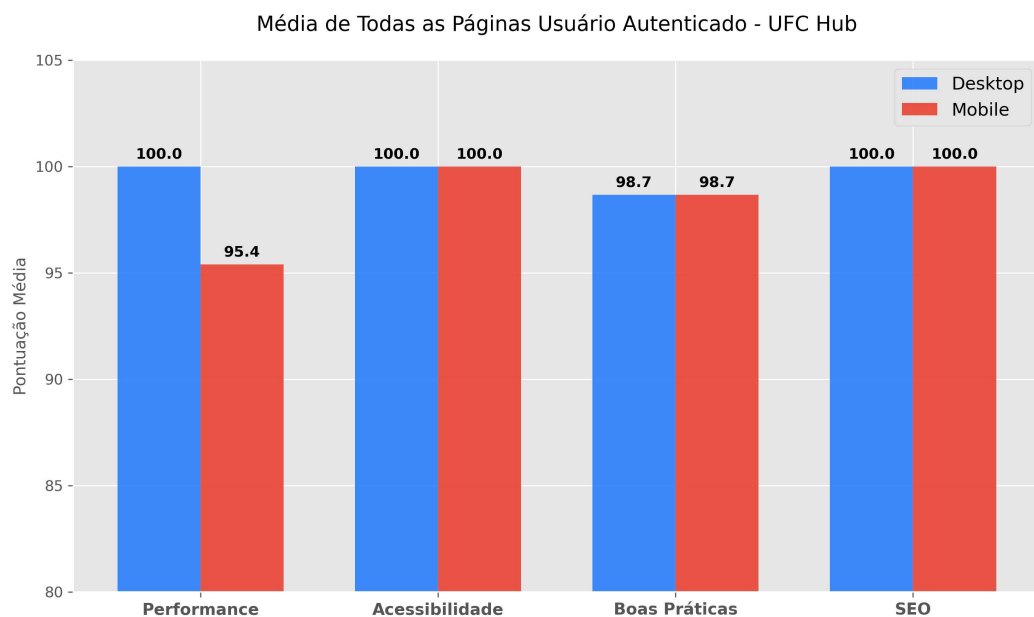
Fonte: Elaborado pelo autor

Tabela 6 – Pontuações do Lighthouse para a tela de grupos

Plataforma	Categoria	Média	Mediana	Desvio Padrão	Mínimo	Máximo
Desktop	Desempenho	100	100	0,00	100	100
Desktop	Acessibilidade	100	100	0,00	100	100
Desktop	Boas Práticas	100	100	0,00	100	100
Desktop	SEO	100	100	0,00	100	100
Mobile	Desempenho	97	97	0,45	97	98
Mobile	Acessibilidade	100	100	0,00	100	100
Mobile	Boas Práticas	100	100	0,00	100	100
Mobile	SEO	100	100	0,00	100	100

Fonte: Elaborado pelo autor

Figura 9 – Média de todas as páginas de usuário autenticado - UFC Hub



Fonte: Elaborado pelo autor

4.4.2 Cenário 2: Usuário Visitante

Para o cenário 2, selecionaram-se rotas acessíveis sem autenticação, especificamente: Home, página de postagem do blog e página de detalhes do evento. Essas telas apresentam, em sua maior parte, conteúdo estático e integralmente público, não envolvendo consultas personalizadas vinculadas ao usuário autenticado. Em razão dessas características, espera-se que exibam desempenho superior ao observado no Cenário 1, no qual predominam páginas dinâmicas e condicionadas ao processo de autenticação.

Conforme as Tabelas 7 e 8, em *desktop* observaram-se tempos reduzidos para as métricas, como observado no cenário anterior, enquadrando a experiência na faixa *rápida* de acordo com os limiares adotados. Em *mobile*, pela primeira vez, os resultados permaneceram confortavelmente dentro dos limites de boa experiência.

Tabela 7 – Métricas de desempenho da página Home - Lighthouse

Plataforma	Métrica	Média	Mediana	Desvio Padrão	Mínimo	Máximo	Unidade
Desktop	TTFB	67	62	10,49	58	84	ms
Desktop	FCP	371	372	0,95	370	372	ms
Desktop	TBT	0	0	0,00	0	0	ms
Desktop	LCP	435	443	17,66	403	444	ms
Desktop	CLS	0,00	0,00	0,00	0,00	0,00	–
Desktop	SI	390	382	24,42	372	431	ms
Desktop	Total Transfer Size	633,72	633,63	0,17	633,57	633,96	KB
Mobile	TTFB	64	65	1,52	62	66	ms
Mobile	FCP	1155	1157	5,07	1147	1160	ms
Mobile	TBT	4	4	1,77	2	6	ms
Mobile	LCP	1755	1729	67,41	1720	1876	ms
Mobile	CLS	0,00	0,00	0,00	0,00	0,00	–
Mobile	SI	1173	1158	35,01	1151	1235	ms
Mobile	Total Transfer Size	548,28	548,23	0,18	548,16	548,59	KB

Fonte: Elaborado pelo autor

Tabela 8 – Pontuações do Lighthouse para a página Home

Plataforma	Categoria	Média	Mediana	Desvio Padrão	Mínimo	Máximo
Desktop	Desempenho	100	100	0,00	100	100
Desktop	Acessibilidade	100	100	0,00	100	100
Desktop	Boas Práticas	96	96	0,00	96	96
Desktop	SEO	100	100	0,00	100	100
Mobile	Desempenho	100	100	0,45	99	100
Mobile	Acessibilidade	100	100	0,00	100	100
Mobile	Boas Práticas	96	96	0,00	96	96
Mobile	SEO	100	100	0,00	100	100

Fonte: Elaborado pelo autor

Dessa forma, os escores de desempenho atingiram a pontuação máxima em ambas as plataformas, evidenciando a vantagem do uso de *cache* em telas predominantemente estáticas. Em contrapartida, a pontuação de boas práticas apresentou redução em relação ao cenário anterior, devido à emissão de uma mensagem de erro no console da aplicação.

Adicionalmente, executou-se a auditoria por meio do PageSpeed Insights, de modo a observar o comportamento da página em um contexto de avaliação externa. As Tabelas 9 e 10 apresentam a síntese desses resultados. Nota-se que, nessa modalidade, o TTFB e o SI tendem a apresentar valores superiores aos obtidos em execução local, bem como maior amplitude entre os valores mínimos e máximos, uma vez que o navegador de teste da ferramenta encontra-se localizado na América do Norte.

Ainda assim, no ambiente *desktop*, apenas o SI aproximou-se da categoria *moderada*, enquanto as demais métricas permaneceram classificadas como *rápidas*. Em dispositivos móveis, além do SI, o LCP enquadrou-se na categoria *moderada* e apresentou desvio padrão elevado. Portanto, os escores de desempenho mantiveram-se bastante elevados, somente observou-se redução na pontuação de acessibilidade em relação ao teste anterior, uma vez que o modo claro (utilizado pelo PSI) apresenta contraste inferior ao modo escuro, adotado nos testes locais com o Lighthouse.

Tabela 9 – Métricas de desempenho da página Home - PageSpeed Insights

Plataforma	Métrica	Média	Mediana	Desvio Padrão	Mínimo	Máximo	Unidade
Desktop	TTFB	294	350	79,37	182	350	ms
Desktop	FCP	319	318	13,67	298	336	ms
Desktop	TBT	14	10	7,16	8	25	ms
Desktop	LCP	535	561	63,87	421	571	ms
Desktop	CLS	0,00	0,00	0,00	0,00	0,00	–
Desktop	SI	1129	901	424,32	872	1861	ms
Desktop	Total Transfer Size	669,96	670,04	0,14	669,81	670,10	KB
Mobile	TTFB	361	349	100,81	232	515	ms
Mobile	FCP	1090	1094	15,14	1065	1106	ms
Mobile	TBT	13	15	6,93	3	19	ms
Mobile	LCP	2285	2551	681,08	1072	2701	ms
Mobile	CLS	0,00	0,00	0,00	0,00	0,00	–
Mobile	SI	2629	2460	367,67	2355	3215	ms
Mobile	Total Transfer Size	506,76	506,77	0,03	506,73	506,80	KB

Fonte: Elaborado pelo autor

Como resultado, embora o navegador do PSI esteja hospedado em outro continente e o banco de dados da aplicação encontre-se fisicamente distante do servidor acessado no Brasil, esse fator não se refletiu em degradação relevante da *performance*. Observou-se que os escores

Tabela 10 – Pontuações do PageSpeed Insights para a página Home

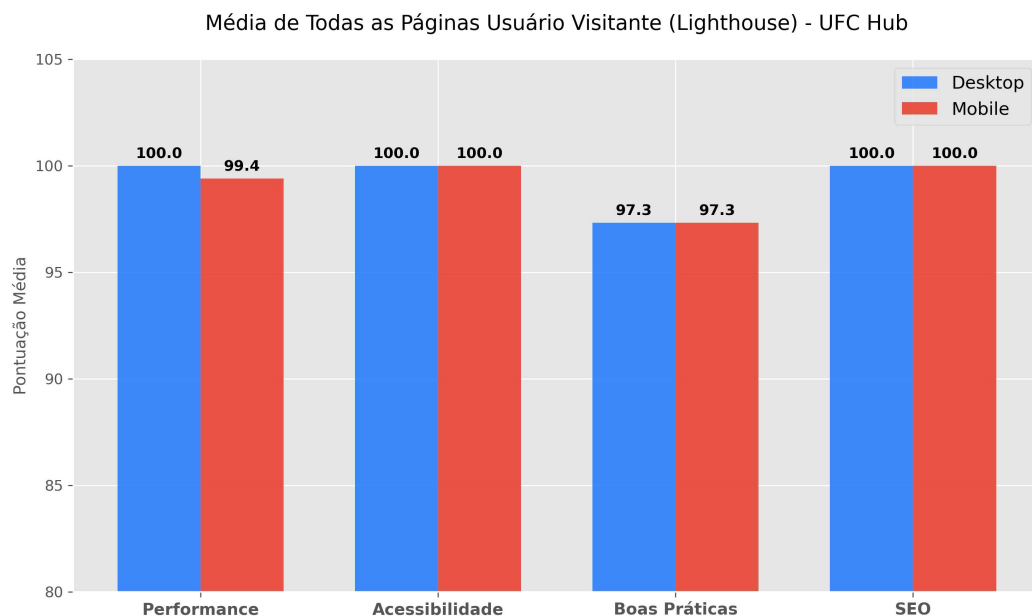
Plataforma	Categoria	Média	Mediana	Desvio Padrão	Mínimo	Máximo
Desktop	Desempenho	99	100	1,30	97	100
Desktop	Acessibilidade	96	96	0,00	96	96
Desktop	Boas Práticas	96	96	0,00	96	96
Desktop	SEO	100	100	0,00	100	100
Mobile	Desempenho	97	97	1,10	96	99
Mobile	Acessibilidade	96	96	0,00	96	96
Mobile	Boas Práticas	96	96	0,00	96	96
Mobile	SEO	100	100	0,00	100	100

Fonte: Elaborado pelo autor

e as métricas obtidos remotamente permaneceram próximos daqueles medidos localmente, indicando variações apenas marginais. Dessa forma, a distância geográfica entre os componentes avaliados não se mostrou determinante para os resultados de desempenho no cenário considerado.

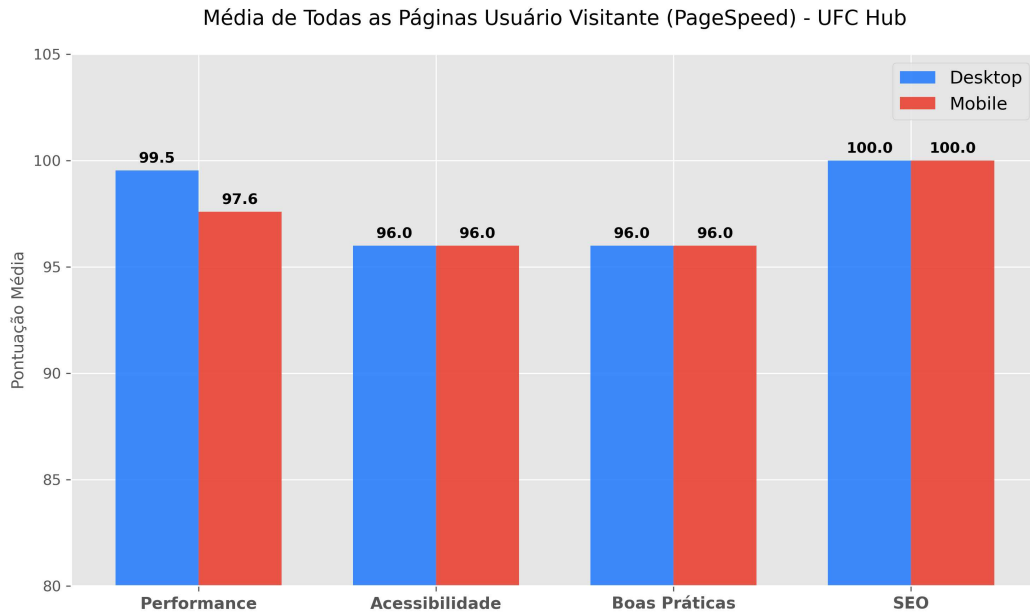
Para sintetizar os resultados deste cenário, elaboraram-se as Figuras 10 e 11 com o objetivo de firmar o desempenho do UFC Hub no cenário de usuário visitante, contemplando tanto auditorias locais quanto avaliações remotas. Essas visualizações consolidam os resultados das páginas analisadas: Home, detalhes do evento e página de postagem do blog, por meio do cálculo da média dos escores de todas as categorias, agrupados por plataforma. Dessa forma, os gráficos permitem comparar, de maneira direta, o comportamento da aplicação entre *desktop* e *mobile* e entre os dois métodos de avaliação.

Figura 10 – Média de todas as páginas de usuário visitante (Lighthouse) - UFC Hub



Fonte: Elaborado pelo autor

Figura 11 – Média de todas as páginas de usuário visitante (PageSpeed Insights) - UFC Hub



Fonte: Elaborado pelo autor

4.4.3 Responsividade e Acessibilidade

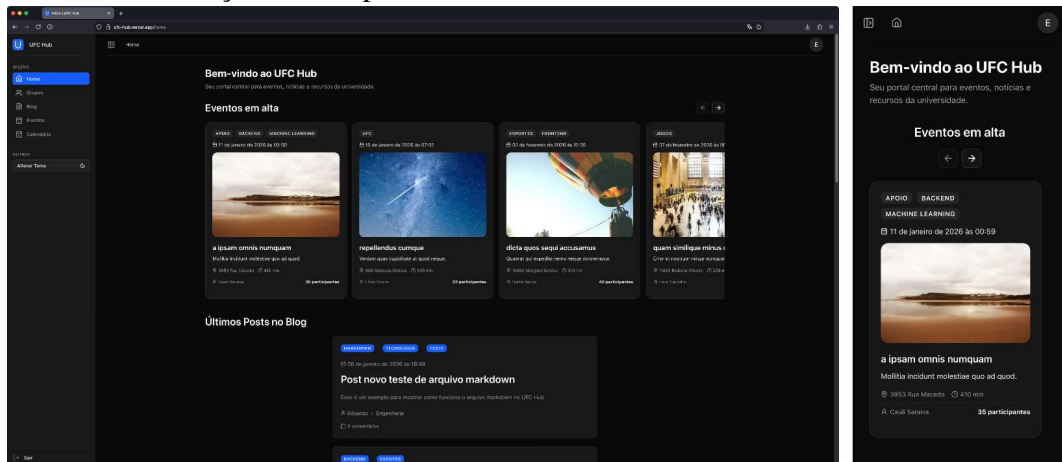
Complementando a análise quantitativa de desempenho, a validação qualitativa da interface é fundamental para comprovar o atendimento ao requisito não funcional de usabilidade, que estipula a necessidade de uma interface responsiva e acessível a diversos usuários.

A Figura 12 ilustra a adaptabilidade da página inicial. Em ambiente *desktop*, a aplicação utiliza a largura de tela disponível para exibir menus laterais de navegação e uma disposição de eventos em lista horizontal. Em contrapartida, na visualização *mobile*, observa-se a reorganização automática dos elementos para uma coluna vertical e o recolhimento dos menus em componentes de sobreposição (Figura 13), preservando a hierarquia da informação e a facilidade de toque em telas menores.

Em módulos de maior complexidade interativa, como o sistema de comunicação em grupos, a responsividade exige não apenas ajustes estéticos, mas alterações funcionais na navegação. A Figura 14 demonstra essa adaptação: enquanto a versão *desktop* adota um *layout* de painel duplo, permitindo a seleção de grupos à esquerda e a visualização do chat à direita simultaneamente, a versão móvel fragmenta essa experiência em telas distintas. Essa abordagem garante que a área de troca de mensagens ocupe a totalidade da tela, otimizando a leitura e a digitação em dispositivos compactos.

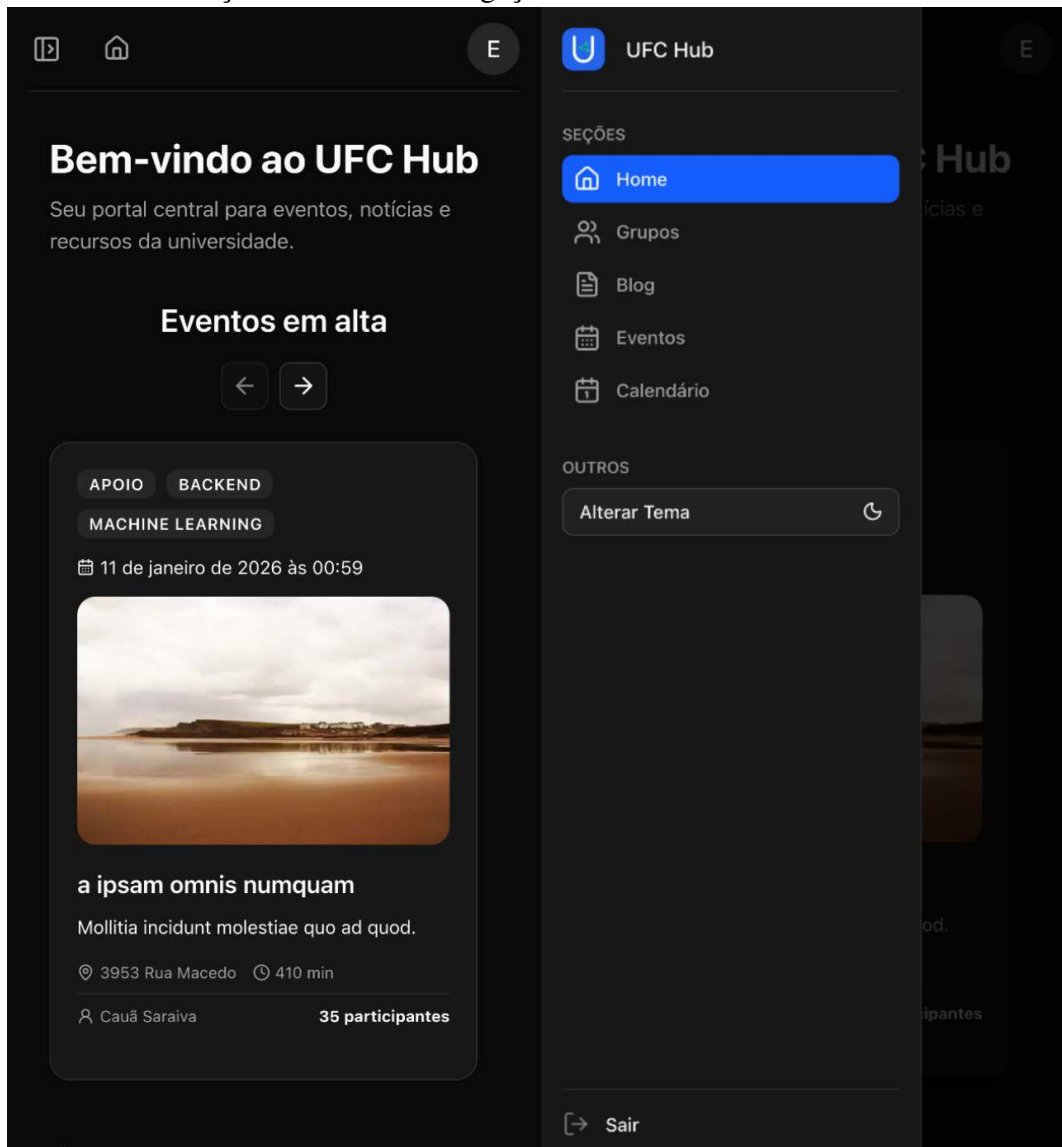
Adicionalmente à responsividade, o UFC Hub contempla requisitos fundamentais de acessibilidade visual. A aplicação implementou suporte a múltiplos temas, permitindo ao usuário

Figura 12 – Visualização Desktop e Mobile da Home



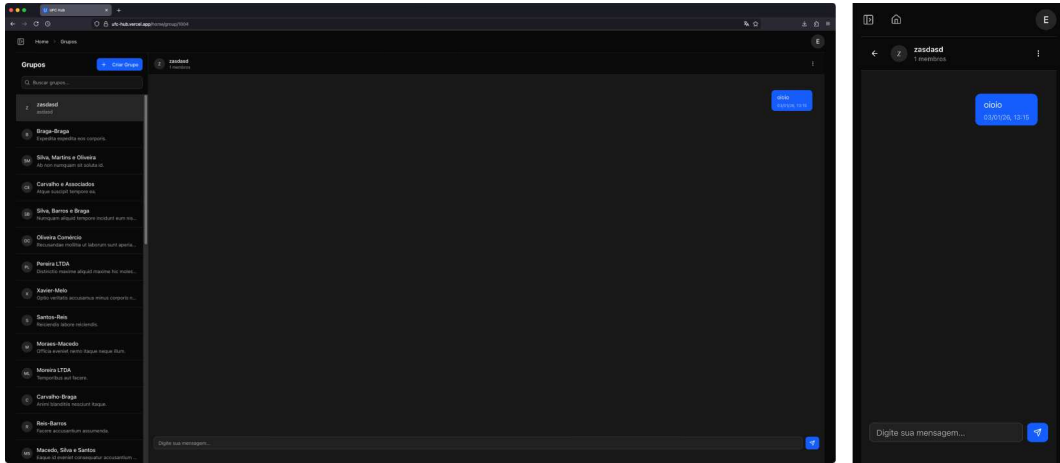
Fonte: Elaborado pelo autor

Figura 13 – Visualização da barra de navegação - Mobile



Fonte: Elaborado pelo autor

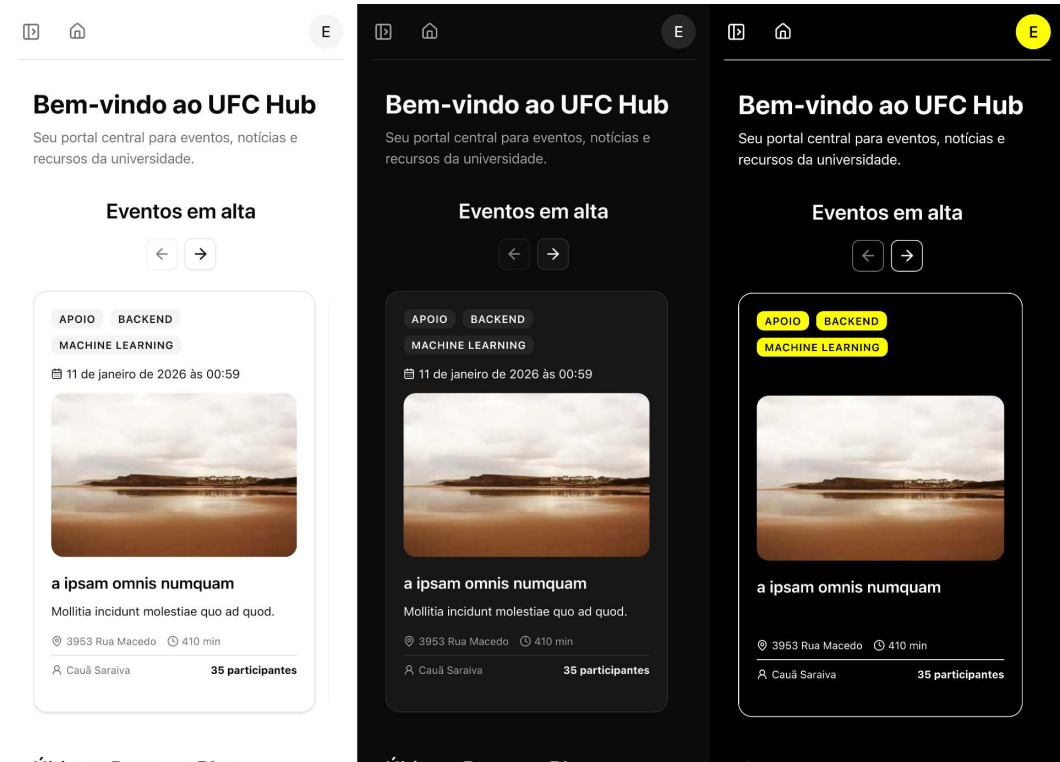
Figura 14 – Visualização Desktop e Mobile da Interface de Grupos



Fonte: Elaborado pelo autor

alternar entre modos de visualização conforme sua preferência ou necessidade. Por padrão, o sistema detecta e aplica automaticamente o tema definido nas configurações do navegador do usuário. A Figura 15 apresenta, respectivamente, o tema claro, o tema escuro e o tema de alto contraste. O modo de alto contraste, em particular, utiliza uma paleta de cores (amarelo sobre preto) que maximiza a legibilidade de elementos textuais e de interface.

Figura 15 – Comparativo entre temas: claro, escuro e alto contraste - Mobile



Fonte: Elaborado pelo autor

4.5 Comparativo com Outros Sites

Com o objetivo de contextualizar o desempenho do UFC Hub em relação a outras plataformas *web* amplamente utilizadas por estudantes e professores da UFC, conduziu-se uma análise comparativa com dois serviços de referência: o SIGAA e o portal UFC Notícias. Para assegurar comparabilidade metodológica, selecionaram-se duas páginas representativas de cada sistema. O procedimento experimental seguiu o mesmo protocolo adotado nas auditorias do UFC Hub, com cinco execuções por página e utilização do Lighthouse e do PageSpeed Insights como ferramentas de avaliação. Ressalta-se que, por se tratar de uma rota autenticada, a página inicial do discente no SIGAA foi avaliada apenas via Lighthouse. As páginas avaliadas estão listadas a seguir:

- **SIGAA**
 - Tela de *login* do SIGAA.
 - Página inicial do discente. (somente Lighthouse)
- **UFC Notícias**
 - Página de listagem das notícias da UFC.
 - Página de detalhes de uma notícia (2025).

4.5.1 Análise Geral de Desempenho

As Tabelas 11 e 12 apresentam uma visão consolidada das métricas de desempenho obtidas no comparativo entre os três sites. Em ambas, os resultados agregam as execuções realizadas em todas as páginas selecionadas de cada plataforma, sendo organizados por tipo de auditoria e por ambiente de execução. Essa estrutura permite comparar, de maneira direta, o comportamento médio de carregamento, responsividade e estabilidade visual entre o UFC Hub, o SIGAA e o portal UFC Notícias.

Na visualização em *desktop*, as métricas de velocidade de carregamento (FCP, LCP e SI) indicam, em média, que o SIGAA e o UFC Notícias se situam na faixa *moderada*, ao passo que o UFC Hub permanece classificado como *rápido*. Em dispositivos móveis, observou-se agravamento desse cenário (Tabela 12), uma vez que o SI dos sistemas comparados deteriorou-se para a categoria *lenta*. Adicionalmente, nota-se que o UFC Notícias apresenta instabilidade visual, evidenciada por CLS de 0,01, ainda que discreta. Observa-se também um breve período de bloqueio durante o carregamento, o que pode induzir a percepção de travamento por parte do

Tabela 11 – Médias das métricas de desempenho no comparativo entre sites (Lighthouse).

Plataforma	Métrica	UFC Hub	SIGAA	UFC Notícias	Unidade
Desktop	TTFB	64	550	428	ms
Desktop	FCP	359	1299	1041	ms
Desktop	TBT	0	0	2	ms
Desktop	LCP	551	1750	1804	ms
Desktop	CLS	0,00	0,00	0,02	–
Desktop	SI	418	2038	1146	ms
Desktop	Total Transfer Size	554,26	533,91	3919,05	KB
Mobile	TTFB	63	547	437	ms
Mobile	FCP	1111	4031	2489	ms
Mobile	TBT	4	11	156	ms
Mobile	LCP	2377	5934	3020	ms
Mobile	CLS	0,00	0,00	0,00	–
Mobile	SI	1130	6058	2516	ms
Mobile	Total Transfer Size	442,14	534,07	3917,75	KB

Fonte: Elaborado pelo autor

Tabela 12 – Médias das métricas de desempenho no comparativo entre sites (PageSpeed Insights).

Plataforma	Métrica	UFC Hub	SIGAA	UFC Notícias	Unidade
Desktop	TTFB	295	179	10	ms
Desktop	FCP	316	730	651	ms
Desktop	TBT	16	39	160	ms
Desktop	LCP	531	939	1175	ms
Desktop	CLS	0,00	0,00	0,01	–
Desktop	SI	1029	4558	3134	ms
Desktop	Total Transfer Size	542,24	329,83	3773,26	KB
Mobile	TTFB	360	182	11	ms
Mobile	FCP	1082	2453	2322	ms
Mobile	TBT	23	1	233	ms
Mobile	LCP	2245	3285	3679	ms
Mobile	CLS	0,00	0,00	0,01	–
Mobile	SI	2646	11046	9197	ms
Mobile	Total Transfer Size	439,13	249,70	3769,74	KB

Fonte: Elaborado pelo autor

usuário.

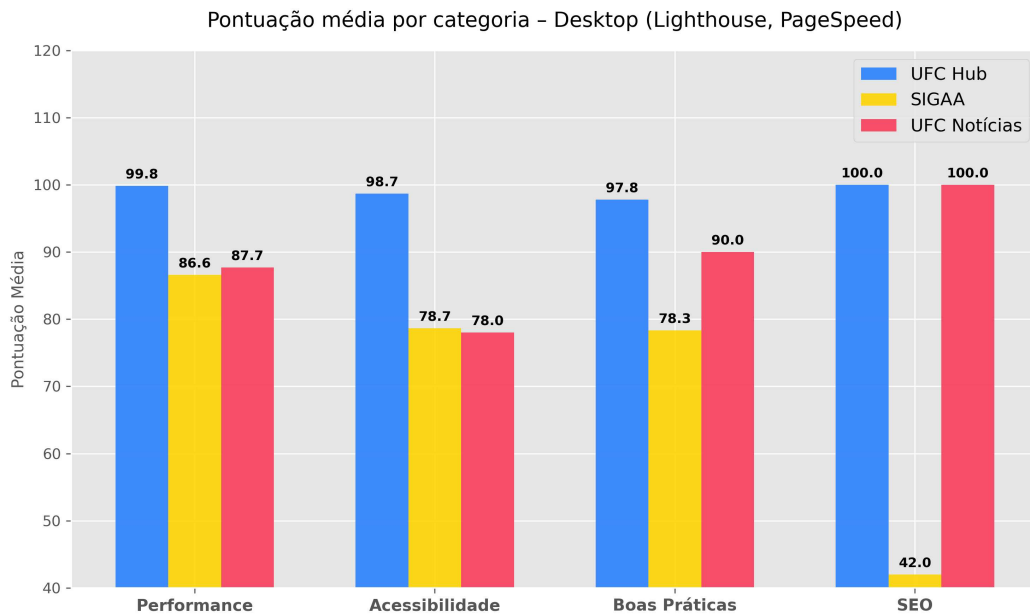
Em relação ao *Total Transfer Size*, enquanto o portal UFC Notícias trafega entre 2,5 MB e 5 MB de dados, o UFC Hub mantém-se na faixa de 440 KB a 550 KB, patamar próximo ao do SIGAA. O peso elevado de dados transferidos impacta diretamente usuários com planos de dados limitados ou conexões instáveis, comuns no ambiente universitário.

Por outro lado, os valores de TTFB do UFC Hub apresentados na Tabela 12 mostram-se elevados tanto em computadores quanto em celulares. Esse comportamento contrasta com os resultados da Tabela 11, na qual os maiores tempos de resposta concentram-se no SIGAA e no UFC Notícias. Ainda assim, no UFC Hub, esse aumento de TTFB não se refletiu em piora

proporcional das métricas de carregamento percebido, sugerindo que a resposta inicial mais lenta não foi suficiente para degradar a experiência final medida pelas demais auditorias.

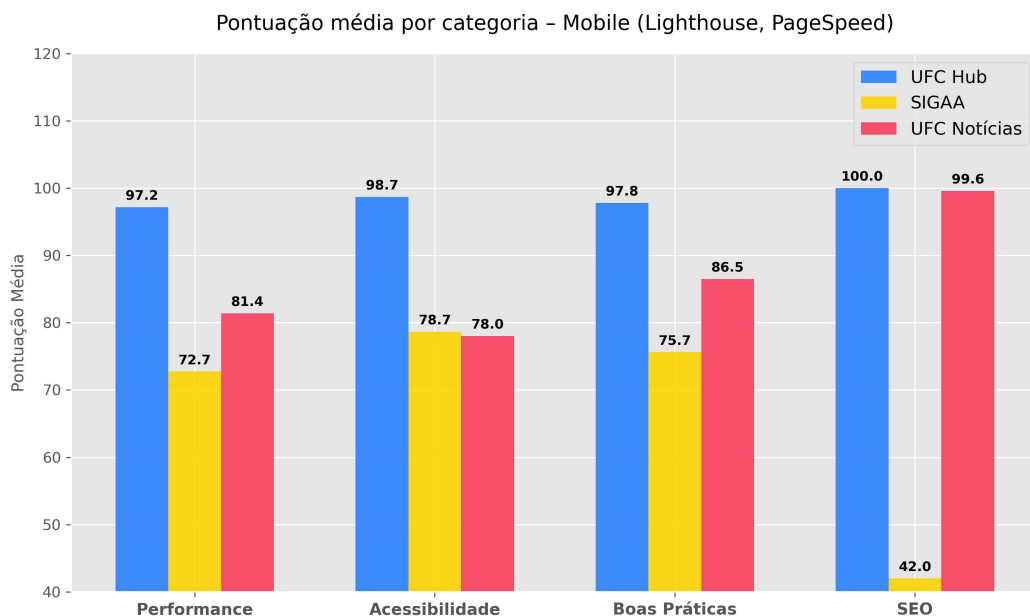
Por fim, as Figuras 16 e 17 sintetizam a pontuação média obtida nas quatro categorias de avaliação do Google: *Performance*, *Acessibilidade*, *Boas Práticas* e *SEO*.

Figura 16 – Comparativo de pontuação média por categoria - Desktop



Fonte: Elaborado pelo autor

Figura 17 – Comparativo de pontuação média por categoria - Mobile



Fonte: Elaborado pelo autor

Observa-se que o UFC Hub lidera consistentemente em todas as categorias, atingindo

pontuação máxima ou próxima a ela em Desempenho e Acessibilidade.

4.6 Conclusão dos resultados

A avaliação sistemática apresentada neste capítulo permite concluir que a arquitetura de *software* e a pilha tecnológica adotadas no UFC Hub atenderam, de forma consistente, aos requisitos de desempenho e qualidade definidos na fase de proposta. Em especial, os resultados sustentam o cumprimento do requisito não funcional de desempenho, uma vez que a aplicação apresentou baixa latência e boa fluidez de renderização mesmo operando sobre uma base de dados populada com mais de 40 mil registros de usuários e milhares de interações simuladas.

A escolha do *framework* Next.js, aliada às estratégias de renderização híbrida SSR e SSG, contribuiu para reduzir gargalos de carregamento e estabilizar a entrega do conteúdo. Esse comportamento é refletido nas métricas de FCP e LCP mantidas em níveis baixos ao longo das rotas avaliadas, indicando que o conteúdo principal é exibido em tempos adequados tanto em ambiente *desktop* quanto em dispositivos móveis. Além disso, a estabilidade visual observada nas páginas analisadas sugere que a implementação da interface com Tailwind CSS e shadcn/ui foi eficaz em evitar mudanças bruscas de *layout*, contribuindo diretamente para uma experiência de uso mais previsível e natural.

No que se refere à acessibilidade e à usabilidade, os escores elevados nas auditorias do Lighthouse e a análise qualitativa do comportamento responsivo corroboram a adequação do *design* inclusivo adotado. A oferta de temas visuais (claro, escuro e alto contraste) atendeu aos requisitos previstos e, ao mesmo tempo, não apresentou indícios de impacto negativo relevante no desempenho, preservando a leveza e a responsividade da aplicação.

Apesar dos resultados favoráveis, é importante registrar limitações deste processo de avaliação. As medições realizadas priorizaram cenários de navegação individual e não incluíram testes de carga com múltiplos usuários simultâneos, o que impede afirmar, com o mesmo grau de confiança, como o sistema se comporta sob concorrência elevada e picos de acesso. Do mesmo modo, a análise de segurança permaneceu em um nível introdutório, baseada sobretudo em boas práticas de implementação e validações funcionais, sem a execução de testes mais expressivos.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento e a análise do UFC Hub, uma plataforma *web* voltada à comunicação e organização no ambiente universitário da UFC. A motivação partiu da identificação de uma lacuna nas plataformas institucionais existentes, como o SIGAA, que, embora robustas em aspectos administrativos, apresentam limitações quanto à interação social entre estudantes, à formação de grupos de estudo e à organização de eventos de iniciativa discente, especialmente quando se considera a experiência em dispositivos móveis.

Como resposta, foi projetado e implementado o UFC Hub, consolidando funcionalidades essenciais ao contexto acadêmico em uma única plataforma. O sistema adota um modelo de acesso em que usuários cadastrados têm uso completo das funcionalidades, enquanto conteúdos públicos podem ser divulgados por links compartilháveis. Essa abertura não compromete a plataforma, pois as regras de permissão e autoria são mantidas de forma consistente.

A avaliação sistemática de desempenho demonstrou que as escolhas arquiteturais foram eficazes. Nos testes conduzidos, o UFC Hub obteve pontuações consistentemente elevadas nas categorias testadas, superando, em diversos indicadores, plataformas institucionais como o próprio SIGAA e o portal UFC Notícias. Adicionalmente, a análise qualitativa da interface confirmou o atendimento aos requisitos de usabilidade e acessibilidade, com responsividade adequada entre dispositivos *desktop* e *mobile*, além da implementação de temas visuais que ampliam a inclusão de usuários com diferentes necessidades.

5.1 Limitações

Apesar dos resultados positivos, algumas limitações devem ser reconhecidas. A hospedagem em serviços gratuitos, como Vercel e Prisma Cloud, impõe restrições de banda, tempo de execução e operações mensais que inviabilizam, na prática, a implantação em larga escala sem migração para planos comerciais ou infraestrutura própria.

Além disso, funcionalidades relevantes foram excluídas do escopo inicial, tais como: integração com sistemas institucionais para validação automática de vínculo, autenticação via e-mail institucional, sistema de notificações push ou por e-mail, e hospedagem de arquivos de mídia diretamente na plataforma. Adicionalmente, a validação de segurança e desempenho limitou-se a testes funcionais e de navegação individual, não contemplando testes de carga massiva com concorrência elevada.

5.2 Trabalhos Futuros

Como trabalhos futuros, sugere-se:

- Migrar a infraestrutura para serviços com capacidade compatível ao uso institucional, seja por meio de planos comerciais ou pela hospedagem em servidores da própria universidade.
- Desenvolver um aplicativo móvel nativo, ampliando a acessibilidade e a conveniência de uso.
- Integração com o SIGAA ou sistemas de autenticação institucional, permitindo login unificado e validação automática de vínculo acadêmico.
- Implementar notificações em tempo real, por push ou e-mail, para eventos, mensagens e interações do *chat*.
- Realizar testes de carga com múltiplos usuários simultâneos e auditorias de segurança mais aprofundadas.
- Conduzir estudos com usuários reais para validação qualitativa da experiência, identificação de melhorias de usabilidade e de novas funcionalidades.
- Aprofundar estudo para acessibilidade da aplicação: estudar formas de melhorar o uso da plataforma para pessoas neurodivergentes por exemplo.
- Criação de um *chatbot* com inteligência artificial para buscar eventos e postagens do blog de maior interesse para o usuário, transformando linguagem escrita em consultas para serem usadas no *backend*.

REFERÊNCIAS

- Amazon. **O poder dos Micro-Frontends: Introdução a SSR, CSR e IR.** 2023. Disponível em: <<https://aws.amazon.com/pt/blogs/aws-brasil/o-poder-dos-micro-frontends-introducao-a-ssr-csr-e-ir>>. Acesso em: Dezembro de 2025.
- COSTA, J.; OLIVEIRA, M. **Monography: Uma Rede Social com Foco em TCC.** Trabalho de Conclusão de Curso — Universidade Federal da Paraíba, João Pessoa, 2022. Disponível em: <<https://repositorio.ufpb.br/jspui/handle/123456789/29076>>. Acesso em: Dezembro de 2025.
- EDGAR, M. **Speed Metrics Guide: Choosing the Right Metrics to Use When Evaluating Websites.** [S.l.]: Apress, 2024. ISBN 979-8-8688-0154-9.
- FERGUSON, N.; SCHNEIER, B.; KOHNO, T. **Cryptography Engineering: Design Principles and Practical Applications.** Indianapolis, IN: Wiley, 2010. ISBN 9780470474242.
- FOWLER, M. **UML Essencial: Um breve guia para a linguagem padrão de modelagem orientada a objetos.** 3. ed. Porto Alegre: Bookman, 2004. ISBN 978-85-7679-003-1.
- FOWLER, M.; RICE, D.; FOEMMEL, M.; HEATT, E.; MEE, R.; STAFFORD, R. **Patterns of Enterprise Application Architecture.** [S.l.]: Addison-Wesley, 2002. ISBN 978-0321127426.
- FusionAuth. **Components of JWTs Explained.** 2025. Disponível em: <<https://fusionauth.io/articles/tokens/jwt-components-explained>>. Acesso em: Dezembro de 2025.
- GitHub. **Octoverse 2025: Top Programming Languages.** 2025. Disponível em: <<https://github.blog/news-insights/octoverse/octoverse-a-new-developer-joins-github-every-second-as-ai-leads-typescript-to-1/>>. Acesso em: Dezembro de 2025.
- Google Developers. **Pontuação de desempenho do Lighthouse.** 2019. Disponível em: <<https://developer.chrome.com/docs/lighthouse/performance/performance-scoring>>. Acesso em: Janeiro de 2026.
- Google Developers. **Introdução ao Lighthouse.** 2025. Disponível em: <<https://developer.chrome.com/docs/lighthouse/overview>>. Acesso em: Janeiro de 2026.
- Google Developers. **Pontuação de acessibilidade do Lighthouse.** 2025. Disponível em: <<https://developer.chrome.com/docs/lighthouse/accessibility/scoring?hl=pt-br>>. Acesso em: Janeiro de 2026.
- Google Developers. **Sobre o PageSpeed Insights.** 2025. Disponível em: <<https://developers.google.com/speed/docs/insights/v5/about>>. Acesso em: Janeiro de 2026.
- GRIGORIK, I. **High Performance Browser Networking: What Every Web Developer Should Know About Networking and Web Performance.** [S.l.]: O'Reilly Media, Inc., 2013.
- HOGAN, L. C. **Designing for Performance: Weighing Aesthetics and Speed.** [S.l.]: O'Reilly Media, Inc., 2014.

KUNSCH, M. M. K. A. Comunicação organizacional na era digital: contextos, percursos e possibilidades. **Signo y Pensamiento**, scieloco, p. 38 – 51, 12 2007. ISSN 0120-4823. Disponível em: <http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-48232007000200005&nrm=iso>.

LEAL, A. P.; LEAL, A. M. d. M. **Scholar: Desenvolvimento de um aplicativo móvel genérico de apoio acadêmico a estudantes em universidades**. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) — Universidade Federal de Santa Catarina, Araranguá, 2019. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/202733>>. Acesso em: Dezembro de 2025.

LIMA, P. N. L. **Aplicativo acadêmico móvel**. Trabalho de Conclusão de Curso — Universidade Federal do Ceará, Sobral, 2018. Disponível em: <<https://repositorio.ufc.br/handle/riufc/41996>>. Acesso em: Dezembro de 2025.

Medium. **(JWT) JSON Web Token**. 2017. Disponível em: <<https://programadriano.medium.com/json-web-token-jwt-c469834849a8>>. Acesso em: Dezembro de 2025.

Meta Platforms. **React: The library for web and native user interfaces**. 2025. Disponível em: <<https://react.dev/>>. Acesso em: Dezembro de 2025.

Microsoft. **TypeScript Documentation**. 2025. Disponível em: <<https://www.typescriptlang.org/docs/>>. Acesso em: Dezembro de 2025.

ONORIO, M.; BARRETO, R. L. **IFES Meet: desenvolvimento de uma plataforma para otimização e gerenciamento de eventos institucionais**. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) — Instituto Federal do Espírito Santo, Vitória, 2025. Disponível em: <<https://repositorio.ifes.edu.br/handle/123456789/6681>>. Acesso em: Dezembro de 2025.

PATI, S.; ZAKI, Y. **Evaluating the Efficacy of Next.js: A Comparative Analysis with React.js on Performance, SEO, and Global Network Equity**. 2025. Disponível em: <<https://arxiv.org/abs/2502.15707>>. Acesso em: Dezembro de 2025.

PEYROTT, S. **The JWT Handbook**. Auth0, 2024. Versão 0.14.2. Disponível em: <<https://auth0.com/resources/ebooks/jwt-handbook>>. Acesso em: Dezembro de 2025.

PostgreSQL Global Development Group. **PostgreSQL 18.1 Documentation**. 2025. Disponível em: <<https://www.postgresql.org/docs/>>. Acesso em: Dezembro de 2025.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de Software: Uma Abordagem Profissional**. 8. ed. Porto Alegre: McGraw-Hill Education / Artmed, 2016. ISBN 978-85-8055-533-2.

Prisma Data. **Prisma Documentation**. 2025. Disponível em: <<https://www.prisma.io/docs>>. Acesso em: Dezembro de 2025.

Prisma Data. **Prisma Pricing**. 2025. Disponível em: <<https://www.prisma.io/pricing>>. Acesso em: Dezembro de 2025.

prismic. **What is Client-side Rendering (CSR)**. 2024. Disponível em: <<https://prismic.io/blog/client-side-rendering>>. Acesso em: Dezembro de 2025.

PROGEP. **Quadro Quantitativo de Servidores – Abril/2019**. 2019. Disponível em: <<https://progep.ufc.br/wp-content/uploads/2019/05/quantvo-servidores-abril-2019.pdf>>. Acesso em: Dezembro de 2025.

RAMAKRISHNAN, R.; GEHRKE, J. **Database Management Systems**. 2nd. ed. New York, NY: McGraw-Hill Higher Education, 2000. ISBN 0-07-246535-2.

Shadcn. **Shadcn/ui: Build your component library**. 2025. Disponível em: <<https://ui.shadcn.com/>>. Acesso em: Dezembro de 2025.

SILVA, R. d. L. T. **Desenvolvimento de um sistema de gerenciamento de eventos para grupos PET**. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) — Universidade Federal de Santa Maria, Santa Maria, 2022. Disponível em: <<https://repositorio.ufsm.br/handle/1/26172>>. Acesso em: Dezembro de 2025.

SMITH, J. P. **Entity Framework Core in Action, Second Edition**. [S.l.]: Manning Publications, 2021. ISBN 978-1617298406.

SOMMERVILLE, I. **Engenharia de Software**. 10. ed. São Paulo: Pearson, 2016. ISBN 978-85-430-1129-3.

STUTTARD, D.; PINTO, M. **The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws**. 2. ed. [S.l.]: John Wiley & Sons, 2011. ISBN 9781118026472.

Tailwind Labs. **Tailwind CSS Documentation**. 2025. Disponível em: <<https://tailwindcss.com/docs>>. Acesso em: Dezembro de 2025.

UFC. **Censo comprova crescimento da UFC nos principais indicadores de qualidade**. 2019. Disponível em: <<https://www.ufc.br/noticias/13031-censo-comprova-crescimento-da-ufc-nos-principais-indicadores-de-qualidade>>. Acesso em: Dezembro de 2025.

Vercel. **Next.js Documentation**. 2025. Disponível em: <<https://nextjs.org/docs>>. Acesso em: Dezembro de 2025.

Vercel. **Vercel Functions Limits**. 2025. Disponível em: <<https://vercel.com/docs/functions/limitations>>. Acesso em: Dezembro de 2025.