

**UNIVERSIDADE PAULISTA**

**EDUARDO EUFRÁSIO JÚNIOR – N789530**

**GUSTAVO GRILO DEVIDES – N863757**

**JUAN CAMILO MARROQUIN PUENTES – N888695**

**LUCAS CORREIA DA SILVA – G4537H1**

**RENAN KOVACHICH DE OLIVEIRA REIS – N839589**

**DESENVOLVIMENTO DE UM WEBSITE PARA GESTÃO E RESERVA DE  
ESPAÇOS**

**SÃO PAULO**

**2025**

EDUARDO EUFRÁSIO JÚNIOR – N789530  
GUSTAVO GRILO DEVIDES – N863757  
JUAN CAMILO MARROQUIN PUENTES – N888695  
LUCAS CORREIA DA SILVA – G4537H1  
RENAN KOVACHICH DE OLIVEIRA REIS – N839589

DESENVOLVIMENTO DE UM WEBSITE PARA GESTÃO E RESERVA DE  
ESPAÇOS

Trabalho de conclusão de curso para  
obtenção do título de graduação em  
Ciência da Computação apresentado à  
Universidade Paulista – UNIP.

Orientador: Prof. Msc. Álvaro A. C. Prado.  
Coorientadora: Profa. Dra. Eliane Oliveira  
Santiago.

SÃO PAULO  
2025

EDUARDO EUFRÁSIO JÚNIOR – N789530  
GUSTAVO GRILO DEVIDES – N863757  
JUAN CAMILO MARROQUIN PUENTES – N888695  
LUCAS CORREIA DA SILVA – G4537H1  
RENAN KOVACHICH DE OLIVEIRA REIS – N839589

DESENVOLVIMENTO DE UM WEBSITE PARA GESTÃO E RESERVA DE  
ESPAÇOS

Trabalho de conclusão de curso para  
obtenção do título de graduação em  
Ciência da Computação apresentado à  
Universidade Paulista – UNIP.

Aprovado(a) em: \_\_\_\_/\_\_\_\_/\_\_\_\_

BANCA EXAMINADORA

---

Prof. ou Profa. Dr(a)/Me(a). xxxxxxxxxxxx  
Universidade Paulista - UNIP

---

Prof. ou Profa. Dr(a)/ Me(a). xxxxxxxxxxxx  
Universidade Paulista - UNIP

## DEDICATÓRIA

“A tecnologia move o mundo. Mas é a colaboração entre pessoas, unida a boas ideias, que transforma espaços comuns em lugares extraordinários.”

## **AGRADECIMENTOS**

Espaço opcional, o autor pode agradecer pessoas que contribuíram para elaboração de seu trabalho. Não recebe indicativo numérico. Elemento pré-textual complementar, não obrigatório.

## EPÍGRAFE

“A tecnologia move o mundo. Mas é a colaboração entre pessoas, unida a boas ideias, que transforma espaços comuns em lugares extraordinários.”

Bill Gates

## RESUMO

Este trabalho de conclusão de curso tem por objetivo desenvolver uma plataforma web denominada *LinkSpace*, voltada à gestão e reserva de espaços compartilhados em ambientes corporativos. O projeto parte da necessidade crescente de soluções que atendam ao modelo de trabalho híbrido, pelo qual empresas buscam otimizar o uso de seus escritórios por meio de ferramentas digitais eficientes. Para o desenvolvimento do sistema, foram utilizadas metodologias de engenharia de software, com levantamento de requisitos, modelagem de dados, desenvolvimento front-end e back-end, além de testes de usabilidade. A aplicação foi construída com foco na acessibilidade, segurança e experiência do usuário, permitindo que os colaboradores realizem reservas de mesas de forma intuitiva, com visualização dos espaços disponíveis em tempo real. Os resultados obtidos indicam que a plataforma oferece ganhos significativos em termos de organização interna, redução de conflitos por ocupação e aproveitamento mais racional dos ambientes físicos. O *LinkSpace* se mostra uma solução viável e escalável para empresas que desejam modernizar seus processos internos e acompanhar as transformações do ambiente corporativo contemporâneo. Considera-se que este projeto contribui para o avanço da digitalização nos escritórios, promovendo flexibilidade, autonomia e eficiência na gestão do espaço de trabalho.

**Palavras-chave:** gestão de espaços; reservas online; trabalho híbrido; ambiente corporativo; plataforma web.

## ABSTRACT

This undergraduate thesis aims to develop a web platform named *LinkSpace*, focused on managing and booking shared spaces in corporate environments. The project is based on the increasing need for solutions that support hybrid work models, where companies seek to optimize office use through efficient digital tools. The system was developed using software engineering methodologies, including requirement gathering, data modeling, frontend and backend development, as well as usability testing. The application was designed with an emphasis on accessibility, security, and user experience, allowing employees to book desks intuitively, with real-time visualization of available spaces. The results indicate that the platform provides significant improvements in internal organization, reduces booking conflicts, and promotes more rational use of physical environments. *LinkSpace* proves to be a viable and scalable solution for companies looking to modernize their internal processes and adapt to contemporary corporate transformations. It is concluded that this project contributes to the advancement of workplace digitalization, promoting flexibility, autonomy, and efficiency in space management.

**Keywords:** space management; online booking; hybrid work; corporate environment; web platform.



## LISTA DE FIGURAS

Figura 1 - Crescimento global de espaços de coworking entre 2018 e 2024 .....	18
Figura 2 - Método “Duplo Diamante” do Design Thinking.....	21
Figura 3 - Brad Neuberg em uma palestra .....	25
Figura 4 - Distribuição dos espaços coworking .....	26
Figura 5 - Diagrama de fluxo MVC .....	32
Figura 6 - Modelo lógico de dados .....	37
Figura 7 – Diagrama de uso .....	73
Figura 8 – UC01 .....	75
Figura 9 – UC02 .....	77
Figura 10 – UC03 .....	79
Figura 11 – UC04 .....	81
Figura 12 – UC05 .....	83
Figura 13 – UC06 .....	85
Figura 14 – UC07 .....	87
Figura 15 – UC08 .....	89
Figura 16 – UC09 .....	91
Figura 17 – UC10 .....	93
Figura 18 – UC11 .....	95
Figura 19 – UC12 .....	97
Figura 20 – Diagrama de Classe.....	103
Figura 21 – UC01 – Cadastrar Espaço .....	106
Figura 22 – UC02 – Editar Espaço.....	107
Figura 23 – Remover Espaço.....	108
Figura 24 – UC04 – Visualizar Espaços.....	109
Figura 25 – UC05 – Cadastrar Usuário .....	110
Figura 26 – UC06 - Editar Usuário .....	111
Figura 27 – UC07 – Remover Usuário .....	112
Figura 28 – UC08 – Visualizar Reservas.....	113
Figura 29 – UC09 - Reservar Espaços.....	114
Figura 30 – UC10 – Cancelar Reserva .....	115
Figura 31 – UC11 – Receber Confirmação de Reserva .....	116
Figura 32 – UC12 – Visualizar Histórico de Reservas.....	117
Figura 33 – Diagrama de Banco de Dados .....	119
Figura 34 – Página Inicial .....	125
Figura 35 – Página de Login .....	126
Figura 36 – Página Criar Conta.....	126
Figura 37 – Dashboard do Administrador .....	127
Figura 38 – Dashboard do Usuário Comum .....	128
Figura 39 – Página de Espaços (Administrador).....	128
Figura 40 – Página de Espaços (Usuário Comum) .....	129
Figura 41 – Página de Reservas (Administrador).....	130
Figura 42 – Página de Reservas (Usuário Comum).....	130
Figura 43 – Página de Usuários (Administrador).....	131
Figura 44 – Página de Configurações (Administrador).....	132
Figura 45 – Página de Configurações (Usuário Comum).....	132
Figura 46 – Cypress Teste 01.....	138
Figura 47 – Cypress Teste 02.....	139

Figura 48 – Cypress Teste 03.....	140
Figura 49 – Cypress Resultado Final .....	140
Figura 50 – Vitest Testes unitários.....	141
Figura 51 – Lighthouse Teste .....	142
Figura 52 – Lighthouse Acessibilidade .....	144

## **LISTA DE QUADROS**

LISTA DE QUADROS

## LISTA DE TABELAS

Tabela 1 - Benchmark de Ferramentas para Desenvolvimento do LinkSpace .....	30
Tabela 2 - Tecnologias e suas respectivas atuações.....	33
Tabela 3 - Tipos de SGBDR .....	38
Tabela 4 - Pacotes de Material UI (MUI) .....	41
Tabela 5 - Recursos e Funcionalidades Implementadas .....	44
Tabela 6 - Elementos do Material Design da Interface .....	49
Tabela 7 - Tecnologias para segurança do projeto .....	54
Tabela 8 – Histórico de Revisão .....	60
Tabela 9 – Resumo do envolvidos.....	62
Tabela 10 – Resumo dos Usuários.....	63
Tabela 11 – Requisitos Funcionais .....	64
Tabela 12 – Requisitos não funcionais .....	65
Tabela 13 – Requisitos Funcionais do Sistema .....	69
Tabela 14 – Requisitos não funcionais do sistema.....	70
Tabela 15 – CRC (Empresa) .....	120
Tabela 16 – CRC (Usuário) .....	121
Tabela 17 – CRC (Espaço).....	122
Tabela 18 – CRC (Reserva) .....	123
Tabela 19 – CRC (Notificação).....	124

## LISTA DE GRÁFICOS

LISTA DE GRÁFICOS

## **LISTA DE ABREVIATURAS E SIGLAS**

LISTA DE ABREVIATURAS E SIGLAS

## SUMÁRIO

1	INTRODUÇÃO .....	15
1.1	Contextualização .....	16
1.2	Problema .....	17
1.3	Justificativa .....	17
1.4	Objetivos .....	20
1.5	Metodologia aplicada na pesquisa .....	20
1.6	Estrutura do trabalho .....	23
2	REFERENCIAL TEÓRICO .....	24
2.1	Conceitos de Espaços Compartilhados .....	24
2.2	Gestão de Reservas Online.....	27
2.3	Tecnologias Web: Front-end e Back-end.....	31
2.4	Banco de Dados Relacionais.....	36
2.5	Frameworks e Ferramentas Utilizadas .....	39
2.6	Experiência do Usuário (UX/UI).....	46
2.7	Segurança da Informação em Aplicações Web .....	50
3	TRABALHOS RELACIONADOS .....	56
3.1	Criação de um website sobre tecnologias da informação e comunicação na área da saúde .....	56
3.2	Construindo Espaços de Interação Social a partir de Relações e Práticas de Trabalho Compartilhado .....	56
3.3	Compartilhamento de conhecimento entre empresas e suas consequências .....	57
3.4	Desenvolvimento de uma plataforma web para monitoramento das mídias sociais das empresas juniores da Universidade Federal de Ouro Preto .....	58
3.5	O uso do design responsivo no desenvolvimento de websites: aplicação dos métodos na construção de um website .....	59
4	Projeto de Software.....	60
4.1	Objetivo deste documento .....	60
4.2	Histórico de Revisão.....	60
4.3	Escopo do Produto .....	60
4.4	Não escopo do produto .....	61
4.5	Descrição dos envolvidos .....	61
4.6	Resumo dos Envolvidos .....	62
4.7	Resumo dos Usuários .....	63
4.8	Visão geral do produto .....	63
4.9	Requisitos funcionais.....	64
4.10	Requisitos não funcionais .....	65

4.11	Restrições .....	67
4.12	Política de versionamento .....	68
4.13	Documento de requisitos (Funcionais e não funcionais) .....	69
4.14	Diagramas de casos de uso .....	73
4.15	Arquitetura da Solução .....	99
4.16	Diagrama de classe e diagramas de sequência elaborados segundo a documentação dos casos de uso .....	103
4.17	Diagrama de Sequência .....	105
4.18	Projeto lógico de banco de dados MER/DER .....	118
4.19	Modelo CRC .....	120
4.20	Protótipo do projeto (de alta fidelidade) .....	125
5	TESTES E RESULTADOS .....	134
5.1	Estratégia e escopo .....	135
5.2	Ambiente de teste .....	136
5.3	Dados e critérios de aceite .....	136
5.4	Testes Funcionais (GUI e API) .....	137
5.5	Testes unitários e de integração .....	141
5.6	Testes E2E (navegador) .....	141
5.7	Testes de interface e não funcionais .....	142
6	REFERÊNCIAS BIBLIOGRÁFICAS .....	148
7	APÊNDICE (Caso necessário) .....	157



## 1 INTRODUÇÃO

Nas últimas décadas, os avanços nas Tecnologias da Informação e Comunicação (TICs) têm modificado substancialmente a dinâmica dos ambientes organizacionais, proporcionando maior integração entre sistemas, dados e pessoas. Essa transformação se intensificou com a popularização dos modelos de trabalho híbrido e remoto, os quais foram fortemente impulsionados pela pandemia da COVID-19 e consolidaram a necessidade de soluções digitais mais eficientes e acessíveis (MAIA, 2023; PICCINI et al., 2024).

Nesse contexto, destaca-se o papel estratégico das plataformas tecnológicas voltadas à gestão de ambientes físicos. A automatização dos processos de reserva e controle de espaços compartilhados representa não apenas uma modernização das práticas administrativas, mas também uma resposta concreta às demandas por flexibilidade, agilidade e produtividade no cotidiano organizacional (FERREIRA; RAMOS; TEIXEIRA, 2021). No entanto, muitos espaços ainda são gerenciados de forma manual ou com sistemas não adaptáveis, o que acarreta conflitos de agendamento, desperdício de recursos e baixa experiência do usuário (PICCINI et al., 2024).

Além disso, a crescente utilização de dispositivos móveis e a necessidade de acessibilidade reforçam a importância de soluções responsivas, intuitivas e multiplataforma, que sejam capazes de atender usuários com diferentes perfis e necessidades (CAMPOS, 2017). A usabilidade e a eficiência na interface se tornam fatores-chave para o sucesso da adesão ao sistema e para a garantia de sua eficácia no ambiente corporativo.

Neste cenário, o presente trabalho propõe o desenvolvimento do LinkSpace, um website funcional e responsivo destinado à gestão e reservas de espaços compartilhados. A plataforma busca aliar organização, praticidade e inovação, promovendo não apenas uma melhor experiência para os usuários, mas também contribuindo para o uso racional dos ambientes físicos e para a integração entre colaboradores. A proposta está alinhada com as tendências contemporâneas de

digitalização dos processos organizacionais e tem potencial de agregar valor a diferentes contextos empresariais.

### 1.1 Contextualização

O contexto contemporâneo de transformação digital tem impulsionado mudanças significativas na forma como as organizações operam e utilizam seus espaços físicos. A popularização dos modelos de trabalho híbrido e remoto, somada ao crescimento da economia colaborativa, evidencia a necessidade de ambientes flexíveis, tecnologicamente equipados e capazes de promover a interação social entre indivíduos e equipes. Nesse cenário, os espaços compartilhados, como *coworkings* e escritórios com áreas de uso comum, emergem como alternativas eficientes e inovadoras de ocupação (PICCINI et al., 2024).

O conceito de compartilhamento de ativos produtivos, aplicado aos espaços de trabalho, reforça a noção de que tais ambientes não devem apenas fornecer infraestrutura, mas também fomentar colaboração, aprendizado e construção de redes profissionais (Lasrado & Lugmayr, 2013; Moriset, 2014; ap.). A gestão eficiente desses ambientes, no entanto, demanda soluções digitais que otimizem a ocupação e promovam um uso racional dos recursos disponíveis.

As tecnologias da informação e comunicação (TICs) desempenham papel central nesse processo, atuando como facilitadoras na gestão de conhecimento e na integração entre usuários. Estudos demonstram que plataformas digitais próprias e ferramentas interativas, como WhatsApp e redes sociais, são amplamente utilizadas em *coworkings* para reserva de espaços, divulgação de eventos e conexão entre membros, inclusive entre filiais distantes (SIMEONOVA, 2018; IGLESIAS-PRADAS et al., 2017).

Contudo, conforme apontado por Soto-Acosta, Cegarra-Navarro e Garcia-Perez (2017), a adoção eficaz dessas tecnologias não depende exclusivamente de sua disponibilidade, mas sim do engajamento ativo dos usuários, da clareza na interface e da adequação ao perfil de uso. A ausência desses elementos pode tornar a ferramenta obsoleta, mesmo que tecnicamente robusta.

Nesse contexto, é essencial que plataformas de gestão de espaços compartilhados priorizem a usabilidade, a acessibilidade e a responsividade, garantindo que a experiência do usuário seja satisfatória e intuitiva, independentemente do dispositivo utilizado. Tais critérios não apenas ampliam a adesão ao sistema, como também promovem maior eficiência na organização e no aproveitamento dos espaços corporativos e colaborativos.

## **1.2 Problema**

Apesar do avanço tecnológico e da disseminação de modelos de trabalho mais flexíveis, muitas organizações ainda enfrentam dificuldades na gestão eficaz de seus espaços físicos compartilhados. A ausência de ferramentas digitais apropriadas para reserva de mesas, salas ou estações de trabalho resulta em conflitos de agendamento, baixa ocupação, desperdício de recursos e insatisfação entre os usuários (PICCINI et al., 2024). Além disso, quando tais sistemas existem, frequentemente apresentam interfaces pouco intuitivas, ausência de responsividade para dispositivos móveis ou dificuldade de acesso por pessoas com diferentes níveis de letramento digital (CAMPOS, 2017).

Segundo Maia (2023), a falta de soluções centradas na experiência do usuário compromete diretamente a aderência e a eficácia dos sistemas, reduzindo seu impacto positivo no ambiente organizacional. Diante disso, questiona-se: como desenvolver uma plataforma digital acessível, intuitiva e responsiva que otimize a gestão e a reserva de espaços compartilhados em ambientes corporativos e colaborativos, promovendo organização, engajamento e eficiência no uso do espaço?

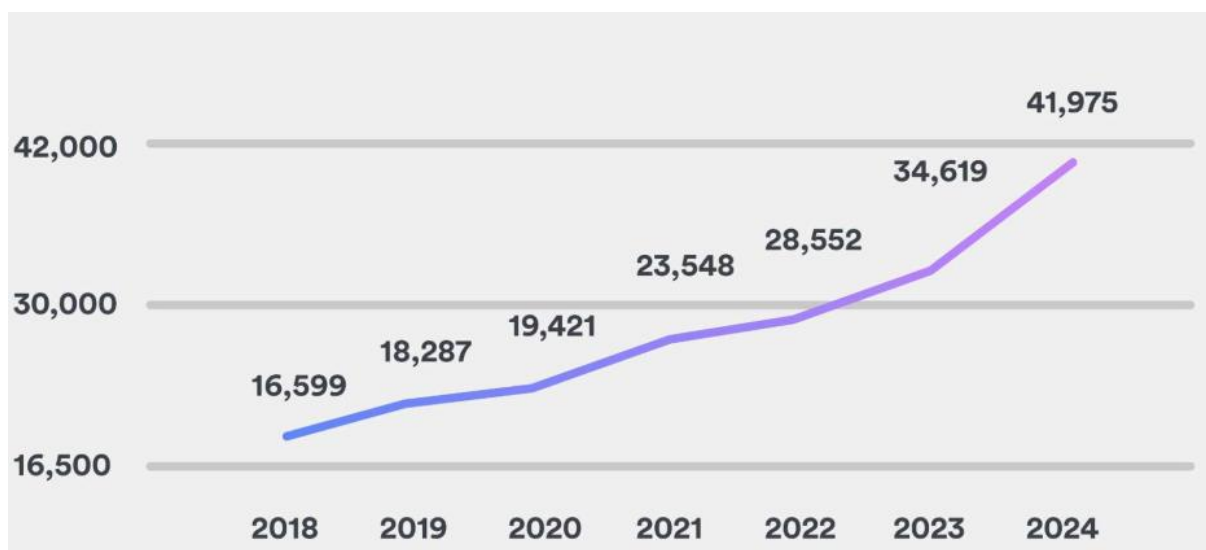
## **1.3 Justificativa**

O crescimento acelerado da economia compartilhada e o surgimento de novas formas de ocupação dos espaços físicos, como os ambientes de coworking, têm demandado soluções digitais que otimizem a gestão e a reserva de espaços compartilhados. Segundo pesquisa da Coworking Brazil (2023), o Brasil possui mais de 1.700 espaços de coworking, sendo o maior mercado da América Latina nesse

setor, e com expectativa de crescimento de 15% ao ano. Este cenário revela uma demanda crescente por sistemas inteligentes e acessíveis que organizem os fluxos de reservas, promovam o uso eficiente dos recursos e melhorem a experiência dos usuários desses ambientes.

Apesar da popularização dos espaços de coworking, ainda é comum o uso de planilhas, e-mails e agendas manuais para controle de horários e usuários, o que resulta em conflitos de agendamento, baixa eficiência operacional e insatisfação dos frequentadores (Boucken & Reuschl, 2018). Essa realidade aponta para uma lacuna tecnológica importante que pode ser preenchida com plataformas digitais especializadas, desenvolvidas sob princípios de usabilidade, escalabilidade e segurança da informação (Alavi & Leidner, 2001).

Figura 1 - Crescimento global de espaços de coworking entre 2018 e 2024



Fonte: Coworking Resources (2024)

A importância de sistemas de reservas automatizados é reforçada por Piccini, Behr e Schiavi (2024), ao destacarem que o uso de plataformas digitais próprias e mídias interativas, como WhatsApp e Instagram, são as principais ferramentas de suporte à interação e ao compartilhamento de conhecimento entre usuários de espaços compartilhados. No entanto, os autores também apontam a carência de soluções integradas que consigam reunir funcionalidades específicas como: controle

de permissões, análise de ocupação, relatórios gerenciais e integração com calendários e meios de pagamento.

Do ponto de vista acadêmico, o desenvolvimento de um website voltado à gestão e reservas de espaços compartilhados representa uma contribuição significativa para o campo da Computação, ao aliar fundamentos de engenharia de software, banco de dados, experiência do usuário (UX/UI) e segurança da informação. Conforme Santos (2019), aplicações web voltadas à gestão de informações estratégicas são capazes de transformar dados brutos em decisões operacionais com alto impacto nas organizações. Assim, o projeto proposto também se alinha aos princípios da inteligência operacional e automação de processos, temas de relevância crescente em ambientes corporativos e educacionais.

Além disso, a escolha por uma aplicação web se justifica pelo alcance multiplataforma, pela facilidade de manutenção e pela possibilidade de integração com APIs e serviços externos, como gateways de pagamento e sistemas de autenticação. Segundo Campos (2017), o design responsivo é essencial para garantir que a aplicação ofereça boa experiência de uso em dispositivos variados, como smartphones, tablets e desktops, o que reforça a aderência do projeto às tendências atuais de acessibilidade e mobilidade.

Do ponto de vista social e econômico, o presente trabalho pode auxiliar empreendedores, gestores de espaços culturais, educacionais e corporativos, bem como usuários em geral, a lidarem com a reserva de ambientes de forma mais organizada, transparente e segura. Isso contribui para o fortalecimento da cultura colaborativa, para a democratização do acesso à infraestrutura de trabalho e para o estímulo à inovação por meio do uso consciente e eficiente dos espaços urbanos.

Portanto, este projeto se justifica não apenas pela lacuna tecnológica identificada e pelas evidências de mercado, mas também pelo potencial de impacto positivo na experiência dos usuários e na gestão de ambientes compartilhados. A pesquisa e o desenvolvimento da solução proposta dialogam com os Objetivos de Desenvolvimento Sustentável da ONU, especialmente o ODS 9 (Indústria, Inovação e

Infraestrutura) e o ODS 11 (Cidades e Comunidades Sustentáveis), ao promoverem o uso inteligente de recursos por meio da tecnologia.

## **1.4 Objetivos**

### **1.4.1 Objetivo geral**

Desenvolver uma plataforma web funcional e responsiva para gestão e reservas de espaços compartilhados, visando maior eficiência, usabilidade e organização.

### **1.4.2 Objetivos específicos**

- Projetar uma interface intuitiva e acessível;
- Implementar funcionalidades de controle e visualização de reservas;
- Garantir compatibilidade com múltiplos dispositivos.

## **1.5 Metodologia aplicada na pesquisa**

A construção da plataforma LinkSpace baseou-se em uma abordagem metodológica centrada no usuário, combinando elementos do Design Thinking, do modelo Duplo Diamante e de princípios do design de experiência do usuário (UX). Essa combinação metodológica visa garantir que o sistema desenvolvido atenda de forma precisa às reais necessidades dos usuários, além de ser funcional, acessível e visualmente intuitivo.

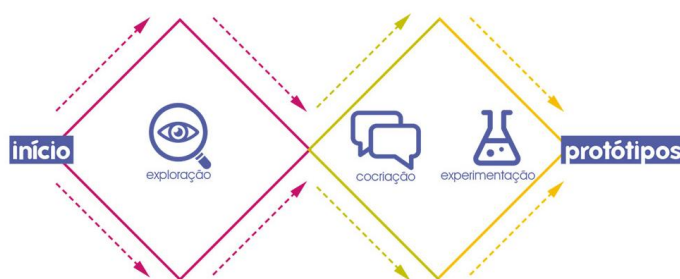
A metodologia adotada neste estudo é de natureza aplicada, com abordagem qualitativa e caráter exploratório-descritivo. Segundo Polit e Beck (2011), pesquisas metodológicas promovem a criação e avaliação de instrumentos tecnológicos voltados à solução de problemas práticos, sendo especialmente úteis no desenvolvimento de plataformas digitais inovadoras.

### 1.5.1 Design Thinking e Duplo Diamante

O processo de Design Thinking foi estruturado em três espaços sobrepostos, conforme proposto por Brown e Wyatt (2009): inspiração, ideação e implementação. A fase de inspiração consistiu na identificação do problema e levantamento de requisitos a partir da análise do cenário atual de gestão de espaços compartilhados. Em seguida, a etapa de ideação permitiu a geração e o refinamento de ideias por meio da prototipação e do brainstorming colaborativo. Por fim, a fase de implementação englobou o desenvolvimento e a validação da solução proposta, com foco na experiência do usuário.

Complementarmente, aplicou-se o modelo Duplo Diamante, estruturado nas fases: descobrir, definir, desenvolver e entregar, conforme delineado pelo Design Council (2015). Esse modelo contribuiu para organizar o processo de forma iterativa, alternando entre pensamento divergente e convergente, favorecendo a exploração de possibilidades e a definição precisa da solução a ser construída.

Figura 2 - Método “Duplo Diamante” do Design Thinking



Fonte: Design Council – UK 2005

### 1.5.2 Etapas do desenvolvimento

A primeira etapa, descoberta, envolveu o levantamento do problema e a identificação das necessidades dos usuários por meio de análises contextuais, estudos de caso e mapeamento de jornadas de uso. Com base nesses dados, definiu-se a persona representativa do público-alvo da plataforma, conforme prática recomendada por Garrett (2011), o que permitiu orientar decisões de design com base em perfis reais de uso.

Na etapa definir, foram consolidadas as descobertas anteriores e delineados os requisitos funcionais e não funcionais do sistema. Os fluxos de navegação, a arquitetura da informação e o escopo das funcionalidades foram estabelecidos com base em ferramentas visuais como mapas de site e wireframes.

Durante a fase de desenvolvimento, foram utilizados princípios do design responsivo (CAMPOS, 2017) e técnicas de prototipação iterativa com feedback contínuo. A aplicação foi construída com tecnologias web modernas, garantindo compatibilidade com dispositivos móveis e navegadores diversos. O método de Garrett (2011) serviu como referência para estruturar os cinco planos de desenvolvimento: estratégico, escopo, estrutura, esqueleto e superfície, assegurando coesão entre objetivos organizacionais e experiência do usuário final.

Por fim, na etapa de entrega, foram realizados testes exploratórios com usuários simulados. A validação envolveu verificação de usabilidade, responsividade e adequação das funcionalidades. Os testes seguiram critérios de Nielsen (1994) e Krug (2014), que destacam a importância de sistemas “autoexplicativos” e de fácil interação.

### 1.5.3 Considerações metodológicas

A escolha por uma abordagem iterativa e centrada no usuário foi fundamental para alinhar o desenvolvimento da solução às reais demandas do ambiente organizacional contemporâneo. O Design Thinking, conforme destacado por Brown (2009), oferece um arcabouço eficaz para promover inovação prática, ao integrar aspectos técnicos, sociais e humanos em todas as fases do projeto.

Assim, a metodologia aplicada neste trabalho não apenas orientou a criação da plataforma LinkSpace, mas também serviu como instrumento de investigação e validação da eficácia de soluções digitais no contexto da gestão de espaços compartilhados.



## **1.6 Estrutura do trabalho**

Este trabalho está estruturado em cinco capítulos, organizados de forma a garantir clareza na exposição dos fundamentos teóricos, metodológicos e práticos que embasam o desenvolvimento da plataforma LinkSpace.

O Capítulo 1 – Introdução apresenta o tema, contextualiza o problema de pesquisa, delimita os objetivos e discorre sobre a motivação e relevância do projeto.

O Capítulo 2 – Objetivos descreve, de forma clara e objetiva, o objetivo geral e os objetivos específicos do trabalho, além de apresentar a estrutura adotada na organização do estudo.

O Capítulo 3 – Metodologia aplicada na pesquisa detalha os métodos utilizados para a concepção e desenvolvimento da solução proposta, incluindo os referenciais adotados na abordagem centrada no usuário e nas etapas de design e implementação da plataforma.

O Capítulo 4 – Revisão bibliográfica reúne os principais conceitos e estudos relacionados à gestão de espaços compartilhados, usabilidade, design responsivo, tecnologias web e plataformas digitais, servindo como base teórica para a proposta desenvolvida.

Por fim, o Capítulo 5 – Desenvolvimento da solução e considerações finais apresenta as etapas práticas de construção do sistema, os resultados obtidos, as limitações identificadas e sugestões para trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

### 2.1 Conceitos de Espaços Compartilhados

O conceito de espaços compartilhados está diretamente relacionado às transformações nas formas de trabalho, estudo e convívio urbano características da sociedade contemporânea. Também denominados *shared spaces*, esses ambientes são projetados com o objetivo de otimizar a infraestrutura física, reduzir custos operacionais, estimular a colaboração e promover a inovação organizacional (MOROZOV; BRIA, 2018). Segundo Castells (1999), vivemos em uma era de redes, na qual os fluxos de informação, pessoas e conhecimento ultrapassam barreiras físicas, exigindo novas formas de organização do espaço e do trabalho.

Tais ambientes são caracterizados pela coexistência de múltiplos indivíduos ou organizações em um mesmo local, utilizado para fins diversos como atividades profissionais, educacionais, culturais e comunitárias (OLIVEIRA; FONSECA, 2020). Essa configuração favorece tanto a flexibilidade quanto a interdisciplinaridade, uma vez que estimula interações espontâneas e trocas de saberes, fundamentais para contextos que demandam inovação constante (BENCHIMOL, 2019).

Assim, os “*shared spaces*” representam uma resposta concreta às demandas de um mundo mais conectado, urbano e colaborativo, sendo aplicáveis a *coworkings*, laboratórios, centros culturais, escolas e outros espaços que valorizam o uso racional e integrado da infraestrutura física.

#### 2.1.1 Origem e evolução

Historicamente, a ideia de compartilhamento de espaços remete a modelos de convivência já adotados em comunidades, bibliotecas e universidades. No entanto, foi com o advento dos *coworkings* que os espaços compartilhados se consolidaram como um fenômeno global. Segundo Gandini (2015), os *coworkings* representam uma ruptura com os tradicionais modelos de trabalho fixo, promovendo uma lógica de flexibilidade e *networking* como pilares centrais.

O primeiro espaço de coworking foi fundado em 2005, em São Francisco (EUA), por Brad Neuberg, e tinha como foco profissionais autônomos e freelancers. Desde então, o modelo evoluiu e passou a abranger startups, pequenas empresas, grandes corporações e até órgãos públicos.

Figura 3 - Brad Neuberg em uma palestra



Fonte: Coworking Resources (2023)

### 2.1.2 Tipos de espaços compartilhados

Os espaços compartilhados podem assumir diferentes formas, a depender do público-alvo e da finalidade a que se destinam. Coworkings, por exemplo, são ambientes voltados ao trabalho profissional, à inovação e ao empreendedorismo, reunindo profissionais autônomos, startups e pequenas empresas em um mesmo local. Já os chamados espaços *maker* se caracterizam pela disponibilidade de ferramentas para prototipagem e fabricação digital, como impressoras 3D e cortadoras a laser, favorecendo a experimentação prática.

Outro tipo bastante comum são os laboratórios de inovação, conhecidos como *labs*, que se constituem como ambientes voltados à experimentação e à criação colaborativa, muitas vezes integrando equipes multidisciplinares. Os centros comunitários, por sua vez, são destinados à realização de eventos, cursos e atividades sociais, desempenhando um papel importante de integração e desenvolvimento local. Além disso, há também as salas multiuso e auditórios, frequentemente presentes em empresas ou instituições públicas, cuja gestão e reserva são geralmente feitas por meio de plataformas digitais.

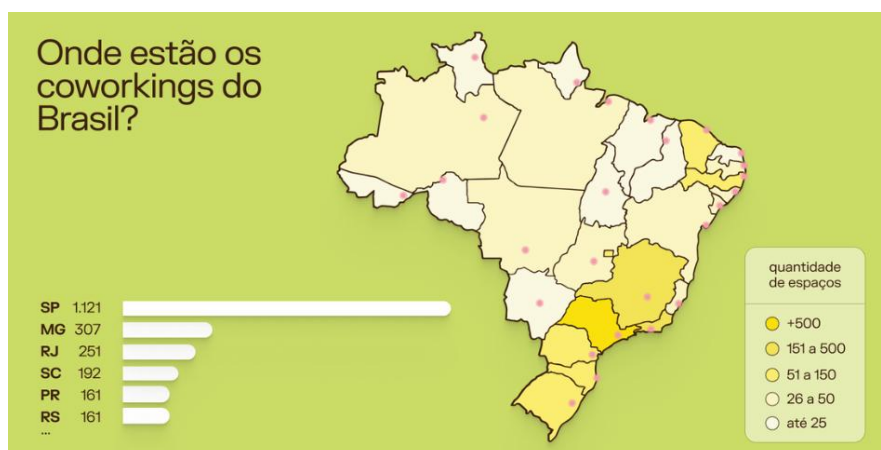
Esses diferentes espaços desempenham um papel central na constituição de ecossistemas de inovação, ao promoverem a interação entre pessoas, ideias e tecnologias. Como destacam Benitez, Ayala e Frank (2020), os ecossistemas podem se consolidar e evoluir ao compreender como o valor é cocriado dentro deles. À medida que a confiança e o compromisso — trocas sociais — crescem, a estrutura de poder passa da centralidade da associação empresarial para um mecanismo de coordenação neutra dos projetos.

### 2.1.3 Panorama no Brasil

No Brasil, o movimento de expansão dos espaços compartilhados tem ganhado destaque. De acordo com a Associação Brasileira de Coworkings (2023), o país já conta com mais de 2.000 unidades desse tipo, concentradas principalmente nas regiões Sudeste e Sul. Observa-se também a adoção crescente de modelos híbridos, nos quais empresas tradicionais transferem parte de suas operações para esses ambientes mais flexíveis e colaborativos.

Nos setores educacional e governamental, o uso compartilhado de espaços tem como objetivo ampliar o acesso à infraestrutura, otimizar custos e incentivar a interdisciplinaridade. Essa tendência se alinha aos princípios das *smart cities*, onde a digitalização e a gestão inteligente são utilizadas para maximizar o uso de recursos urbanos e aprimorar a oferta de serviços públicos.

Figura 4 - Distribuição dos espaços coworking



Fonte: Censo Coworking by Woba 2025

#### 2.1.4 Relação com o projeto LinkSpace

No contexto do presente trabalho, o conceito de espaços compartilhados constitui a base para o desenvolvimento da plataforma LinkSpace. O sistema é direcionado à gestão e reserva de ambientes físicos em instituições públicas ou privadas, possibilitando um uso mais eficiente, transparente e democrático da infraestrutura disponível. Como apontam Oliveira e Fonseca (2020), ambientes compartilhados exigem soluções tecnológicas que garantam não apenas o controle de uso, mas também promovam acesso igualitário aos recursos institucionais.

Além de facilitar a alocação e o monitoramento do uso dos espaços, a plataforma LinkSpace busca estimular a integração entre usuários de diferentes áreas, favorecendo a colaboração interdisciplinar e ampliando o potencial de inovação organizacional (BENCHIMOL, 2019). Essa abordagem está alinhada às diretrizes de modernização administrativa que propõem a adoção de tecnologias para tornar os processos mais ágeis, rastreáveis e baseados em dados (BRASIL, 2020).

A implementação de um sistema web de reservas, como o proposto, contribui diretamente para a transformação digital da gestão patrimonial, promovendo o uso racional dos espaços disponíveis e fortalecendo a governança sobre bens públicos ou privados. Segundo Morozov e Bria (2018), sistemas digitais voltados à infraestrutura urbana e institucional são fundamentais para promover transparência e eficiência no uso dos recursos, especialmente em ambientes coletivos.

## 2.2 Gestão de Reservas Online

A gestão de reservas online tem se consolidado como uma prática indispensável para a administração eficiente de espaços físicos, especialmente em contextos nos quais há alta rotatividade de uso, múltiplos usuários e limitação de recursos. Essa prática é parte do movimento mais amplo de transformação digital da administração de infraestruturas institucionais e urbanas (MOROZOV; BRIA, 2018).

Trata-se da digitalização do processo de agendamento, controle e acompanhamento do uso de ambientes como salas de reunião, auditórios,

laboratórios, coworkings e demais instalações compartilhadas. Essa digitalização se dá por meio de plataformas web ou mobile, que tornam o processo mais acessível, transparente e rastreável, reduzindo conflitos de agendamento e melhorando o aproveitamento dos espaços (ALMEIDA; SANTOS, 2021).

De acordo com Silva e Pereira (2020), a adoção de sistemas de reserva online promove ganhos significativos em produtividade, governança e gestão patrimonial, uma vez que permite o acompanhamento em tempo real do uso dos ambientes, a geração de relatórios e a automatização de tarefas antes realizadas de forma manual. Isso reforça a importância da tecnologia como ferramenta estratégica para instituições que buscam eficiência e otimização de recursos.

#### 2.2.1 Transformação digital e eficiência operacional

O avanço da transformação digital no setor público e privado exige soluções que otimizem tempo, reduzam burocracias e promovam a transparência na utilização dos recursos. Sistemas de reservas online integram banco de dados, calendários inteligentes, notificações e relatórios automáticos, reduzindo erros humanos e evitando conflitos de agendamento. Segundo estudo da McKinsey & Company (2020), organizações que adotam processos digitais para agendamento e gestão de espaços registram ganhos de eficiência de até 25% na administração de infraestrutura física.

#### 2.2.2 Vantagens dos sistemas de reservas online

De acordo com Maia (2023), ao implementar um website para gestão de TICs na saúde pública, foi possível ampliar o acesso dos servidores a recursos antes subutilizados. Esse exemplo é transponível para qualquer instituição que lida com múltiplos ambientes, pois a gestão informatizada oferece inúmeras vantagens. Entre elas, destaca-se a automatização de processos, que reduz a necessidade de controle manual por meio de planilhas ou formulários físicos (Maia, 2023). Outro benefício importante é o acesso remoto, permitindo que os usuários consultem a disponibilidade e realizem reservas de qualquer dispositivo conectado à internet (Maia, 2023).

Além disso, a eliminação de conflitos de horário torna-se viável com o uso de calendários integrados e bloqueios automáticos, os quais impedem reservas simultâneas e otimizam o uso dos espaços (Maia, 2023). A geração de relatórios de uso também se mostra essencial, pois possibilita o acompanhamento de métricas como taxa de ocupação, frequência por tipo de espaço e tempo médio de utilização, oferecendo dados relevantes para a tomada de decisão (Maia, 2023). Por fim, a segurança e rastreabilidade são reforçadas por registros digitais, que garantem um histórico completo das reservas, com a devida identificação dos responsáveis, promovendo maior transparência e controle (Maia, 2023).

### 2.2.3 Aplicações práticas

A gestão online de reservas é uma prática amplamente difundida em diversos contextos organizacionais. Em universidades, é utilizada para o agendamento de salas de aula, laboratórios e auditórios, otimizando a ocupação dos espaços e facilitando o planejamento acadêmico (SOUZA; RODRIGUES, 2020). No contexto corporativo, empresas adotam sistemas de reserva para o controle de uso de salas de reunião e áreas comuns, promovendo maior organização e evitando conflitos de uso entre equipes (COSTA; FERREIRA, 2019).

Espaços de coworking também se beneficiam de plataformas digitais para gerenciar estações de trabalho, estúdios ou cabines privativas, proporcionando uma experiência mais flexível e transparente para os usuários (ALMEIDA; SANTOS, 2021). Já em órgãos públicos e organizações não governamentais (ONGs), a gestão digital de espaços tem se mostrado eficiente para a otimização de recursos compartilhados entre departamentos, contribuindo para maior racionalidade no uso da infraestrutura (BRASIL, 2020).

Em todos esses casos, a digitalização dos processos de reserva contribui diretamente para a agilidade operacional, a equitatividade no acesso aos recursos e para a formação de uma base de dados estratégica, capaz de subsidiar decisões gerenciais sobre a infraestrutura disponível (SOMMERVILLE, 2011).

## 2.2.4 Tecnologias e ferramentas adotadas

Entre as ferramentas mais utilizadas para essa finalidade estão:

Tabela 1 - Benchmark de Ferramentas para Desenvolvimento do LinkSpace

<b>Ferramenta/Plataforma</b>	<b>Recursos Principais</b>	<b>Tipo de Aplicação</b>
Google Calendar + Add-ons	Agendamento com notificações automáticas	Pequenas instituições e escolas
Microsoft Bookings	Integração com Microsoft 365, dashboards	Corporações e universidades
Skedda / Roomzilla	Interface responsiva, controle de permissões	Coworkings e espaços culturais
Sistemas personalizados	Integração total com necessidades específicas	Projetos como o LinkSpace

Fonte: Elaboração própria

Esses sistemas fazem uso de linguagens e frameworks modernos (ex: Django, Laravel, React, Vue.js) e bancos de dados como PostgreSQL e MySQL, para garantir escalabilidade, responsividade e integração com outros sistemas institucionais.

## 2.2.5 Contribuições ao projeto LinkSpace

A plataforma LinkSpace, proposta neste trabalho, apresenta um modelo moderno e acessível de sistema de reservas para espaços compartilhados. Sua concepção baseia-se em práticas recomendadas de design de software, priorizando a interface intuitiva e responsiva, que melhora a experiência do usuário e se adapta a diferentes dispositivos (GARRETT, 2011). O sistema inclui controle de acesso por perfil de usuário, com níveis diferenciados como administrador, gestor e usuário comum, garantindo segurança e personalização da navegação (SOMMERVILLE, 2011).

Um dos recursos centrais da plataforma é o calendário de disponibilidade em tempo real, que permite aos usuários verificar horários vagos e agendar com maior agilidade e precisão (PRESSMAN; MAXIM, 2016). A funcionalidade de notificações



por e-mail e dashboard mantém os usuários informados sobre o status de suas reservas e reduz falhas de comunicação (NIELSEN, 2021).

Além disso, o sistema oferece relatórios gerenciais exportáveis em formatos como PDF e Excel, permitindo a análise de métricas importantes, como taxa de uso dos espaços, frequência por usuário e padrões de agendamento (ALMEIDA; SANTOS, 2021). A plataforma foi pensada para atender tanto pequenas instituições quanto estruturas maiores, com possibilidade de implantação em nuvem e suporte a múltiplas unidades geográficas, facilitando a escalabilidade e a flexibilidade da solução (SILVA; PEREIRA, 2020).

### **2.3 Tecnologias Web: Front-end e Back-end**

O desenvolvimento de aplicações web modernas requer uma integração eficiente entre duas camadas essenciais: o front-end, responsável pela interface e experiência do usuário, e o back-end, que lida com o processamento, a lógica de negócios e o acesso a dados. O equilíbrio entre essas camadas é decisivo para garantir desempenho, usabilidade, segurança e escalabilidade, principalmente em plataformas como o LinkSpace, voltadas à gestão de reservas de espaços compartilhados, onde há múltiplos usuários acessando o sistema simultaneamente.

De acordo com Pressman e Maxim (2016), a divisão clara entre a interface e a lógica interna da aplicação permite maior modularidade, facilitando testes, manutenção e evolução do sistema. Sommerville (2011) complementa que, ao estruturar corretamente as camadas de software, torna-se possível aplicar boas práticas de segurança, controlar fluxos de dados e otimizar o desempenho de maneira consistente.

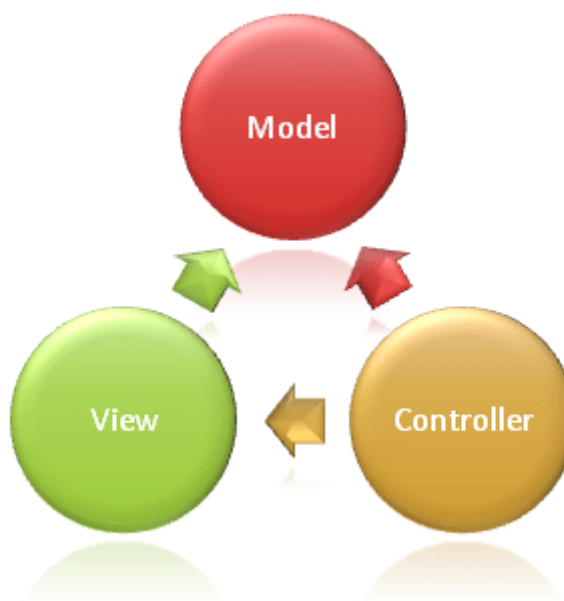
A camada de front-end deve ser projetada com foco na responsividade, acessibilidade e feedback em tempo real, utilizando tecnologias modernas como frameworks JavaScript e bibliotecas de componentes reutilizáveis (NIELSEN, 2021). Já o back-end deve ser capaz de gerenciar autenticação, autorização, verificação de disponibilidade e persistência dos dados com confiabilidade e segurança. Essa estrutura é essencial para garantir que sistemas como o LinkSpace consigam operar com estabilidade e flexibilidade, mesmo em ambientes institucionais complexos.

### 2.3.1 Conceitos e arquitetura

O front-end representa a parte visível da aplicação, ou seja, tudo que o usuário vê e interage em um navegador web. Essa camada é construída com linguagens como HTML, CSS e JavaScript, além de frameworks como React, Vue.js ou Angular, que oferecem estrutura modular e componentizada.

Já o back-end é responsável por processar as requisições do front-end, acessar bancos de dados, aplicar regras de negócio, autenticar usuários e manter a integridade do sistema. É desenvolvido com linguagens como Python (Django/Flask), PHP (Laravel), Java (Spring Boot), Node.js, entre outras.

Figura 5 - Diagrama de fluxo MVC



Fonte: Microsoft Learn (2025)

### 2.3.2 Principais tecnologias utilizadas

A arquitetura tecnológica de plataformas web completas frequentemente emprega um conjunto de tecnologias bem estabelecidas, organizadas em camadas distintas para otimizar o desenvolvimento e a manutenção. A Tabela 2 apresenta uma visão geral dessas tecnologias, categorizadas por suas respectivas camadas de atuação no sistema:

Tabela 2 - Tecnologias e suas respectivas atuações

Camada	Tecnologia / Ferramenta	Descrição
Front-end	React.js (com TypeScript)	Framework principal para construção da interface com tipagem estática.
	Material UI (MUI)	Sistema de design com componentes prontos, responsividade e tema customizável.
	React Router (react-router-dom)	Gerenciamento de rotas e navegação entre páginas SPA.
	React Query (TanStack Query)	Gerenciamento de estado assíncrono e cache de dados provenientes da API.
	Date-fns	Manipulação de datas com suporte a internacionalização.
	TypeScript, ESLint, Prettier	Qualidade de código: tipagem, linting e formatação.
Back-end	Javascript	A interface se conecta a uma API RESTful com autenticação via JWT.
Banco de Dados	PostgreSQL (recomendado)	Banco de dados relacional robusto, seguro e escalável.
Padrões	RESTful APIs	Comunicação entre client e servidor no padrão JSON por meio de HTTP verbs.
Arquitetura	Context API + Hooks personalizados	Compartilhamento e encapsulamento de lógica de estado global.

Fonte: elaboração do autor

### 2.3.3 Integração entre o Front-end e Back-end

A comunicação entre as camadas de front-end e back-end é viabilizada por meio de APIs (Application Programming Interfaces), comumente implementadas no formato RESTful. Essas APIs utilizam chamadas HTTP — como GET, POST, PUT e DELETE — para realizar a troca de dados no formato JSON. Essa abordagem permite uma separação clara entre o que acontece no lado do cliente (client-side) e no lado do servidor (server-side), o que facilita tanto a manutenção quanto o desenvolvimento simultâneo por equipes distintas.

No caso da plataforma LinkSpace, essa integração é refletida em diversos aspectos técnicos. O front-end consome diretamente a API REST disponibilizada pelo back-end, estabelecendo uma ponte eficiente para o envio e recebimento de dados. As rotas da aplicação são protegidas por mecanismos de autenticação baseados em JWT (JSON Web Token), garantindo segurança no acesso às funcionalidades. Além disso, a busca por disponibilidade de salas é realizada em tempo real, utilizando requisições assíncronas por meio de tecnologias como AJAX ou Fetch API, o que contribui para uma experiência de usuário mais fluida e responsiva.

A integração eficiente entre front-end e back-end é um dos pilares da arquitetura moderna de aplicações web. Segundo Fielding (2000), criador do estilo REST, "a uniformidade na interface entre os componentes de um sistema distribuído é essencial para permitir a evolução independente e o desacoplamento entre cliente e servidor". Esse princípio, quando aplicado corretamente, proporciona escalabilidade, robustez e flexibilidade no desenvolvimento de soluções digitais.

#### 2.3.4 Considerações sobre segurança e performance

A adoção de boas práticas de segurança é essencial no desenvolvimento de sistemas, sobretudo em aplicações que lidam com dados sensíveis de usuários e operações como reservas e autenticações. Entre as medidas indispensáveis, destaca-se o uso de HTTPS, por meio dos protocolos TLS/SSL, que garantem a criptografia do tráfego e protegem contra interceptações. A aplicação deve também ser protegida contra ataques comuns, como CSRF (Cross-Site Request Forgery) e XSS (Cross-Site Scripting), que exploram falhas na validação e execução de scripts maliciosos. Senhas devem ser armazenadas com algoritmos de hash robustos, como bcrypt ou PBKDF2, que dificultam ataques de força bruta. Além disso, a validação rigorosa dos dados enviados pelos usuários no lado do servidor é fundamental para evitar injeções e comportamentos indesejados. Medidas como rate limiting e controle de sessões ajudam a mitigar abusos, reduzir sobrecargas e reforçar a estabilidade do sistema.

Como salientam Anderson e Moore (2006), "a segurança deve ser considerada desde o projeto inicial do sistema, e não apenas como um complemento posterior", reforçando a importância do desenvolvimento seguro desde os primeiros estágios.

No que diz respeito à performance, a eficiência do sistema depende de uma série de práticas técnicas. No front-end, o uso de lazy loading para componentes permite que apenas os elementos necessários sejam carregados inicialmente, o que melhora o tempo de resposta percebido pelo usuário. No back-end, a otimização das consultas SQL evita gargalos no banco de dados e melhora a escalabilidade. Técnicas de caching, utilizando soluções como Redis ou Memcached, são fundamentais para acelerar o acesso a dados frequentemente solicitados. Por fim, a minificação e

compressão de arquivos estáticos, como CSS e JavaScript, contribuem para reduzir o tempo de carregamento e o uso de largura de banda.

De acordo com Ousterhout (2018), “o desempenho não é apenas uma questão de velocidade, mas de percepção do usuário sobre a eficiência e fluidez da aplicação”, o que reforça a importância de pensar a performance como parte integral da experiência.

### 2.3.5 Alinhamento com o LinkSpace

No projeto LinkSpace, as decisões tecnológicas foram cuidadosamente alinhadas aos requisitos de escalabilidade, segurança, usabilidade e manutenibilidade da aplicação. A camada de front-end foi desenvolvida com React.js integrado ao TypeScript, proporcionando uma estrutura robusta e tipada para construção de interfaces dinâmicas e seguras. O uso do Material UI (MUI) como sistema de design garante responsividade, personalização visual e aderência a padrões modernos de usabilidade. A navegação entre as páginas é gerenciada pelo React Router, permitindo a implementação fluida de rotas no estilo SPA (Single Page Application), enquanto a comunicação com o back-end e o gerenciamento do estado assíncrono são otimizados com o React Query (TanStack Query), que traz melhorias significativas em termos de performance e cache.

Para manipulação de datas com suporte à internacionalização e integração com componentes de data do MUI, foi adotada a biblioteca Date-fns. O projeto também incorpora práticas de qualidade de código com a utilização conjunta de TypeScript, ESLint e Prettier, assegurando padronização, legibilidade e prevenção de erros em tempo de desenvolvimento.

No que diz respeito ao back-end, a aplicação se comunica com uma API RESTful, que utiliza JavaScript como linguagem base, com autenticação via tokens JWT (JSON Web Token) para controle seguro de acesso. As chamadas entre cliente e servidor seguem o padrão REST, transmitindo dados no formato JSON por meio dos verbos HTTP.

O armazenamento das informações está a cargo do PostgreSQL, banco de dados relacional recomendado para o projeto devido à sua confiabilidade, capacidade de lidar com grandes volumes de dados e suporte a extensões geográficas, o que viabiliza, futuramente, a implementação de funcionalidades de geolocalização dos espaços compartilhados.

A arquitetura do sistema está estruturada sobre o modelo de componentização com Context API e hooks personalizados, promovendo encapsulamento de lógica e compartilhamento eficiente de estado global entre os módulos da aplicação. Além disso, o sistema está preparado para implantação em ambientes de nuvem como Heroku, Vercel ou AWS, o que facilita a escalabilidade horizontal e integração com pipelines de integração e entrega contínua (CI/CD), tornando o LinkSpace uma plataforma segura, modular e preparada para crescimento futuro.

## **2.4 Banco de Dados Relacionais**

Os bancos de dados relacionais (BDRs) são estruturados para armazenar informações em tabelas compostas por linhas e colunas, organizando os dados de forma lógica e estruturada. A integridade referencial entre essas tabelas é mantida por meio do uso de chaves primárias e estrangeiras, o que permite estabelecer relações coerentes e seguras entre os dados. A linguagem utilizada para manipular essas informações é a SQL (Structured Query Language), que possibilita desde simples consultas até operações transacionais complexas.

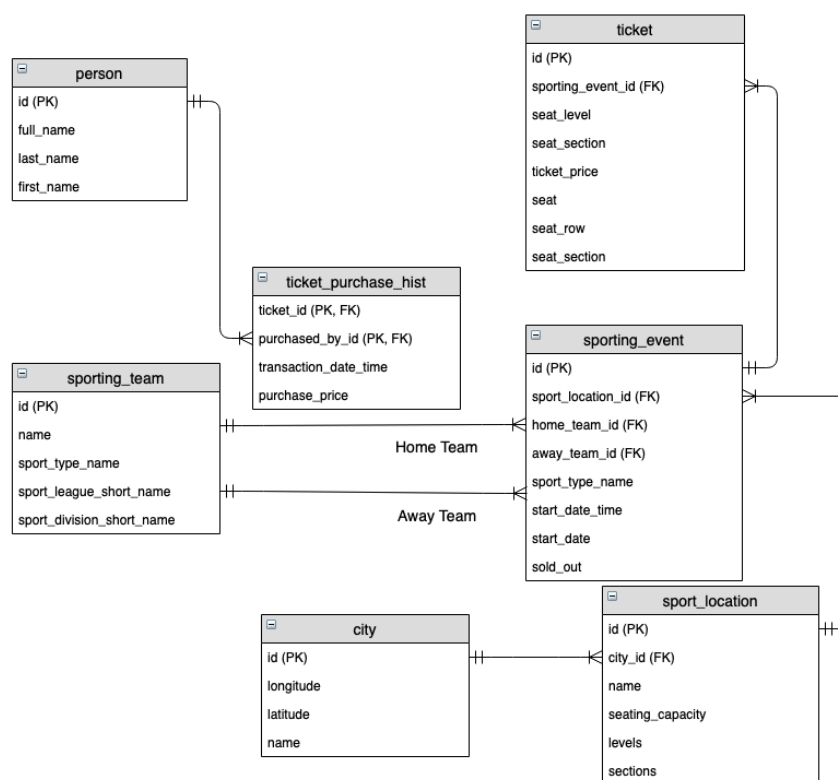
Entre as principais características dos BDRs estão as propriedades ACID — atomicidade, consistência, isolamento e durabilidade — que asseguram a execução correta e segura das transações, mesmo diante de falhas. Além disso, a normalização dos dados é uma prática fundamental nesse modelo, pois evita a redundância e reduz a possibilidade de inconsistências, promovendo uma estrutura mais eficiente. Os BDRs também se destacam por possibilitar consultas relacionais sofisticadas, que envolvem operações como JOINS, o uso de views, stored procedures e triggers, ferramentas que ampliam a capacidade de processamento e automação dentro do banco.

Como ressaltam Elmasri e Navathe (2011), os sistemas de banco de dados relacionais tornaram-se essenciais na construção de aplicações empresariais, devido à sua robustez e capacidade de representar de maneira precisa e estruturada os relacionamentos existentes no mundo real.

#### 2.4.1 Modelagem para sistemas de reserva

A arquitetura do banco de dados do sistema LinkSpace será projetada para espelhar fielmente os processos e as diretrizes operacionais da aplicação. Um modelo relacional, similar ao ilustrado na Figura 5, é uma abordagem comum e eficaz para sistemas com características semelhantes.

Figura 6 - Modelo lógico de dados



Fonte: Amazon AWS (2025)

### 2.4.2 SGBDs recomendados

No panorama atual do desenvolvimento de aplicações web, diversos Sistemas de Gerenciamento de Banco de Dados Relacionais (SGBDRs) destacam-se pela sua robustez e compatibilidade com as demandas de sistemas modernos. A Tabela 3 apresenta uma síntese dos SGBDRs mais prevalentes, juntamente com suas principais vantagens e aplicações comuns.

Tabela 3 - Tipos de SGBDR

SGBDR	Vantagens	Aplicações comuns
<b>PostgreSQL</b>	Suporte avançado a tipos de dados, JSON, GIS, extensões	Projetos acadêmicos e sistemas escaláveis
<b>MySQL/MariaDB</b>	Ampla adoção, excelente performance em leitura	Aplicações web, blogs, CMSs
<b>SQLite</b>	Leve, sem servidor, ideal para protótipos e testes locais	Aplicativos mobile, testes unitários
<b>Oracle</b>	Corporativo, robusto para grandes volumes de dados	Grandes empresas, sistemas legados

Fonte: Date (2004)

Considerando os requisitos específicos do projeto LinkSpace, a adoção do PostgreSQL é fortemente recomendada. Esta escolha se justifica por sua comprovada escalabilidade, aderência estrita ao padrão SQL, funcionalidades avançadas de suporte a operações geoespaciais – um diferencial estratégico para futuras expansões que envolvam a integração de mapas – e sua notável interoperabilidade com frameworks de desenvolvimento web contemporâneos, como Django e Laravel.

### 2.4.3 Boas práticas em projetos com BDRs

No desenvolvimento de aplicações web que se apoiam em bancos de dados relacionais, a implementação de práticas recomendadas é crucial para assegurar a eficiência, a integridade e a robustez do sistema. Dentre essas práticas, destacam-se: a definição estratégica de índices em colunas frequentemente utilizadas em cláusulas WHERE e operações JOIN; a adoção de migrations para o controle de versão da estrutura do banco de dados, como as oferecidas pelo Django ORM ou Laravel Migrations; a aplicação de constraints (NOT NULL, UNIQUE, CHECK, FOREIGN KEY)



para a manutenção da integridade dos dados; a implementação de backups automatizados e mecanismos de replicação visando a alta disponibilidade; e a segregação de ambientes de desenvolvimento, teste e produção, cada um com seu próprio banco de dados isolado.

Como ressalta Date (2019), “A modelagem relacional continua sendo a mais adotada em ambientes de desenvolvimento web devido à sua clareza conceitual, forte base matemática e maturidade tecnológica.”.

#### 2.4.4 Aplicação prática no LinkSpace

O sistema de gerenciamento de banco de dados desempenhará um papel fundamental na sustentação das operações essenciais da aplicação LinkSpace. Suas responsabilidades incluem: a execução eficiente de consultas em tempo real para a verificação da disponibilidade de espaços; o controle automatizado do bloqueio de horários reservados, garantindo a integridade das reservas; a geração de relatórios gerenciais detalhados, fornecendo insights valiosos sobre o uso da plataforma, como a frequência de utilização de salas e os horários de maior demanda; e a gestão robusta do controle de permissões administrativas, assegurando a segurança e a integridade do sistema.

A interação entre a aplicação e o banco de dados será mediada por uma camada de Mapeamento Objeto-Relacional (ORM). Essa abordagem estratégica permitirá abstrair a complexidade inerente à linguagem de consulta estruturada (SQL), promovendo um código mais conciso, manutenível e seguro.

### 2.5 Frameworks e Ferramentas Utilizadas

O desenvolvimento do LinkSpace está fundamentado em uma arquitetura moderna, modular e escalável, que utiliza ferramentas e frameworks amplamente adotados na indústria de software. A seleção criteriosa dessas tecnologias visa garantir alto desempenho, responsividade, reutilização de componentes, boa experiência do usuário (UX), além de facilitar a manutenção e evolução contínua do sistema. Segundo Bass, Clements e Kazman (2013), arquiteturas bem projetadas são

aquelas que conseguem responder às mudanças nos requisitos funcionais e não funcionais sem comprometer a estabilidade da aplicação.

A modularidade é um dos pilares da arquitetura do LinkSpace, permitindo que cada funcionalidade seja desenvolvida, testada e implantada de forma independente, o que reduz o acoplamento entre os componentes e aumenta a manutenibilidade (PRESSMAN; MAXIM, 2016). Além disso, a utilização de frameworks modernos como React (para o front-end) e Node.js (para o back-end) possibilita a criação de sistemas altamente responsivos e com excelente experiência de uso, conforme recomendam Nielsen (2021) e a prática atual do desenvolvimento web.

A adoção de uma arquitetura escalável, com suporte à implementação em nuvem e integração com serviços externos, permite que a plataforma cresça de acordo com a demanda e atenda a múltiplas unidades organizacionais com eficiência. Como destaca Sommerville (2011), sistemas escaláveis devem ser capazes de suportar aumento de carga sem perda significativa de desempenho, o que é essencial para ambientes institucionais dinâmicos como os que o LinkSpace se propõe a atender.

### 2.5.1 React (React.js)

O React é o principal framework utilizado no desenvolvimento da interface do usuário da aplicação. Criado pelo Facebook, ele é amplamente reconhecido por sua performance, flexibilidade e pela grande comunidade ativa que o sustenta. No projeto em questão, optou-se pela utilização da versão mais recente do React, priorizando o uso exclusivo de componentes funcionais, que tornam o código mais conciso e modular. Além disso, adotam-se os React Hooks — como `useState`, `useEffect` e `useContext` — os quais permitem gerenciar o estado e os efeitos colaterais de maneira mais intuitiva e simplificada.

Outro diferencial importante é a utilização do TypeScript para fornecer tipagem estática ao JavaScript, o que contribui para a redução de erros durante o desenvolvimento e melhora significativamente a legibilidade e a manutenibilidade do código. Como destacam Bierman et al. (2020), o uso de tipos estruturais, como os

oferecidos pelo TypeScript, permite que sistemas de larga escala evoluam com segurança e previsibilidade, ao fornecerem garantias estáticas que ajudam na refatoração e no controle da complexidade.

### 2.5.2 Material UI (MUI)

Visando assegurar a consistência visual e otimizar o processo de desenvolvimento da interface de usuário, o projeto LinkSpace utiliza o Material UI (MUI), um sistema de design amplamente adotado no ecossistema React. A incorporação dos seguintes pacotes do MUI é fundamental para a construção da interface, conforme detalhado na Tabela 4:

Tabela 4 - Pacotes de Material UI (MUI)

Pacote	Descrição
@mui/material	Componentes base do Material UI
@mui/icons-material	Conjunto completo de ícones padronizados
@mui/x-date-pickers	Componentes de seleção de data e hora
@mui/x-data-grid	Data grid avançado com paginação, filtros e ordenação automática

Fonte: Elaboração do autor

A Tabela 4 apresenta os pacotes específicos do Material UI integrados ao projeto, juntamente com suas respectivas descrições e funcionalidades. A adoção desses componentes possibilita a criação de uma interface de usuário responsiva, acessível e com suporte nativo aos temas claro e escuro, ambos já implementados no sistema LinkSpace.

### 2.5.3 React Router (react-router-dom)

O gerenciamento de rotas e a navegação entre páginas na aplicação LinkSpace são realizados com o uso da biblioteca react-router-dom, parte integrante do ecossistema React Router. Essa biblioteca é amplamente utilizada no desenvolvimento de aplicações web modernas e fornece uma estrutura para navegação declarativa, eficiente e escalável, com suporte a funcionalidades como rotas aninhadas, rotas protegidas e redirecionamentos (ABRAMOV; CLARK, 2022).

A utilização do React Router permite que o desenvolvedor estabeleça uma separação clara entre as páginas principais da aplicação, como Dashboard, Reservas e Espaços, promovendo maior organização estrutural do front-end. De acordo com Pressman e Maxim (2016), a clareza na estruturação da interface é essencial para garantir a manutenibilidade do sistema e facilitar sua expansão futura.

Além disso, o gerenciamento de rotas com essa abordagem proporciona uma experiência de navegação mais fluida, sem recarregamento completo da página, característica das Single Page Applications (SPAs), melhorando significativamente a responsividade e usabilidade do sistema (NIELSEN, 2021). Essa prática também contribui para a segurança da aplicação, por meio da configuração de rotas privadas que exigem autenticação, reforçando o controle de acesso entre diferentes perfis de usuários.

#### 2.5.4 React Query (TanStack Query)

Para o gerenciamento eficiente do estado de dados assíncronos, o projeto adota o uso da biblioteca `@tanstack/react-query` (anteriormente React Query). Essa ferramenta permite o cache local de dados provenientes de APIs, sincronizações automáticas e revalidação de dados, além de fornecer suporte a mutations com feedback visual otimizado. Ao evitar re-fetching desnecessário, o React Query contribui significativamente para a performance e a escalabilidade da aplicação. Segundo McKinley et al. (2021), bibliotecas modernas de gerenciamento de estado como o React Query são fundamentais para aplicações web que demandam sincronização constante com o backend, por reduzirem a complexidade do código e melhorarem a experiência do usuário.

#### 2.5.5 Date-fns

A manipulação de datas na plataforma LinkSpace é realizada por meio da biblioteca `date-fns`, escolhida por sua leveza, modularidade e integração eficiente com bibliotecas de componentes como o Material UI (MUI). A `date-fns` é amplamente reconhecida no ecossistema JavaScript por oferecer um conjunto robusto de funções

puras para parseamento, formatação, adição, subtração e comparação de datas, com suporte total a tipagem via TypeScript (LOBO, 2023).

Segundo a documentação oficial da biblioteca (DATE-FNS, 2024), um dos grandes diferenciais em relação a outras soluções, como Moment.js, é sua estrutura modular, que permite importar apenas os métodos necessários, reduzindo significativamente o peso final da aplicação. Além disso, a date-fns oferece suporte completo à internacionalização (i18n), o que possibilita a adaptação do sistema para diferentes localidades, idiomas e formatos regionais de data com simplicidade e eficiência.

Esse suporte à localização é particularmente importante em aplicações como o LinkSpace, que podem ser implantadas em diferentes contextos organizacionais e geográficos. De acordo com Sommerville (2011), a capacidade de adaptar uma aplicação às preferências e necessidades culturais e linguísticas dos usuários finais é um dos critérios de qualidade de software e fator decisivo para usabilidade e aceitação da ferramenta.

#### 2.5.6 Ferramentas de Desenvolvimento

O projeto segue padrões modernos de engenharia de software, adotando ferramentas e práticas que promovem a qualidade do código e a consistência do desenvolvimento. Entre os recursos utilizados, destaca-se o TypeScript, uma linguagem que adiciona tipagem estática ao JavaScript e fornece autocompletação e verificação de tipos em tempo de desenvolvimento, reduzindo significativamente a ocorrência de erros. Além disso, ferramentas como o ESLint são empregadas para realizar linting — ou seja, a análise automática de código para identificar padrões problemáticos —, o que contribui para a padronização e a manutenção da base de código.

Complementando esse conjunto, utiliza-se o Prettier, um formatador de código que assegura consistência no estilo de escrita entre os desenvolvedores, sendo integrado ao processo de *commit* e ao ambiente de desenvolvimento. Essas ferramentas são incorporadas tanto ao pipeline de integração contínua e entrega

contínua (CI/CD) quanto ao ambiente local dos desenvolvedores, o que garante uma base de código mais legível, coesa e de fácil manutenção. Como reforça Crook (2022), a adoção de linters e formatadores automáticos como ESLint e Prettier não apenas melhora a qualidade do código, mas também reduz discussões triviais em revisões e acelera o ciclo de desenvolvimento colaborativo.

### 2.5.7 Estrutura do Projeto

A arquitetura da aplicação LinkSpace foi concebida seguindo a filosofia de separação de responsabilidades, buscando garantir escalabilidade, legibilidade e facilidade de manutenção. O sistema organiza-se por meio de componentes reutilizáveis e desacoplados, permitindo que funcionalidades comuns sejam centralizadas e reaproveitadas em diferentes partes da interface. As páginas da aplicação são estruturadas de forma modular, com divisão clara entre seções como Dashboard, Espaços, Reservas, Usuários e Configurações, o que facilita tanto a navegação quanto o desenvolvimento incremental por diferentes membros da equipe.

Além disso, a aplicação faz uso de hooks personalizados para encapsular lógicas específicas e promover a reutilização de comportamentos, mantendo os componentes mais enxutos. Para o gerenciamento de estados compartilhados entre múltiplos componentes, é adotada a Context API do React, uma solução nativa que oferece controle eficiente de estados globais sem a necessidade de bibliotecas externas. Como observa Ousterhout (2018), arquiteturas bem desenhadas tendem a minimizar interdependências entre os módulos, o que reduz o acoplamento e aumenta a clareza do sistema como um todo.

Tabela 5 - Recursos e Funcionalidades Implementadas

Recurso Técnico	Descrição
Sistema de temas personalizável	Alternância entre tema claro e escuro com persistência
Layout responsivo	Compatível com dispositivos móveis e desktops
Formulários com validação	Integração com react-hook-form + validações usando yup
Tabelas com filtros e paginação	Uso do MUI X DataGrid com suporte à paginação e busca
Gráficos e visualizações	Integração com bibliotecas como Recharts ou Chart.js

Fonte: Elaboração do autor

### 2.5.8 UI/UX Avançado

A interface do usuário da aplicação foi projetada com base em princípios de usabilidade e design centrado no usuário, buscando ser ao mesmo tempo intuitiva, moderna e funcional. A navegação lateral foi estruturada com o uso de ícones e submenus, de modo a facilitar o acesso aos recursos principais e promover uma experiência mais organizada e acessível, conforme sugerido por Garrett (2011), que destaca a importância de estruturas visuais claras e hierarquias consistentes na experiência digital.

A aplicação também oferece feedback visual imediato nas ações mais relevantes, como salvar, excluir e realizar reservas. Para isso, são utilizados elementos como loaders, toasts e mensagens dinâmicas, que informam o usuário sobre o andamento ou resultado das operações. Essa prática está alinhada ao princípio de “visibilidade do status do sistema” descrito por Nielsen (2021), essencial para manter o usuário informado e confiante durante a navegação.

Em operações sensíveis que exigem confirmação do usuário, como a exclusão de dados ou confirmação de reserva, são empregados modais e diálogos confirmatórios, que fornecem clareza e evitam a ocorrência de erros por ações involuntárias. Segundo Tidwell (2011), tais padrões de interface reforçam a segurança e previsibilidade das ações, reduzindo frustrações e melhorando a eficiência do sistema.

O sistema também incorpora filtros eficientes por data, usuário e status da reserva, permitindo ao usuário localizar rapidamente as informações desejadas. Essa abordagem de filtragem orientada à tarefa contribui para uma interação mais ágil e produtiva, especialmente em cenários com grande volume de dados (NIELSEN, 2021).

### 2.5.9 Padrões de Projeto e Arquitetura

No que diz respeito à arquitetura do projeto, a aplicação segue princípios modernos de engenharia de software. A componentização é uma abordagem central, permitindo a reutilização e o desacoplamento da lógica visual em componentes

independentes. Para encapsular regras de negócio específicas e promover a reutilização de funcionalidades, são utilizados hooks personalizados. A Context API do React é adotada para o compartilhamento eficiente de estado entre componentes, facilitando a gestão do fluxo de dados. A tipagem forte proporcionada pelo TypeScript ajuda a minimizar erros em tempo de execução, tornando o código mais robusto e fácil de manter. Além disso, a arquitetura é organizada de forma modular, com uma clara divisão por domínios funcionais, como Espaços, Usuários e Reservas, o que favorece tanto o desenvolvimento quanto a escalabilidade do sistema. Como observa Ousterhout (2018), a aplicação de uma arquitetura modular e de componentes desacoplados não só melhora a manutenibilidade do código, mas também acelera a evolução do produto ao permitir que as equipes trabalhem em paralelo de forma mais eficiente.

## **2.6 Experiência do Usuário (UX/UI)**

A experiência do usuário (UX – User Experience) e o design da interface do usuário (UI – User Interface) são componentes centrais no desenvolvimento de sistemas web modernos. Esses elementos não apenas influenciam a percepção do usuário sobre a qualidade da aplicação, mas também impactam diretamente na eficiência de uso, retenção, acessibilidade e confiabilidade da solução (GARRETT, 2011; NIELSEN, 2021).

No caso da plataforma LinkSpace, que foi projetada para facilitar a gestão e reserva de espaços compartilhados, a usabilidade se apresenta como um critério técnico fundamental para garantir que o sistema seja adotado e utilizado com sucesso por diferentes perfis de usuários. Conforme destaca Tidwell (2011), a construção de interfaces intuitivas, responsivas e acessíveis melhora significativamente a interação entre sistema e usuário, reduzindo o tempo de aprendizado e aumentando a satisfação geral com a aplicação.

Além disso, uma experiência bem desenhada é essencial para ambientes institucionais com alta rotatividade de usuários e tarefas frequentes, como reservas recorrentes e gerenciamento de espaços. Assim, o foco em UX/UI não é apenas uma



questão estética, mas estratégica, especialmente em plataformas que lidam com interações recorrentes, dados dinâmicos e acesso multiusuário (NORMAN, 2013).

### 2.6.1 Fundamentos de UX e UI

A experiência do usuário (UX) compreende todo o conjunto de sensações, percepções e respostas cognitivas e emocionais que o usuário desenvolve ao interagir com um sistema digital. Já a interface do usuário (UI) refere-se ao aspecto visual e interativo da aplicação, ou seja, aos elementos com os quais o usuário realiza ações diretas, como botões, formulários, menus e ícones (GARRETT, 2011).

Segundo Nielsen (1994), para que um sistema apresente uma boa experiência de uso, ele deve ser eficiente, permitindo que o usuário conclua suas tarefas rapidamente; eficaz, ao possibilitar que o objetivo da tarefa seja atingido com precisão; intuitivo, sendo de fácil compreensão e uso mesmo para iniciantes; e agradável, proporcionando uma navegação satisfatória, fluida e livre de frustrações. Esses princípios são especialmente relevantes em plataformas com múltiplos perfis de usuários, como é o caso do sistema LinkSpace, que atende administradores, gestores de espaços e usuários finais, cada um com fluxos e necessidades distintas de navegação.

Aplicar tais critérios na concepção da interface não só melhora o desempenho do sistema, mas também influencia diretamente na retenção de usuários, produtividade e confiabilidade da solução (NIELSEN, 1994; GARRETT, 2011).

### 2.6.2 Estratégias de UX adotadas no LinkSpace

Para garantir uma experiência do usuário (UX) eficiente e satisfatória na plataforma LinkSpace, foram adotadas estratégias que seguem as melhores práticas em design centrado no usuário. A primeira delas foi o mapeamento de jornadas do usuário, em que fluxos principais foram desenhados com base nas ações mais frequentes, como reservar um espaço, consultar disponibilidade ou cancelar uma reserva. Essa técnica permite entender o comportamento do usuário e eliminar obstáculos durante a navegação (GARRETT, 2011).

Outro elemento essencial foi o feedback imediato, realizado por meio de snackbars, tooltips, loaders e modais que informam o usuário sobre o status de cada ação executada. Segundo Nielsen (2021), manter o usuário informado de forma clara e contínua aumenta a confiança no sistema e reduz erros por mal-entendidos.

Além disso, buscou-se a minimização de cliques nos principais processos da plataforma, como login, cadastro e realização de reservas. A ideia é reduzir o esforço cognitivo e o tempo necessário para o cumprimento de tarefas, respeitando o princípio de eficiência e fluidez na interação (NIELSEN, 2021).

Por fim, o projeto incorpora práticas de acessibilidade digital, como contraste adequado de cores, tamanhos de fonte legíveis, navegação por teclado e compatibilidade com leitores de tela. Tais critérios seguem as recomendações das Diretrizes de Acessibilidade para Conteúdo Web (WCAG), estabelecidas pela World Wide Web Consortium – W3C (W3C, 2018), que visam garantir o acesso inclusivo a pessoas com deficiência visual, motora ou cognitiva.

### 2.6.3 Design Responsivo

Seguindo os princípios do design responsivo, o sistema LinkSpace foi desenvolvido para se adaptar automaticamente a diferentes tamanhos de tela, como desktop, tablet ou smartphone. Essa abordagem visa garantir usabilidade e acessibilidade em qualquer contexto de uso, seja por um administrador em uma secretaria escolar utilizando um monitor widescreen ou por um aluno acessando a plataforma por meio de um dispositivo móvel. De acordo com Marcotte (2014), o design responsivo é essencial para proporcionar experiências consistentes e otimizadas, independentemente do dispositivo utilizado.

Para atingir essa flexibilidade, foram adotadas técnicas específicas de front-end que se alinham aos padrões modernos. Uma delas é a utilização de grids fluidos, baseados no framework @mui/material, que permite a construção de layouts adaptáveis a partir de colunas dinâmicas (MUI, 2024). Além disso, o sistema integra media queries diretamente à configuração de temas do Material UI, o que possibilita

uma personalização precisa do comportamento visual dos componentes em diferentes larguras de tela (W3C, 2018).

Outro aspecto fundamental do projeto foi a criação de componentes adaptáveis, como tabelas, formulários e cards, que se reorganizam e redimensionam de forma inteligente conforme o espaço disponível. Segundo Nielsen (2021), a responsividade não deve se restringir ao layout, mas abranger também os elementos interativos, garantindo uma navegação fluida, intuitiva e acessível para todos os usuários.

#### 2.6.4 Elementos visuais e de interação

A concepção da interface do LinkSpace foi meticulosamente elaborada seguindo as diretrizes do Material Design, uma abordagem que preconiza a uniformidade visual aliada a padrões de comportamento intuitivos para o usuário. A Tabela 6 detalha a aplicação de elementos específicos do Material Design na construção da interface do sistema:

Tabela 6 - Elementos do Material Design da Interface

Elemento	Aplicação no LinkSpace
Botões e ícones	Ícones contextuais e botões com hierarquia visual clara
Tabelas com filtros	Exibição de reservas com ordenação, filtros por data/usuário
Formulários validados	Feedback em tempo real sobre erros e campos obrigatórios
Modo escuro/claro	Alternância dinâmica com persistência em localStorage
Componentes de navegação	Menu lateral colapsável, breadcrumbs, tabs e navegação por seção

Fonte: Elaboração do autor

Conforme explicitado na Tabela 6, a adoção dos princípios do Material Design manifesta-se em diversos componentes da interface do LinkSpace. A utilização de ícones contextuais e botões com hierarquia visual clara facilita a interação do usuário. Tabelas com funcionalidades de ordenação e filtros (por data e usuário) otimizam a visualização das reservas. Formulários com validação em tempo real e indicação de campos obrigatórios aprimoram a experiência de entrada de dados. A implementação de modos escuro e claro, com persistência das preferências do usuário via localStorage, oferece flexibilidade visual. Por fim, componentes de navegação como

menu lateral colapsável, breadcrumbs, tabs e navegação por seção contribuem para uma exploração fluida e organizada do sistema.

#### 2.6.5 Benefícios práticos

A implementação dos princípios de Experiência do Usuário (UX) e Interface do Usuário (UI) no desenvolvimento do LinkSpace gerou impactos significativos e mensuráveis. Observou-se uma considerável redução no tempo médio necessário para a conclusão de reservas, em comparação com sistemas preexistentes. Adicionalmente, a taxa de erros de utilização, como a realização de reservas em horários indisponíveis, foi significativamente minimizada.

A adoção do sistema por usuários com menor familiaridade técnica, como colaboradores administrativos, também apresentou uma melhora notável. Paralelamente, a confiança dos gestores no sistema foi fortalecida, impulsionada pela visualização de dados em tempo real e pela geração de relatórios intuitivos, facilitando a tomada de decisões estratégicas.

Conforme salientado por Krug (2005), “Interfaces claras economizam o tempo do usuário. O bom design é invisível, ele apenas funciona.”. Essa máxima reflete os resultados alcançados no LinkSpace, onde o foco na usabilidade e na clareza da interface contribuiu diretamente para a eficiência e a satisfação dos usuários.

### 2.7 Segurança da Informação em Aplicações Web

A segurança da informação é um dos pilares mais críticos no desenvolvimento de aplicações web, especialmente em sistemas como o LinkSpace, que gerenciam dados sensíveis de usuários, permissões de acesso e históricos de reservas. O objetivo principal é proteger a confidencialidade, integridade e disponibilidade das informações (princípios da tríade CIA: Confidentiality, Integrity, Availability).

### 2.7.1 Ameaças comuns em aplicações web

As aplicações web modernas enfrentam constantemente ameaças à sua segurança, especialmente diante da crescente complexidade de seus ecossistemas e da variedade de tecnologias utilizadas. O projeto OWASP Top 10 (2021), referência internacional em segurança da informação, lista os principais vetores de ataque que desenvolvedores e arquitetos de software devem considerar em suas soluções. Entre esses riscos, destaca-se o Injection, que compreende a inserção de comandos maliciosos em campos de entrada (como SQL ou NoSQL), o que pode comprometer diretamente a integridade do banco de dados (OWASP, 2021).

Outro risco crítico é o Broken Authentication, relacionado a falhas nos processos de autenticação e gerenciamento de sessões, que podem permitir o acesso indevido a áreas restritas do sistema. Já o Cross-Site Scripting (XSS) permite a execução de scripts maliciosos no navegador de usuários, comprometendo tanto a experiência quanto a privacidade do usuário final (OWASP, 2021).

O Cross-Site Request Forgery (CSRF) também figura entre os principais ataques, explorando a confiança do sistema no navegador do usuário para executar ações sem seu consentimento. A exposição de dados sensíveis (Sensitive Data Exposure) ocorre quando informações como senhas, documentos ou dados bancários são transmitidos ou armazenados sem a devida proteção criptográfica. Por fim, a Security Misconfiguration refere-se a configurações inseguras em servidores, APIs ou aplicações, como senhas padrão, permissões excessivas ou falta de atualizações (OWASP, 2021).

Esses riscos demonstram a importância de se considerar boas práticas de desenvolvimento seguro desde as fases iniciais do projeto, aplicando validações robustas, autenticação forte e configurações seguras para mitigar possíveis vulnerabilidades.

### 2.7.2 Estratégias e práticas adotadas no LinkSpace

O projeto LinkSpace adota um conjunto de boas práticas e mecanismos de proteção em múltiplas camadas:

### 2.7.2.1 Autenticação e autorização

Para garantir a segurança e a integridade do acesso ao sistema LinkSpace, implementou-se um robusto sistema de autenticação e autorização. A autenticação é realizada por meio da utilização de tokens JWT (JSON Web Token), um padrão da indústria que oferece segurança e escalabilidade, com a implementação de controle de expiração para mitigar riscos de acesso indevido e a utilização de refresh tokens para uma experiência de usuário contínua e segura. Adicionalmente, o controle de acesso é gerenciado por meio de um modelo RBAC (Role-Based Access Control), que define níveis distintos de permissão para diferentes categorias de usuários, como usuários comuns, gestores e administradores. Essa abordagem granular permite que as permissões sejam validadas de forma sistemática antes da execução de quaisquer operações sensíveis ou do acesso a dados que requeiram restrições.

### 2.7.2.2 Criptografia de dados

A proteção da confidencialidade e da integridade dos dados é uma prioridade fundamental no projeto LinkSpace. Para tanto, implementou-se uma estratégia de criptografia em diversas camadas do sistema. As senhas dos usuários são armazenadas utilizando a função de hashing bcrypt, um algoritmo robusto que dificulta significativamente a recuperação das credenciais em caso de acesso não autorizado. A comunicação entre o cliente e o servidor é estabelecida exclusivamente através do protocolo HTTPS (Hypertext Transfer Protocol Secure), que emprega as camadas de segurança SSL/TLS (Secure Sockets Layer/Transport Layer Security) para garantir a criptografia e a integridade dos dados transmitidos. Adicionalmente, o desenvolvimento do front-end foi realizado com o cuidado de evitar a exposição de informações sensíveis, como tokens de autenticação ou identificadores internos do sistema, diretamente no lado do cliente.

### 2.7.2.3 Validação de entrada e sanitização

A robustez da segurança do sistema LinkSpace é reforçada por um conjunto de práticas de validação e sanitização de dados, implementadas tanto no front-end quanto no back-end da aplicação. Todas as entradas fornecidas pelos usuários

passam por um rigoroso processo de validação e higienização, utilizando bibliotecas como Yup e Zod no front-end para garantir a integridade dos dados ainda no navegador. No back-end, a utilização de Object-Relational Mappers (ORMs) como Django ORM ou Prisma desempenha um papel crucial na mitigação de vulnerabilidades de SQL Injection, abstraindo a interação direta com o banco de dados e prevenindo a injeção de código malicioso. Adicionalmente, a implementação de uma Content Security Policy (CSP) representa uma medida proativa na redução da superfície de ataque a vulnerabilidades de Cross-Site Scripting (XSS), controlando os recursos que o navegador tem permissão para carregar e executar.

#### 2.7.2.4 Logs e auditoria

A manutenção de registros detalhados de eventos críticos e a implementação de trilhas de auditoria constituem um componente essencial da estratégia de segurança e monitoramento do sistema LinkSpace. Todas as ações consideradas sensíveis, como o processo de autenticação (login), a exclusão de reservas e alterações realizadas por administradores, são registradas de forma sistemática. Cada entrada de log inclui informações cruciais como o timestamp da ocorrência, o endereço IP de origem e a identificação do usuário ou sistema responsável pela ação. Adicionalmente, o armazenamento desses logs é realizado de maneira segura, implementando-se restrições de acesso rigorosas e, quando apropriado, mecanismos de criptografia para proteger a integridade e a confidencialidade dos dados de auditoria.

#### 2.7.2.5 Atualizações e testes automatizados

A manutenção da segurança do sistema LinkSpace é abordada de forma proativa através de um processo contínuo de monitoramento de vulnerabilidades e da implementação de testes automatizados. As dependências do projeto são monitoradas de forma sistemática com o auxílio de ferramentas como npm audit, Snyk ou Dependabot, permitindo a identificação e a correção tempestiva de potenciais vulnerabilidades. Adicionalmente, são realizados testes de segurança automatizados, incluindo testes de penetração e testes de regressão, em endpoints considerados críticos para a funcionalidade e a segurança da aplicação. A aplicação de headers de

segurança HTTP, como Strict-Transport-Security, X-Frame-Options e X-Content-Type-Options, representa uma camada adicional de proteção contra ataques explorando vulnerabilidades comuns em aplicações web.

### 2.7.3 Ferramentas auxiliares

Em consonância com as diretrizes de segurança estabelecidas para o projeto LinkSpace, diversas bibliotecas e utilitários foram incorporados com o objetivo de fortalecer a proteção da aplicação contra vulnerabilidades comuns. A Tabela 7 detalha as tecnologias de segurança adotadas e suas respectivas finalidades:

Tabela 7 - Tecnologias para segurança do projeto

<b>Ferramenta/Biblioteca</b>	<b>Finalidade</b>
Helmet (Node.js/Express)	Configuração de headers de segurança HTTP
CSRF tokens (Django/Express)	Proteção contra ataques CSRF
Yup / Zod	Validação e sanitização de dados no front-end
ESLint Security Rules	Análise estática para detectar falhas em JS/TS

Fonte: elaboração pelo autor

### 2.7.4 Conformidade com normas e boas práticas

O projeto LinkSpace foi desenvolvido em estrita observância aos princípios estabelecidos pela Lei Geral de Proteção de Dados (LGPD). As medidas implementadas para garantir a conformidade incluem o requerimento de consentimento explícito e informado para a coleta de quaisquer dados pessoais, a manutenção da transparência em relação às finalidades de uso e aos períodos de retenção desses dados, e a garantia dos direitos dos usuários à exclusão, retificação e portabilidade de suas informações.

A adesão a esses pilares da LGPD reflete o compromisso do projeto com a privacidade e a segurança dos dados dos usuários. Nesse contexto, a assertiva de Bruce Schneier (2000) torna-se particularmente relevante: “Segurança não é um produto, mas um processo contínuo.”. Essa perspectiva sublinha a natureza dinâmica



da proteção de dados, exigindo monitoramento e aprimoramento constantes das práticas implementadas no LinkSpace.

### **3 TRABALHOS RELACIONADOS**

#### **3.1 Criação de um website sobre tecnologias da informação e comunicação na área da saúde**

A dissertação de mestrado profissional desenvolvida por Maia (2023) teve como objetivo principal a criação de um website informativo voltado à capacitação de trabalhadores da Secretaria de Estado da Saúde de Santa Catarina no uso de Tecnologias da Informação e Comunicação (TICs). A motivação do estudo surgiu da constatação de que, apesar da crescente presença das TICs no cotidiano da saúde, muitos profissionais ainda encontram dificuldades em utilizá-las de forma eficiente, seja pela falta de conhecimento técnico, de infraestrutura ou de conectividade.

A pesquisa adotou o referencial do design thinking e utilizou abordagem descritivo-exploratória, apoiando-se nos modelos Duplo Diamante e ADDIE para estruturação metodológica. A coleta de dados ocorreu entre junho e setembro de 2022, por meio de questionário eletrônico aplicado a 107 profissionais, majoritariamente do sexo feminino, com média de idade de 43,81 anos.

A partir das necessidades identificadas, foi desenvolvido o “TICSaúde website”, uma plataforma responsiva, com acesso por desktop e dispositivos móveis, composta por quatro seções principais: “Página inicial”, “Sobre”, “Tecnologias” e “Mais”. O conteúdo do site abrange informações sobre dez tecnologias específicas de interesse dos participantes, além de materiais complementares. Conclui-se que a solução proposta apresenta potencial para qualificar o uso das TICs no ambiente de trabalho, promovendo a disseminação da informação, o aprimoramento contínuo e a melhoria nas práticas de gestão, educação e atenção em saúde.

#### **3.2 Construindo Espaços de Interação Social a partir de Relações e Práticas de Trabalho Compartilhado**

O estudo de Boucken e Reuschl (2018) investiga como os espaços de coworking atuam como catalisadores de interações sociais por meio das práticas e relações estabelecidas entre seus usuários. Diante da flexibilização crescente das

dinâmicas de trabalho e da ascensão de tecnologias digitais, os autores abordam os coworkings como alternativas inovadoras aos modelos tradicionais de escritórios, ampliando seu papel para além do compartilhamento de infraestrutura.

A pesquisa adota uma abordagem qualitativa com forte inspiração etnográfica, sendo conduzida por meio de um estudo de caso em um espaço de coworking. Os métodos utilizados incluíram observação participante e entrevistas semiestruturadas com 40 associados, ao longo de um período de 25 dias.

Os resultados revelam três camadas de relacionamento predominantes nos espaços analisados: a autogestão, os negócios cooperados e as práticas comunitárias. Essas camadas são mediadas por diversas tecnologias digitais, como plataformas de agendamento, redes sociais internas, aplicativos de gestão de tarefas e sistemas colaborativos. As ferramentas tecnológicas não apenas viabilizam, mas também intensificam as trocas cotidianas entre os usuários, fortalecendo um ecossistema baseado na convivência, cooperação e senso de comunidade. O estudo destaca ainda que o coworking não se resume à logística física, mas constitui um ambiente onde o digital atua como catalisador para práticas de colaboração, configurando-se como um espaço híbrido entre o social e o tecnológico.

### **3.3 Compartilhamento de conhecimento entre empresas e suas consequências**

O artigo de Piccini, Behr e Schiavi (2024) analisa o compartilhamento de conhecimento entre empresas que atuam em espaços de coworking, com ênfase nas implicações dessa prática para a inovação, a competitividade e a sustentabilidade dos negócios. A pesquisa foi conduzida por meio de estudo de casos múltiplos em cinco espaços de coworking localizados em Florianópolis, cidade reconhecida por seu ecossistema voltado à inovação e empreendedorismo.

A metodologia adotada incluiu a realização de 31 entrevistas com gestores, empreendedores e colaboradores, bem como observações em campo e análise documental. A análise dos dados utilizou técnicas de análise de conteúdo, com base em quatro dimensões do compartilhamento interorganizacional: atores envolvidos, tipo de conhecimento, mecanismos utilizados e benefícios percebidos.

Os resultados mostram que os espaços de coworking favorecem a troca constante de conhecimento, principalmente entre freelancers e pequenas equipes, com interações mediadas por plataformas digitais como Instagram, WhatsApp e sistemas próprios. O e-mail, por sua vez, mostrou-se menos efetivo nesse ambiente. Destaca-se o papel dos gestores e da estrutura física dos espaços na facilitação das interações, além da importância de eventos formais para o fortalecimento da cultura colaborativa.

O estudo conclui que os espaços de coworking constituem ecossistemas híbridos — físicos e digitais — que promovem redes de confiança e colaboração, gerando valor coletivo e potencializando processos de inovação.

### **3.4 Desenvolvimento de uma plataforma web para monitoramento das mídias sociais das empresas juniores da Universidade Federal de Ouro Preto**

O trabalho de conclusão de curso desenvolvido por Santos (2019) apresenta a criação de uma plataforma web voltada ao monitoramento das mídias sociais das empresas juniores da Universidade Federal de Ouro Preto, com foco na análise de dados provenientes do Facebook. A proposta busca auxiliar na tomada de decisões estratégicas a partir da visualização clara e acessível de métricas sobre engajamento, público-alvo e crescimento das fan pages institucionais.

A aplicação foi construída utilizando a Graph API do Facebook para extração de dados, que posteriormente passam por tratamento e são organizados em relatórios visuais. A estrutura do sistema se apoia na arquitetura MVC (Model-View-Controller), utilizando tecnologias como Laravel, PHP e banco de dados relacional. A plataforma também apresenta dashboards com gráficos gerados por meio de bibliotecas específicas, possibilitando análises detalhadas sobre curtidas, comentários, impressões e alcance das postagens.

A proposta se destaca por transformar dados brutos, frequentemente pouco compreensíveis para o público leigo, em insights estratégicos úteis para o marketing digital das empresas juniores. O trabalho evidencia a importância de sistemas de

monitoramento como ferramentas de apoio à gestão da presença digital, promovendo a profissionalização da comunicação institucional em ambientes acadêmicos.

### **3.5 O uso do design responsivo no desenvolvimento de websites: aplicação dos métodos na construção de um website**

A monografia de Campos (2017) tem como foco central o estudo e aplicação do design responsivo no desenvolvimento de websites, com ênfase na adaptação dos elementos da interface às diferentes dimensões e resoluções de tela. O trabalho fundamenta-se na premissa de que a multiplicidade de dispositivos — como smartphones, tablets, televisores e computadores — exige soluções que garantam a usabilidade e a experiência do usuário independentemente do meio de acesso.

O autor aborda os principais pilares técnicos do design responsivo, como o uso de grids fluídos, imagens flexíveis e media queries, descrevendo também os critérios para definição de tipografia, paleta de cores e estrutura de navegação. Além da fundamentação teórica, o projeto inclui o desenvolvimento de um website para uma empresa fictícia, intitulado “Sabor Verde”, no qual os conceitos aplicados foram validados por meio de testes em múltiplos dispositivos.

O estudo destaca a importância da responsividade como um critério fundamental para o desenvolvimento web moderno, impactando diretamente na acessibilidade, no desempenho e na retenção do usuário. Campos conclui que o design responsivo não é apenas uma tendência, mas uma necessidade diante da diversidade tecnológica contemporânea.

## 4 PROJETO DE SOFTWARE

### 4.1 Objetivo deste documento

O documento de visão tem como finalidade apresentar uma visão geral do sistema a ser desenvolvido, reunindo os principais requisitos de alto nível e restrições de design que orientam o projeto. Seu objetivo é oferecer ao cliente e às partes interessadas uma compreensão clara e abrangente do produto proposto, sem entrar em detalhes técnicos ou implementações específicas. Assim, ele atua como um guia inicial que alinha expectativas, objetivos e escopo do sistema.

### 4.2 Histórico de Revisão

Tabela 8 – Histórico de Revisão

Histórico de Revisão			
Data	Autor	Descrição	Versão
18/02/2025	Camilo	Criação	1.0
12/03/2025	Renan	Revisão	1.1
26/03/2025	Eduardo	Revisão	1.2
21/04/2025	Gustavo	Revisão	1.3
11/05/2025	Lucas	Revisão	1.4

### 4.3 Escopo do Produto

De acordo com estudos sobre uso de infraestrutura em instituições públicas e privadas, é comum a subutilização de espaços físicos compartilhados, como salas, laboratórios, auditórios e áreas comuns. A falta de controle, de visibilidade sobre a disponibilidade dos ambientes e de sistemas informatizados para gestão contribui para conflitos de uso, reservas informais e desperdício de recursos, além de comprometer a transparência na alocação desses espaços.

Com base nessa problemática, este projeto tem como objetivo a otimização do uso de espaços compartilhados, promovendo uma gestão mais eficiente, organizada e acessível. Para isso, propõe-se o desenvolvimento de uma solução tecnológica denominada LinkSpace, que centraliza o processo de reserva e controle desses

ambientes em uma plataforma web moderna e responsiva.

O LinkSpace é um sistema voltado à gestão de reservas em instituições com múltiplos usuários e ambientes, oferecendo funcionalidades como cadastro de espaços, verificação de disponibilidade em tempo real, filtros por datas e status, reservas com confirmação, cancelamento e histórico detalhado de uso. Além disso, o sistema conta com controle de acesso por perfil (usuário, gestor, administrador) e geração de relatórios gerenciais para acompanhamento da ocupação dos ambientes.

A plataforma também busca promover a integração entre setores, ampliar a transparência nas reservas e reduzir o trabalho manual com planilhas e formulários físicos. Por meio do uso do LinkSpace, espera-se contribuir para a modernização da gestão patrimonial, o uso racional da infraestrutura disponível e a criação de um ambiente institucional mais colaborativo e eficiente.

#### **4.4 Não escopo do produto**

Não é escopo desse projeto:

- Gestão de patrimônio físico e manutenção predial, como controle de inventário, agendamento de reparos ou solicitações de manutenção de equipamentos;
- Integração com sistemas financeiros ou de cobrança, como geração de boletos, faturas ou controle de pagamento por uso do espaço;
- Gerenciamento de acesso físico aos ambientes, como controle por catracas, QR Code ou dispositivos biométricos;

#### **4.5 Descrição dos envolvidos**

Os principais envolvidos neste projeto são as equipes de desenvolvimento, os gestores institucionais e os responsáveis pela administração dos espaços, que colaboram na definição dos requisitos e na validação das funcionalidades, embora nem todos estejam diretamente envolvidos como usuários finais da aplicação.

O público-alvo do sistema inclui os usuários internos das instituições, como servidores, colaboradores, professores e estudantes, que necessitam reservar espaços como salas, auditórios, laboratórios e áreas comuns. Além disso, gestores de espaço também utilizarão o sistema para acompanhar, aprovar ou gerenciar reservas.

O principal objetivo do sistema é oferecer uma plataforma centralizada e acessível, que promova agilidade, organização e transparência no uso da infraestrutura física, otimizando o processo de reserva e contribuindo para a modernização da gestão institucional.

#### 4.6 Resumo dos Envolvidos

Tabela 9 – Resumo do envolvidos

<b>Nome</b>	<b>Descrição</b>	<b>Responsabilidade</b>
Equipe desenvolvimento de da plataforma LinkSpace	Estudantes da Disciplina Ciência da computação	Desenvolvimento de um website para gestão e reserva de espaço
Orientador	Professor na Universidade Paulista, no campus Tatuapé, atual professor da disciplina Trabalho de curso	Orientar as equipes de desenvolvimento e gestão em eventuais dúvidas.



## 4.7 Resumo dos Usuários

Tabela 10 – Resumo dos Usuários

Nome	Responsabilidade	Perfil
Usuário	<ul style="list-style-type: none"> <li>• Visualizar os espaços disponíveis;</li> <li>• Realizar reservas de espaços conforme necessidade;</li> <li>• Cancelar reservas;</li> <li>• Receber confirmação de reservas realizadas;</li> <li>• Acessar o histórico de reservas feitas.</li> </ul>	Utilizador do sistema via web para consultar, reservar e gerenciar o uso de espaços compartilhados, como salas, auditórios ou laboratórios, dentro de uma instituição.
Administrador da Empresa	<ul style="list-style-type: none"> <li>• Cadastrar, editar e remover espaços no sistema;</li> <li>• Cadastrar, editar e remover usuários vinculados à empresa;</li> <li>• Visualizar todas as reservas feitas pelos usuários.</li> </ul>	Usuário com perfil administrativo responsável pela gestão dos recursos do sistema, controle de acessos e manutenção das informações relacionadas aos espaços e usuários da instituição.

## 4.8 Visão geral do produto

Esta seção descreve o que se espera do sistema LinkSpace, proposto para gerenciar de forma centralizada e digitalizada a reserva de espaços compartilhados em instituições públicas ou privadas. Um requisito é definido como “uma condição ou capacidade que o sistema deve satisfazer” e, por isso, deve estar diretamente alinhado aos objetivos do negócio e às necessidades dos usuários.

Os requisitos funcionais representam as ações que o sistema deve ser capaz de executar. No caso do LinkSpace, incluem operações relacionadas ao cadastro e gerenciamento de espaços, reserva, cancelamento, visualização de disponibilidade e controle de usuários. Essas ações são detalhadas nos diagramas de casos de uso, que ilustram a interação entre os atores (usuário e administrador) e as funcionalidades do sistema.

Já os requisitos não funcionais definem atributos de qualidade do sistema, como desempenho, usabilidade, segurança e acessibilidade. Eles descrevem como o sistema deve se comportar, em vez de o que ele deve fazer, e são essenciais para garantir a satisfação dos usuários finais e a eficiência da solução implantada.

Abaixo, esses requisitos serão apresentados de forma estruturada para orientar o desenvolvimento técnico da solução e alinhar as expectativas de todos os envolvidos.

#### 4.9 Requisitos funcionais

Tabela 11 – Requisitos Funcionais

ID	Requisito funcional	Descrição
RF01	Cadastrar Espaços	Permite ao administrador cadastrar um novo espaço.
RF02	Editar Espaços	Permite ao administrador editar as informações.
RF03	Remover Espaços	Permite ao administrador remover um espaço.
RF04	Visualizar Espaços	Permite ao administrador ou usuário visualizar a listagem dos espaços.
RF05	Cadastrar Usuário	Permite ao administrador cadastrar um novo usuário vinculado à sua empresa.

RF06	Editar Usuário	Permite editar dados de um usuário vinculado à empresa.
RF07	Remover Usuário	Permite excluir um usuário do sistema.
RF08	Visualizar Reservas	Permite ao administrador visualizar todas as reservas feitas na plataforma.
RF09	Reservar Espaços	Permite realizar uma reserva de espaço informando data e horário.
RF10	Cancelar Reserva	Permite cancelar uma reserva registrada anteriormente.
RF11	Receber Confirmação de Reserva	Usuário recebe a confirmação de que sua reserva foi registrada com sucesso.
RF12	Visualizar Histórico de Reservas	Permite ao usuário consultar todas as reservas feitas anteriormente.

#### 4.10 Requisitos não funcionais

Tabela 12 – Requisitos não funcionais

ID	Categoria	Requisito Funcional	Não	Descrição
RNF01	Desempenho	Tempo de resposta		O sistema deve responder em até 2 segundos em 95% das requisições
RNF02	Desempenho	Capacidade de processamento	de	Suportar até 200 requisições simultâneas
RNF03	Desempenho	Taxa de transferência de dados	de de	Deve permitir transferências de até 1MB/s entre cliente e servidor
RNF04	Usabilidade	Facilidade de aprendizado	de	Usuário deve ser capaz de utilizar o sistema com noções básicas de web.

RNF05	Usabilidade	Acessibilidade	Interface deve seguir diretrizes WCAG 2.1 nível AA
RNF06	Usabilidade	Intuitividade da interface	Fluxo de navegação não deve ultrapassar 3 cliques por funcionalidade
RNF07	Segurança	Controle de acesso	Acesso baseado em papéis: usuário e administrado
RNF08	Segurança	Proteção contra ataques cibernéticos	Implementar proteção contra SQL Injection, XSS e CSRF
RNF09	Segurança	Criptografia de dados	Dados sensíveis devem ser armazenados com SHA-256 e HTTPS
RNF10	Confiabilidade e Disponibilidade	Tolerância a falhas	Sistema deve exibir mensagens claras em falhas, sem perda de dados
RNF11	Confiabilidade e Disponibilidade	Tempo médio entre falhas (MTBF)	Deve manter uma média mínima de 21 dias sem falhas críticas.
RNF12	Confiabilidade e Disponibilidade	Tempo de recuperação após falha (MTTR)	O sistema deve ser recuperado em até 30 minutos após falha grave.
RNF13	Escalabilidade	Suporte a aumento de usuários	Suportar até 500 usuários ativos simultaneamente
RNF14	Escalabilidade	Crescimento da base de dados	Capacidade de armazenar até 20 mil registros sem degradação de desempenho.
RNF15	Escalabilidade	Expansão de infraestrutura	Sistema deve permitir deploy em servidores de nuvem escaláveis.
RNF16	Manutenibilidade	Facilidade de atualização	Atualizações devem ser feitas com tempo de inatividade inferior a 10 min.
RNF17	Manutenibilidade	Modularidade do código	Código deve seguir arquitetura modular com separação clara de camadas
RNF18	Manutenibilidade	Documentação clara	Código e API devem estar documentados

RNF19	Compatibilidade e Portabilidade	Suporte a diferentes dispositivos navegadores e	Suporte pleno a Chrome, Firefox e Edge.
RNF20	Compatibilidade e Portabilidade	Interoperabilidade com outros sistemas	Permitir exportação de dados em CSV
RNF21	Compatibilidade e Portabilidade	Capacidade de migração entre plataformas	Código deve ser adaptável para servidores Windows ou Linux.

#### 4.11 Restrições

O sistema deverá atender às exigências da Lei Geral de Proteção de Dados (LGPD), restringindo o tipo de informações coletadas e a forma de armazenamento dos dados sensíveis dos usuários, assegurando sua privacidade e segurança.

A aplicação deverá ser plenamente responsiva, adaptando-se de maneira eficiente a diferentes tamanhos de tela, com especial prioridade para dispositivos móveis, como smartphones e tablets.

Será necessário garantir a compatibilidade do sistema com os principais navegadores modernos, incluindo Google Chrome, Mozilla Firefox, Safari e Microsoft Edge, considerando suas duas versões mais recentes.

O desenvolvimento deverá levar em consideração eventuais limitações de desempenho, especialmente em ambientes com conexão móvel de baixa velocidade, como redes 3G ou 4G, otimizando o carregamento e a execução das funcionalidades.

Na primeira versão do sistema, a quantidade de espaços disponíveis para reserva será limitada, sendo prevista a ampliação dessa capacidade em futuras atualizações do projeto.

Por fim, a interface do usuário deverá ser desenvolvida com base em princípios de design responsivo e acessível, respeitando diretrizes de usabilidade e garantindo uma experiência inclusiva para todos os perfis de usuários.

#### **4.12 Política de versionamento**

O controle de versão do projeto será realizado por meio da plataforma GitHub, que permitirá o registro organizado e transparente de todas as alterações feitas ao longo do desenvolvimento do sistema. Para isso, será adotada a política de versionamento semântico (Semantic Versioning), baseada na especificação SemVer 2.0.0, amplamente utilizada em projetos de software.

Cada versão do sistema será identificada por um número composto por três partes: MAJOR, MINOR e PATCH. O identificador MAJOR será incrementado sempre que forem feitas alterações significativas que quebram a compatibilidade com versões anteriores. O número MINOR será alterado quando forem adicionadas novas funcionalidades de forma compatível com o que já existe. Já o PATCH será modificado quando forem realizadas correções de erros ou melhorias internas que não afetam o funcionamento geral do sistema.

Durante o desenvolvimento, serão utilizadas ramificações (branches) no GitHub para organizar as etapas de produção, como desenvolvimento, testes e versão final. Além disso, cada nova versão será identificada com uma tag, que poderá indicar se se trata de uma versão estável, uma versão de testes (beta) ou uma versão candidata a lançamento (release candidate). Essas versões serão acompanhadas por um resumo das alterações (release notes), facilitando a compreensão das mudanças implementadas.

Essa política tem como objetivo manter a organização do projeto, garantir a rastreabilidade de todas as mudanças realizadas, facilitar a comunicação entre os desenvolvedores e usuários, e oferecer maior segurança e previsibilidade nas atualizações do sistema.

#### 4.13 Documento de requisitos (Funcionais e não funcionais)

O documento de requisitos detalha as funcionalidades e qualidades desejadas para o website. Os requisitos funcionais incluem o cadastro e *login* de usuários, a busca e filtragem de espaços por tipo, localização, data, horário e capacidade, a visualização da disponibilidade dos espaços em um calendário intuitivo, a reserva de espaços com o fornecimento de informações como data, horário e finalidade, a geração de confirmações de reserva com detalhes, o gerenciamento de reservas pelos usuários, o gerenciamento de espaços pelos administradores (incluindo cadastro, edição e definição de disponibilidade), o gerenciamento de usuários pelos administradores, a geração de relatórios de reservas e, opcionalmente, a integração com um *gateway* de pagamento para processar pagamentos *online*. Os requisitos não funcionais priorizam o desempenho do sistema, com tempo de resposta máximo de 3 segundos, a usabilidade da interface, a responsividade do *website* em diferentes dispositivos, a segurança dos dados e transações, a escalabilidade da arquitetura, a manutenibilidade do código e a acessibilidade do *website* para pessoas com deficiência.

Tabela 13 – Requisitos Funcionais do Sistema

ID	Requisito funcional	Descrição
RF01	Cadastrar Espaços	Permite ao administrador cadastrar um novo espaço disponível para reserva na plataforma, informando dados como nome, descrição do espaço, localização do espaço, capacidade, horários disponíveis.
RF02	Editar Espaços	Permite ao administrador editar as informações de um espaço já cadastrado na plataforma, como nome, descrição do espaço, localização do espaço, capacidade, horários disponíveis.
RF03	Remover Espaços	Permite ao administrador remover um espaço previamente cadastrado na plataforma, tornando-o indisponível para reservas futuras.
RF04	Visualizar Espaços	Permite ao administrador ou usuário visualizar a listagem dos espaços disponíveis para reserva, incluindo informações como nome, descrição, localização, capacidade e horários disponíveis.

RF05	Cadastrar Usuário	Permite ao administrador cadastrar um novo usuário vinculado à sua empresa, fornecendo dados como nome, e-mail e senha.
RF06	Editar Usuário	Permite atualizar as informações de um usuário da empresa, como nome, e-mail e cargo, garantindo que os dados estejam corretos e atualizados.
RF07	Remover Usuário	Permite remover um usuário vinculado à empresa, excluindo seu acesso ao sistema e suas permissões, quando não for mais necessário manter o vínculo.
RF08	Visualizar Reservas	Permite visualizar todas as reservas registradas, exibindo detalhes como data, horário, espaço reservado, status e usuário responsável.
RF09	Reservar Espaços	Permite reservar um espaço disponível, selecionando data, horário e finalidade da reserva, com confirmação registrada no sistema.
RF10	Cancelar Reserva	Permite cancelar uma reserva registrada, liberando o espaço para novos agendamentos e atualizando o status no sistema.
RF11	Receber Confirmação de Reserva	Permite ao usuário receber uma confirmação após efetuar uma reserva, com detalhes como data, horário, espaço reservado e status, enviada por e-mail ou exibida no sistema.
RF12	Visualizar Histórico de Reservas	Permite ao usuário acessar o histórico de reservas realizadas, com informações como datas, horários, espaços utilizados e status de cada reserva.

Tabela 14 – Requisitos não funcionais do sistema

ID	Categoria	Requisito Não Funcional	Descrição
RNF01	Desempenho	Tempo de resposta	O sistema deve responder em até 2 segundos em 95% das requisições
RNF02	Desempenho	Capacidade de processamento	Suportar até 200 requisições simultâneas



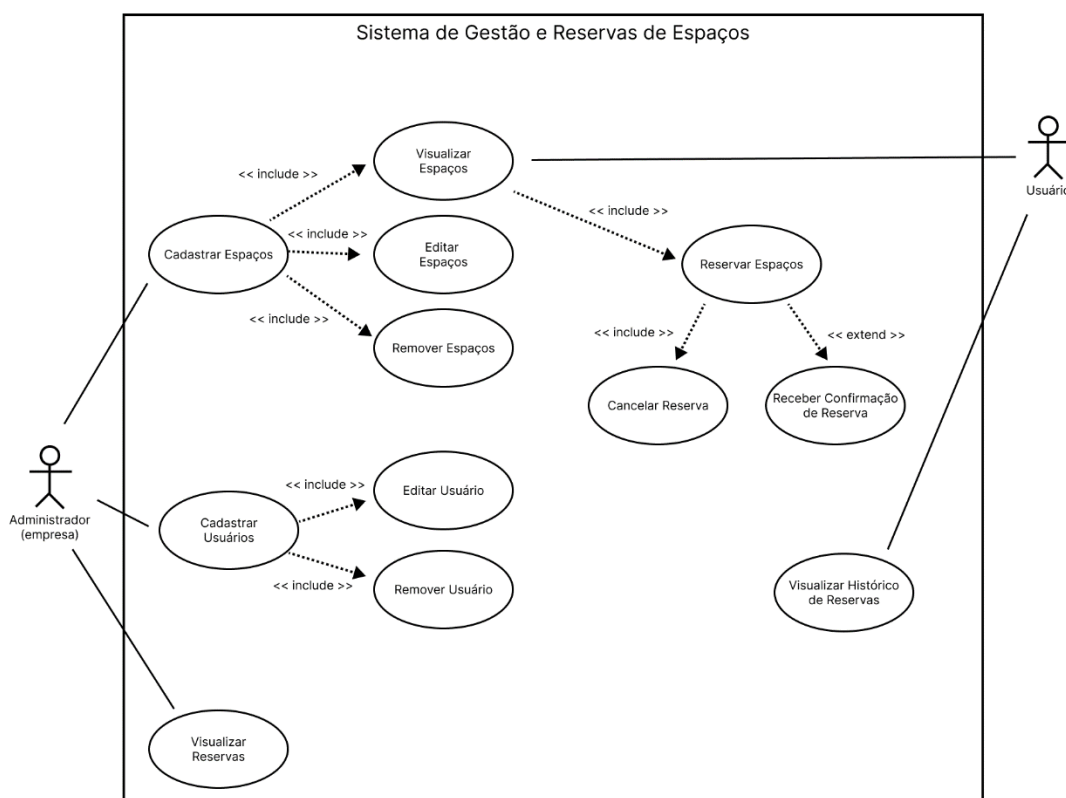
RNF03	Desempenho	Taxa de transferência de dados	Deve permitir transferências de até 1MB/s entre cliente e servidor
RNF04	Usabilidade	Facilidade de aprendizado	Usuário deve ser capaz de utilizar o sistema com noções básicas de web.
RNF05	Usabilidade	Acessibilidade	Interface deve seguir diretrizes WCAG 2.1 nível AA
RNF06	Usabilidade	Intuitividade da interface	Fluxo de navegação não deve ultrapassar 3 cliques por funcionalidade
RNF07	Segurança	Controle de acesso	Acesso baseado em papéis: usuário e administrador
RNF08	Segurança	Proteção contra ataques cibernéticos	Implementar proteção contra SQL Injection, XSS e CSRF
RNF09	Segurança	Criptografia de dados	Dados sensíveis devem ser armazenados com SHA-256 e HTTPS
RNF10	Confiabilidade e Disponibilidade	Tolerância a falhas	Sistema deve exibir mensagens claras em falhas, sem perda de dados
RNF11	Confiabilidade e Disponibilidade	Tempo médio entre falhas (MTBF)	Deve manter uma média mínima de 21 dias sem falhas críticas.
RNF12	Confiabilidade e Disponibilidade	Tempo de recuperação após falha (MTTR)	O sistema deve ser recuperado em até 30 minutos após falha grave.
RNF13	Escalabilidade	Suporte a aumento de usuários	Suportar até 500 usuários ativos simultaneamente

RNF14	Escalabilidade	Crescimento da base de dados	Capacidade de armazenar até 20 mil registros sem degradação de desempenho.
RNF15	Escalabilidade	Expansão de infraestrutura	Sistema deve permitir deploy em servidores de nuvem escaláveis.
RNF16	Manutenibilidade	Facilidade de atualização	Atualizações devem ser feitas com tempo de inatividade inferior a 10 min.
RNF17	Manutenibilidade	Modularidade do código	Código deve seguir arquitetura modular com separação clara de camadas
RNF18	Manutenibilidade	Documentação clara	Código e API devem estar documentados
RNF19	Compatibilidade e Portabilidade	Suporte a diferentes dispositivos e navegadores	Suporte pleno a Chrome, Firefox e Edge.
RNF20	Compatibilidade e Portabilidade	Interoperabilidade com outros sistemas	Permitir exportação de dados em CSV
RNF21	Compatibilidade e Portabilidade	Capacidade de migração entre plataformas	Código deve ser adaptável para servidores Windows ou Linux.

#### 4.14 Diagramas de casos de uso

Os diagramas de casos de uso representam as interações entre os atores (usuário comum e administrador) e o sistema, descrevendo as funcionalidades do ponto de vista do usuário. Por exemplo, os casos de uso para o usuário comum incluem cadastrar-se, fazer *login*, buscar espaços, visualizar disponibilidade, reservar espaço, visualizar, alterar e cancelar reservas, e fazer *logout*, enquanto os casos de uso para o administrador incluem fazer *login*, gerenciar espaços e usuários, gerar relatórios de reservas e fazer *logout*.

Figura 7 – Diagrama de uso



##### 4.14.1 Documentação dos casos de uso

A documentação dos casos de uso é uma etapa essencial no processo de desenvolvimento de sistemas orientados a objetos, pois descreve de maneira clara e estruturada as interações entre os usuários (atores) e o sistema. Essa documentação permite compreender como o sistema deve se comportar diante das diferentes ações executadas pelos usuários, servindo como base para o levantamento de requisitos,

desenvolvimento, testes e validação da solução.

Cada caso de uso representa uma funcionalidade específica do sistema, detalhando o fluxo de eventos, as condições de entrada e saída, os atores envolvidos, e possíveis exceções. Essa abordagem contribui para a organização do projeto, facilita a comunicação entre os membros da equipe de desenvolvimento e garante maior assertividade na entrega de um sistema funcional e alinhado às necessidades do usuário final.

No presente trabalho, os casos de uso foram elaborados com base nas funcionalidades identificadas para o sistema proposto, considerando tanto os requisitos técnicos quanto as necessidades práticas dos usuários. A seguir, apresenta-se a descrição detalhada de cada um dos casos de uso definidos, do UC01 ao UC12.

## 4.14.2 UC01 – Cadastrar Espaço

Figura 8 – UC01

Nome do Caso de Uso	
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	
<b>Atores Secundários</b>	
<b>Resumo</b>	
<b>Pré-condições</b>	
<b>Pós-condições</b>	
<b>Cenário Principal</b>	
Ações do Ator	Ações do Sistema
1. Acessa a opção "Cadastrar Espaço"	1. Exibe formulário com os campos obrigatórios
2. Preenche os dados do novo espaço	
3. Confirma o cadastro	2. Valida os dados inseridos
	3. Salva os dados no banco de dados
	4. Exibe mensagem de sucesso
<b>Cenário Alternativo</b>	
Ações do ator	Ações do Sistema
2a. Deixa campos obrigatórios em branco	2a. Exibe mensagem informando os campos que devem ser preenchidos
2b. Informa dados inválidos (ex: horário mal formatado)	2b. Exibe mensagem de erro com detalhes sobre os dados incorretos
3a. Corrige os dados e envia novamente	3a. Valida novamente e prossegue com o fluxo principal
3b. Desiste do cadastro	3b. Cancela o processo e retorna à listagem de espaços
<b>Cenário de Exceção</b>	
Ações do ator	Ações do Sistema
3x. Envia o formulário e há erro de conexão	3x. Exibe mensagem de erro e orienta o administrador a tentar novamente mais tarde

O caso de uso UC01 – Cadastrar Espaço tem como objetivo permitir que o administrador da empresa registre um novo espaço na plataforma, tornando-o disponível para visualização e reserva pelos usuários autorizados. Esta funcionalidade é essencial para viabilizar a gestão dos ambientes compartilhados, possibilitando a organização, o controle e o agendamento de espaços de forma centralizada.

O ator principal envolvido neste caso de uso é o Administrador, que representa a pessoa responsável por gerenciar os espaços disponibilizados no sistema. O ator secundário é o Website, que representa a interface do sistema por meio da qual as interações são realizadas.

Para que o administrador possa realizar o cadastro de um novo espaço, é necessário que esteja autenticado no sistema, garantindo assim a segurança e rastreabilidade das ações executadas. Ao acessar a funcionalidade “Cadastrar Espaço”, o sistema exibe um formulário com todos os campos obrigatórios necessários para o registro do novo ambiente.

O fluxo principal se desenvolve conforme o administrador preenche os dados do novo espaço e confirma o envio. O sistema, então, valida as informações inseridas, realiza o salvamento dos dados no banco de dados e, em seguida, exibe uma mensagem de sucesso, confirmando a operação. A partir desse momento, o espaço torna-se disponível para os usuários da empresa, que poderão visualizá-lo e realizar reservas conforme suas necessidades.

Este caso de uso também contempla cenários alternativos, como o preenchimento incompleto do formulário ou o uso de dados inválidos (por exemplo, um horário mal formatado). Nestes casos, o sistema informa ao administrador os erros identificados, permitindo que os dados sejam corrigidos e reenviados. O administrador também pode optar por desistir do cadastro, situação na qual o sistema cancela o processo e retorna à listagem de espaços disponíveis.

Em situações excepcionais, como uma falha de conexão no momento do envio do formulário, o sistema exibe uma mensagem de erro orientando o administrador a tentar novamente mais tarde, de forma a minimizar frustrações e garantir a continuidade do uso da plataforma.

## 4.14.3 UC02 – Editar Espaço

Figura 9 – UC02

Nome do Caso de Uso	
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	
<b>Atores Secundários</b>	
<b>Resumo</b>	
<b>Pré-condições</b>	
<b>Pós-condições</b>	
<b>Cenário Principal</b>	
Ações do Ator	Ações do Sistema
1. Acessa a listagem de espaços	1. Exibe os espaços disponíveis para edição
2. Seleciona o espaço a ser editado	2. Exibe formulário preenchido com os dados atuais
3. Altera as informações desejadas	
4. Confirma a edição	3. Valida os dados modificados
	4. Atualiza os dados no banco de dados
	5. Exibe mensagem de sucesso
<b>Cenário Alternativo</b>	
Ações do ator	Ações do Sistema
4a. Deixa campos obrigatórios em branco	3a. Exibe mensagem de erro indicando os campos obrigatórios
4a. Preenche dados inválidos	3b. Exibe mensagem de erro com validação de formato e regras
4b. Corrige os dados e reenviar	3c. Realiza nova validação dos dados
<b>Cenário de Exceção</b>	
Ações do ator	Ações do Sistema
4x. Tenta salvar mas ocorre falha na atualização	4x. Exibe mensagem de erro técnico e orienta tentar novamente mais tarde

O caso de uso UC02 – Editar Espaço está inserido no processo de gerenciamento de espaços e tem como objetivo permitir que o administrador atualize as informações de um espaço previamente cadastrado. Esta funcionalidade é essencial para garantir a manutenção e a consistência dos dados disponíveis na plataforma, assegurando que os usuários tenham acesso a informações corretas e atualizadas sobre os espaços disponíveis para reserva.

O ator principal deste caso de uso é o Administrador, responsável por gerenciar os espaços vinculados à empresa. O Website atua como ator secundário, fornecendo a interface por meio da qual ocorrem as interações com o sistema.

Para a execução do caso de uso, duas pré-condições devem ser satisfeitas: o administrador deve estar autenticado no sistema, e o espaço a ser editado deve existir. Uma vez atendidas essas condições, o administrador acessa a listagem de espaços disponíveis e seleciona aquele que deseja editar. O sistema, por sua vez, exibe um formulário já preenchido com os dados atuais do espaço selecionado.

Após realizar as alterações desejadas, o administrador confirma a edição. O sistema valida os novos dados fornecidos, atualiza as informações no banco de dados e exibe uma mensagem de sucesso, indicando que a operação foi concluída com êxito. Como pós-condição, as informações atualizadas ficam imediatamente disponíveis para visualização pelos usuários da empresa.

Este caso de uso também contempla cenários alternativos. Se o administrador deixar campos obrigatórios em branco ou inserir dados inválidos (como um horário mal formatado ou fora das regras estabelecidas), o sistema exibirá mensagens de erro específicas, indicando os problemas encontrados. Após corrigir os dados, o administrador poderá reenviar o formulário, e o sistema realizará uma nova validação, prosseguindo com a atualização caso os dados estejam corretos.

Em cenários de exceção, como falhas técnicas durante o processo de salvamento dos dados, o sistema deverá informar ao administrador que ocorreu um erro e orientá-lo a tentar novamente mais tarde, garantindo uma comunicação transparente e evitando a perda de informações.



## 4.14.4 UC03 – Remover Espaço

Figura 10 – UC03

Nome do Caso de Uso	
Caso de Uso Geral	<b>UC03 - Remover Espaço</b>
Ator Principal	Gerenciamento de Espaço
Atores Secundários	Administrador
Resumo	Website
Pré-condições	Permite ao administrador remover um espaço.
Pós-condições	O administrador deve estar autenticado no sistema e o espaço deve estar cadastrado.
<b>Cenário Principal</b>	
Ações do Ator	Ações do Sistema
1. Acessa a listagem de espaços	1. Exibe os espaços disponíveis
2. Seleciona o espaço a ser removido	2. Exibe detalhes e opção de confirmação de remoção
3. Confirma a remoção	3. Remove o espaço do banco de dados
	4. Exibe mensagem de sucesso
<b>Cenário Alternativo</b>	
Ações do ator	Ações do Sistema
3a. Desiste da remoção após visualização dos dados	3a. Cancela o processo e retorna à listagem de espaços
<b>Cenário de Exceção</b>	
Ações do ator	Ações do Sistema
3x. Tenta remover espaço mas há reservas futuras vinculadas	3x. Exibe mensagem de erro informando que a exclusão não é permitida enquanto houver reservas pendentes

O caso de uso UC03 – Remover Espaço tem como finalidade permitir que o administrador exclua um espaço previamente cadastrado na plataforma. Essa funcionalidade é necessária para assegurar que apenas ambientes válidos, atualizados e disponíveis para uso permaneçam acessíveis aos usuários. A remoção de espaços inativos ou descontinuados contribui para a integridade da base de dados e para uma melhor experiência do usuário.

Neste caso, o Administrador é o ator principal, sendo o responsável por gerenciar os espaços da empresa. O Website, por sua vez, atua como ator secundário, representando a interface do sistema utilizada para executar a operação.

As pré-condições para realizar a remoção incluem a autenticação do administrador no sistema e a existência do espaço na base de dados. Atendidas essas condições, o administrador acessa a listagem de espaços disponíveis, seleciona aquele que deseja remover e visualiza os detalhes do espaço, juntamente com a opção de confirmação da exclusão. Após confirmar a ação, o sistema realiza a remoção do espaço no banco de dados e apresenta uma mensagem de sucesso, indicando a conclusão da operação. Como pós-condição, o espaço removido deixa de estar disponível para visualização ou reserva pelos usuários.

O caso de uso prevê ainda um cenário alternativo, no qual o administrador, após visualizar os dados do espaço, decide não prosseguir com a exclusão. Neste caso, o sistema cancela a operação e retorna à listagem geral de espaços sem realizar alterações.

Há também a consideração de um cenário de exceção, quando o administrador tenta remover um espaço que possui reservas futuras vinculadas. Nessa situação, o sistema bloqueia a exclusão e apresenta uma mensagem informando que a operação não pode ser concluída enquanto houver pendências associadas ao espaço, evitando assim inconsistências e conflitos de uso.

## 4.14.5 UC04 – Visualizar Espaços

Figura 11 – UC04

Nome do Caso de Uso	
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	
<b>Atores Secundários</b>	
<b>Resumo</b>	
<b>Pré-condições</b>	
<b>Pós-condições</b>	
<b>Cenário Principal</b>	
Ações do Ator	Ações do Sistema
1. Acessa a funcionalidade "Visualizar Espaços"	1. Consulta os espaços disponíveis no banco de dados
	2. Exibe a listagem de espaços
2. Clica em um espaço da listagem	3. Exibe os detalhes do espaço selecionado
Cenário Alternativo	
Ações do ator	Ações do Sistema
1a. Aplica filtros de busca (ex: capacidade, local, horário)	1a. Filtra os resultados de acordo com os critérios aplicados
1b. Nenhum espaço atende aos critérios	1b. Exibe mensagem "Nenhum espaço encontrado com os filtros aplicados"
Cenário de Exceção	
Ações do ator	Ações do Sistema
1x. Acessa a funcionalidade, mas ocorre erro na conexão com o banco de dados	1x. Exibe mensagem de erro técnico e orienta o usuário a tentar novamente

O caso de uso UC04 – Visualizar Espaços descreve a funcionalidade que permite tanto ao administrador quanto ao usuário comum visualizar a listagem de espaços disponíveis para consulta e reserva. Esta ação é fundamental para que os usuários tenham acesso às informações atualizadas sobre os ambientes cadastrados no sistema, favorecendo a organização e a tomada de decisão quanto ao uso dos espaços compartilhados.

Neste cenário, o Usuário (ator principal), que pode ser um administrador ou um usuário comum, interage com o sistema por meio do Website (ator secundário). Para utilizar essa funcionalidade, é necessário que o usuário esteja devidamente autenticado no sistema, o que constitui uma pré-condição para garantir segurança e

rastreabilidade.

No fluxo principal, o usuário acessa a funcionalidade "Visualizar Espaços". O sistema então realiza uma consulta no banco de dados e exibe a listagem de espaços disponíveis. Ao selecionar um espaço específico da lista, o sistema apresenta os detalhes completos daquele ambiente, como localização, capacidade e horários de funcionamento, conforme configurado pelo administrador.

O caso de uso também contempla cenários alternativos, como a aplicação de filtros de busca (por exemplo, capacidade, local ou horário). Nessa situação, o sistema filtra os resultados de acordo com os critérios informados. Caso nenhum espaço atenda aos critérios aplicados, o sistema informa ao usuário por meio de uma mensagem clara: "Nenhum espaço encontrado com os filtros aplicados", permitindo que ele refine sua busca.

Em casos de exceção, como falhas técnicas na conexão com o banco de dados no momento do acesso à funcionalidade, o sistema exibe uma mensagem de erro técnico e orienta o usuário a tentar novamente mais tarde. Essa abordagem busca preservar a experiência do usuário, mesmo diante de problemas temporários na infraestrutura.

Ao final da operação, como pós-condição, o usuário estará apto a consultar os espaços disponíveis e visualizar suas características detalhadas, podendo posteriormente seguir com o processo de reserva, se desejar.

## 4.14.6 UC05 – Cadastrar Usuário

Figura 12 – UC05

Nome do Caso de Uso	
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	
<b>Atores Secundários</b>	
<b>Resumo</b>	
<b>Pré-condições</b>	
<b>Pós-condições</b>	
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessa a funcionalidade "Cadastrar Usuário"	1. Exibe formulário de cadastro com os campos obrigatórios
2. Preenche os dados do novo usuário	
3. Confirma o cadastro	2. Valida os dados inseridos
	3. Salva os dados no banco de dados
	4. Exibe mensagem de sucesso
Cenário Alternativo	
Ações do ator	Ações do Sistema
3a. Deixa campos obrigatórios em branco	2a. Exibe mensagem de erro indicando os campos obrigatórios faltantes
3b. Preenche e-mail em formato inválido	2b. Exibe mensagem de erro indicado o formato incorreto
3b. Informa o e-mail já existente	2c. Exibe mensagem de erro informando que o e-mail já está cadastrado
3d. Preenche senha muito curta ou fraca	2d. Informa que a senha não atende aos requisitos mínimos de segurança
3e. Corrige os dados e reenviar	2e. Realiza nova validação e segue o fluxo principal a partir da etapa 3
Cenário de Exceção	
Ações do ator	Ações do Sistema
3x. Envia o formulário e há erro de conexão	3x. Exibe mensagem de erro e orienta o administrador a tentar novamente mais tarde

O caso de uso UC05 – Cadastrar Usuário tem como finalidade permitir que o administrador registre novos usuários vinculados à sua empresa no sistema. Esta funcionalidade é fundamental para garantir que apenas pessoas autorizadas tenham acesso à plataforma, de acordo com os níveis de permissão definidos, e possam usufruir das funcionalidades, como visualização e reserva de espaços.

Neste caso, o Administrador atua como ator principal, sendo responsável por cadastrar os usuários da organização. O Website é o ator secundário, responsável por prover a interface de interação com o sistema.

A pré-condição para realizar o cadastro é que o administrador esteja

devidamente autenticado no sistema. Uma vez autenticado, o administrador acessa a funcionalidade "Cadastrar Usuário", e o sistema exibe um formulário contendo os campos obrigatórios para o cadastro.

No fluxo principal, o administrador preenche os dados do novo usuário e confirma a ação. O sistema valida os dados inseridos, salva as informações no banco de dados e exibe uma mensagem de sucesso, indicando que o novo usuário foi cadastrado com êxito. Como pós-condição, o usuário recém-cadastrado poderá acessar o sistema com as permissões correspondentes.

O cenário alternativo contempla diversas situações de erro durante o preenchimento dos dados. Se o administrador deixar campos obrigatórios em branco, o sistema exibe uma mensagem informando os campos ausentes. Caso o e-mail informado esteja em formato inválido ou já esteja cadastrado, o sistema também notifica o erro de forma adequada. A mesma lógica se aplica ao preenchimento de senhas que não atendem aos requisitos mínimos de segurança (por exemplo, comprimento mínimo ou complexidade). Após receber a notificação de erro, o administrador pode corrigir os dados e reenviar o formulário. O sistema então realiza uma nova validação e retoma o fluxo principal a partir da etapa de confirmação do cadastro.

Em um cenário de exceção, pode haver falhas de conexão no momento do envio do formulário. Nessa situação, o sistema deve apresentar uma mensagem de erro técnico e orientar o administrador a tentar novamente mais tarde, garantindo a continuidade da operação assim que o problema for resolvido.

## 4.14.7 UC06 – Editar Espaço

Figura 13 – UC06

Nome do Caso de Uso	
<b>UC06 - Editar Usuário</b>	
Caso de Uso Geral	
Ator Principal	
Atores Secundários	
Resumo	
Pré-condições	
Pós-condições	
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessa a listagem de usuários	1. Exibe os usuários cadastrados
2. Seleciona o usuário para edição	2. Exibe formulário com os dados atuais preenchidos
3. Altera as informações	
4. Confirma a edição	3. Valida os dados
	4. Atualiza no banco de dados
	5. Exibe mensagem de sucesso
Cenário Alternativo	
Ações do ator	Ações do Sistema
4a. Deixa campos obrigatórios em branco	3a. Exibe erro informando os campos obrigatórios
4b. Preenche e-mail em formato inválido	3b. Informa erro de formatação
4c. Informa e-mail já utilizado por outro usuário	3c. Exibe erro de e-mail duplicado
4d. Corrige os dados e reenviar	3d. Valida novamente e segue o fluxo normal
Cenário de Exceção	
Ações do ator	Ações do Sistema
4x. Tenta salvar, mas ocorre erro no sistema	4x. Exibe mensagem de falha técnica e orienta tentar novamente

O caso de uso UC06 – Editar Usuário tem como objetivo permitir que o administrador atualize os dados de um usuário vinculado à sua empresa. Essa funcionalidade é importante para manter as informações cadastrais dos usuários atualizadas, refletindo alterações como troca de e-mail, redefinição de senha ou ajustes em dados pessoais.

Neste cenário, o Administrador é o ator principal responsável por realizar as alterações. O Website, por sua vez, atua como ator secundário, fornecendo os meios de interação entre o administrador e o sistema.

A pré-condição para executar essa funcionalidade é que o administrador esteja autenticado na plataforma. A partir disso, o fluxo principal se inicia com o acesso à listagem de usuários. O sistema exibe todos os usuários cadastrados, permitindo que o administrador selecione aquele cujas informações deseja editar. Em seguida, o sistema apresenta um formulário preenchido com os dados atuais do usuário. O administrador pode então realizar as alterações necessárias e confirmar a edição. O sistema valida os dados modificados, atualiza as informações no banco de dados e exibe uma mensagem de sucesso, encerrando o processo. Como pós-condição, os dados do usuário são atualizados no sistema.

O cenário alternativo contempla possíveis erros no preenchimento do formulário. Caso o administrador deixe campos obrigatórios em branco, informe um e-mail em formato inválido ou forneça um e-mail que já esteja cadastrado para outro usuário, o sistema exibirá mensagens de erro específicas. Após corrigir os dados, o sistema realiza nova validação e retoma o fluxo a partir da etapa de confirmação.

Em um cenário de exceção, o sistema pode enfrentar uma falha técnica no momento da atualização das informações. Nessa situação, uma mensagem de erro é apresentada ao administrador, juntamente com a orientação para que a operação seja tentada novamente mais tarde.



## 4.14.8 UC07 – Remover Usuário

Figura 14 – UC07

Nome do Caso de Uso		UC07 - Remover Usuário	
Caso de Uso Geral		Cadastrar Usuários	
Ator Principal		Administrador	
Atores Secundários		Website	
Resumo		Permite excluir um usuário do sistema.	
Pré-condições		O administrador deve estar autenticado e o usuário a ser removido deve existir.	
Pós-condições		O usuário removido não poderá mais acessar o sistema.	
Cenário Principal			
Ações do Ator		Ações do Sistema	
1. Acessa a listagem de usuários		1. Exibe todos os usuários vinculados à empresa	
2. Seleciona o usuário a ser removido		2. Exibe os dados e a opção de confirmação	
3. Confirma a remoção		3. Exclui o usuário do banco de dados	
		4. Exibe mensagem de sucesso	
Cenário Alternativo			
Ações do ator		Ações do Sistema	
2a. Seleciona usuário com permissões maiores ou iguais		2a. Exibe mensagem de restrição (não é permitido remover este usuário)	
3a. Desiste da remoção		3a. Cancela a operação e retorna para a listagem	
Cenário de Exceção			
Ações do ator		Ações do Sistema	
3x. Tenta remover o usuário mas ocorre falha na operação		3x. Exibe mensagem de erro técnico e orienta tentar novamente mais tarde	

O caso de uso UC07 – Remover Usuário tem como finalidade permitir que o administrador exclua um usuário vinculado à empresa, impossibilitando o seu acesso futuro ao sistema. Essa funcionalidade é essencial para manter o controle de acesso e garantir que apenas usuários ativos e autorizados estejam presentes na base de dados do sistema.

O Administrador é o ator principal responsável por realizar a remoção, enquanto o Website atua como ator secundário, fornecendo a interface de interação com o sistema.

Para que o processo seja iniciado, é necessário que o administrador esteja

autenticado no sistema e que o usuário a ser removido esteja previamente cadastrado, conforme as pré-condições estabelecidas. O fluxo principal inicia-se com o acesso à listagem de usuários vinculados à empresa. O sistema apresenta todos os usuários cadastrados, permitindo ao administrador selecionar aquele que será removido. Em seguida, são exibidos os dados do usuário e a opção de confirmação. Ao confirmar a remoção, o sistema exclui o usuário do banco de dados e apresenta uma mensagem de sucesso, encerrando o processo. Como pós-condição, o usuário removido perde o acesso ao sistema.

No cenário alternativo, o administrador pode tentar remover um usuário que possua permissões de acesso iguais ou superiores às suas próprias. Nessa situação, o sistema bloqueia a ação e exibe uma mensagem de restrição, impedindo a exclusão. Além disso, se o administrador desistir da remoção após visualizar os dados do usuário, o sistema cancela a operação e retorna à listagem.

Já o cenário de exceção contempla falhas técnicas que podem ocorrer durante o processo de exclusão. Caso o sistema apresente algum erro no momento da operação, é exibida uma mensagem informativa orientando o administrador a tentar novamente mais tarde.

## 4.14.9 UC08 – Visualizar Reservas

Figura 15 – UC08

Nome do Caso de Uso	
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	
<b>Atores Secundários</b>	
<b>Resumo</b>	
<b>Pré-condições</b>	
<b>Pós-condições</b>	
<b>Cenário Principal</b>	
Ações do Ator	Ações do Sistema
1. Acessa a funcionalidade "Visualizar Reservas"	1. Recupera todas as reservas do banco de dados
	2. Exibe a listagem com informações como espaço, usuário, data e horário da reserva
<b>Cenário Alternativo</b>	
Ações do ator	Ações do Sistema
1a. Aplica filtros de busca (por espaço, usuário, data.)	1a. Filtra os resultados conforme os critérios definidos
1b. Nenhuma reserva encontrada com os filtros	1b. Exibe mensagem "Nenhuma reserva encontrada com os critérios informados"
<b>Cenário de Exceção</b>	
Ações do ator	Ações do Sistema
1x. Tenta acessar as reservas mas há falha no sistema	1x. Exibe mensagem de erro técnico e orienta tentar novamente mais tarde

O caso de uso UC08 – Visualizar Reservas descreve a funcionalidade que permite ao administrador consultar todas as reservas realizadas na plataforma. Essa ação é essencial para fins de gestão e monitoramento, fornecendo visibilidade sobre o uso dos espaços pela organização e permitindo a tomada de decisões com base em dados históricos e atuais.

O ator principal é o Administrador, responsável por gerenciar os espaços e usuários. O Website atua como ator secundário, disponibilizando a interface para consulta dos registros de reserva.

Para acessar essa funcionalidade, o administrador deve estar devidamente

autenticado no sistema, atendendo à pré-condição estabelecida. Ao iniciar o processo, o administrador acessa a funcionalidade “Visualizar Reservas”. O sistema, então, recupera todas as reservas armazenadas no banco de dados e apresenta uma listagem contendo informações relevantes como o nome do espaço reservado, o usuário responsável pela reserva, além da data e horário correspondentes.

O caso de uso contempla ainda um cenário alternativo, no qual o administrador aplica filtros de busca com base em critérios como espaço, usuário ou data. O sistema filtra os resultados conforme os parâmetros fornecidos. Caso nenhum registro seja encontrado, o sistema exibirá uma mensagem informando que “Nenhuma reserva encontrada com os critérios informados”, incentivando o refinamento da pesquisa.

Em um cenário de exceção, falhas técnicas podem ocorrer durante a tentativa de acesso aos dados. Nessa situação, o sistema exibirá uma mensagem de erro e orientará o administrador a tentar novamente mais tarde, minimizando o impacto da falha momentânea e garantindo uma experiência informativa.

Ao final do processo, como pós-condição, as reservas são listadas com todos os dados relevantes, ficando disponíveis para consulta do administrador conforme as permissões do sistema.

## 4.14.10 UC09 – Reservar Espaço

Figura 16 – UC09

Nome do Caso de Uso		UC09 - Reserva Espaços	
Caso de Uso Geral		Visualizar Reservas	
Ator Principal		Administrador, Usuário	
Atores Secundários		Website	
Resumo		Permite realizar uma reserva de espaço informando data e horário.	
Pré-condições		O ator deve estar autenticado e o espaço desejado deve estar disponível.	
Pós-condições		A reserva será registrada no sistema e vinculada ao usuário.	
Cenário Principal			
Ações do Ator		Ações do Sistema	
1. Acessa a lista de espaços disponíveis		1. Exibe os espaços com botão para reserva	
2. Seleciona o espaço desejado		2. Exibe formulário de reserva com campos de data e horário	
3. Informa os dados da reserva			
4. Confirma a solicitação		3. Valida os dados e verifica disponibilidade	
		4. Registra a reserva no banco de dados	
		5. Exibe mensagem de sucesso	
Cenário Alternativo			
Ações do ator		Ações do Sistema	
3a. Informa horário já reservado		3a. Exibe mensagem de conflito de agendamento	
3b. Informa data ou horário inválido		3b. Exibe mensagem de erro indicando formato ou horário fora do expediente	
4a. Desiste da reserva antes da confirmação		4a. Cancela a operação e retorna à listagem de espaços	
4b. Corrige os dados após erro e reenviar		4b. Revalida e prossegue com o fluxo principal	
Cenário de Exceção			
Ações do ator		Ações do Sistema	
4x. Envia o formulário e há erro de conexão		4x. Exibe mensagem de erro e orienta o administrador a tentar novamente mais tarde	

O caso de uso UC09 – Reservar Espaços descreve a funcionalidade que permite ao usuário ou administrador realizar uma reserva de espaço por meio da definição de data e horário desejados. Essa funcionalidade é central para o propósito do sistema, que visa à organização eficiente da utilização de espaços compartilhados dentro de uma empresa ou instituição.

Os atores principais são o Administrador e o Usuário, enquanto o Website atua como ator secundário, fornecendo a interface pela qual as reservas são solicitadas e processadas.

As pré-condições exigem que o usuário esteja autenticado no sistema e que o

espaço selecionado esteja disponível no período informado. Uma vez atendidas essas condições, o processo inicia-se com o acesso à listagem de espaços disponíveis, onde o sistema exibe os ambientes com a opção de reserva. Após selecionar o espaço desejado, o sistema exibe um formulário contendo campos para preenchimento de data e horário. O usuário informa os dados e confirma a solicitação.

O sistema, então, valida os dados fornecidos e verifica a disponibilidade do espaço no período indicado. Estando tudo conforme, a reserva é registrada no banco de dados, e o sistema exibe uma mensagem de sucesso. Como pós-condição, a reserva passa a estar vinculada ao usuário e registrada no sistema.

O cenário alternativo contempla erros e inconsistências no preenchimento dos dados. Caso o usuário informe um horário já reservado, o sistema exibirá uma mensagem de conflito de agendamento. Se a data ou o horário estiverem em formato inválido ou fora do expediente, será exibida uma mensagem de erro com orientação adequada. O usuário pode então optar por cancelar a operação (retornando à listagem de espaços) ou corrigir os dados e reenviar a solicitação, momento em que o sistema revalidará as informações e dará prosseguimento ao fluxo normal.

No cenário de exceção, se ocorrer uma falha de conexão no momento do envio do formulário, o sistema apresentará uma mensagem de erro técnico, orientando o usuário a tentar novamente mais tarde.

## 4.14.11 UC10 – Cancelar Reserva

Figura 17 – UC10

Nome do Caso de Uso	
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	
<b>Atores Secundários</b>	
<b>Resumo</b>	
<b>Pré-condições</b>	
<b>Pós-condições</b>	
UC10 - Cancelar Reserva	
Reserva Espaços	
Administrador, Usuário	
Website	
Permite cancelar uma reserva registrada anteriormente.	
O ator deve estar autenticado e possuir uma reserva ativa.	
A reserva é removida do sistema e o horário liberado para nova utilização.	
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessa a lista de reservas feitas	1. Exibe reservas vinculadas ao usuário ou empresa
2. Seleciona a reserva a ser cancelada	2. Exibe os detalhes e a opção de cancelamento
3. Confirma o cancelamento	3. Remove a reserva do banco de dados
	4. Exibe mensagem de sucesso
Cenário Alternativo	
Ações do ator	Ações do Sistema
2a. Tenta cancelar uma reserva já iniciada	2a. Exibe mensagem de que não é possível cancelar reservas já iniciadas
3a. Desiste do cancelamento	3a. Cancela a operação e retorna à listagem de reservas
3b. Corrige filtros e encontra outra reserva	3b. Exibe os dados corretamente
Cenário de Exceção	
Ações do ator	Ações do Sistema
3x. Envia o formulário e há erro de conexão	3x. Exibe mensagem de erro e orienta o administrador a tentar novamente mais tarde

O caso de uso UC10 – Cancelar Reserva descreve a funcionalidade que permite ao administrador ou usuário cancelar uma reserva previamente registrada no sistema. Essa operação é essencial para manter a flexibilidade do uso dos espaços e garantir a liberação de horários anteriormente ocupados, promovendo um melhor aproveitamento dos recursos disponíveis.

Neste cenário, os atores principais são o Administrador e o Usuário, sendo o Website o ator secundário responsável por intermediar a interação entre o usuário e o sistema.

Como pré-condição, o ator deve estar devidamente autenticado e possuir uma

reserva ativa no sistema. O fluxo principal tem início quando o ator acessa a lista de reservas feitas. O sistema então exibe as reservas vinculadas ao usuário ou à empresa. Em seguida, o ator seleciona a reserva que deseja cancelar. O sistema apresenta os detalhes da reserva juntamente com a opção de cancelamento. Após a confirmação da operação, a reserva é removida do banco de dados e o sistema exibe uma mensagem de sucesso. Como pós-condição, o horário correspondente à reserva cancelada é liberado para nova utilização por outros usuários.

O cenário alternativo prevê situações comuns durante o processo. Caso o ator tente cancelar uma reserva cujo horário já tenha sido iniciado, o sistema impedirá a operação e exibirá uma mensagem informando que não é possível cancelar reservas já iniciadas. Se o ator desistir do cancelamento, a operação será cancelada e o sistema retornará à listagem de reservas. Alternativamente, o usuário pode ajustar os filtros de busca e localizar outra reserva, e o sistema então exibirá os dados correspondentes de forma correta.

No cenário de exceção, se houver falha de conexão no momento do envio do formulário de cancelamento, o sistema apresentará uma mensagem de erro técnico e orientará o usuário ou administrador a tentar novamente mais tarde, assegurando a continuidade do processo após a resolução da falha.



## 4.14.12 UC11 – Receber Confirmação de Reserva

Figura 18 – UC11

Nome do Caso de Uso	
<b>UC11 - Receber Confirmação de Reserva</b>	
<b>Caso de Uso Geral</b>	Reserva Espaços
<b>Ator Principal</b>	Usuário
<b>Atores Secundários</b>	Website
<b>Resumo</b>	Usuário recebe a confirmação de que sua reserva foi registrada com sucesso.
<b>Pré-condições</b>	A reserva deve ter sido realizada com sucesso.
<b>Pós-condições</b>	O usuário é notificado da confirmação e pode visualizar os detalhes da reserva.
<b>Cenário Principal</b>	
Ações do Ator	Ações do Sistema
	1. Registra a reserva no banco de dados
	2. Gera confirmação de reserva
	3. Notifica o usuário via interface, e-mail ou sistema de alertas
	4. Disponibiliza acesso rápido aos detalhes da reserva
<b>Cenário Alternativo</b>	
Ações do ator	Ações do Sistema
	3a. Usuário não visualiza a mensagem imediatamente
	3b. Mensagem permanece disponível na interface até ser visualizada
	3c. Sistema permite reenvio ou reexibição da confirmação
<b>Cenário de Exceção</b>	
Ações do ator	Ações do Sistema
	3x. Falha no envio da notificação automática
	3x. Exibe notificação alternativa na próxima seção do usuário
	3x. Opcionalmente, envia mensagem para o administrador sobre falha de notificação

O caso de uso UC11 – Receber Confirmação de Reserva tem como objetivo notificar o usuário após a realização bem-sucedida de uma reserva de espaço. Essa funcionalidade garante a comunicação clara e imediata com o usuário, confirmando o registro da operação e disponibilizando o acesso às informações da reserva efetuada.

O Usuário é o ator principal responsável pela ação de reserva, enquanto o Website atua como ator secundário, responsável por notificar o usuário por meio de canais definidos, como interface da aplicação, sistema de alertas ou e-mail.

A pré-condição para a execução deste caso de uso é que a reserva tenha sido registrada com sucesso no sistema. A partir disso, o fluxo principal do sistema inicia-

se com o registro da reserva no banco de dados e, em seguida, a geração automática de uma confirmação. O sistema então notifica o usuário por meio da interface, e-mail ou sistema de alertas e disponibiliza acesso rápido aos detalhes da reserva realizada, facilitando o acompanhamento da atividade.

Como pós-condição, o usuário é devidamente notificado sobre a confirmação da reserva e pode consultar os detalhes da operação a qualquer momento.

No cenário alternativo, o sistema prevê situações em que o usuário não visualiza imediatamente a notificação enviada. Nesses casos, a mensagem permanece disponível na interface do sistema até que seja visualizada. Além disso, o sistema oferece a opção de reenviar ou exibir novamente a confirmação da reserva, assegurando que o usuário tenha acesso à informação mesmo que ocorra atraso na leitura inicial.

No cenário de exceção, pode haver falhas técnicas no envio da notificação automática. Quando isso ocorre, o sistema exibe uma notificação alternativa assim que o usuário acessar a próxima seção da plataforma. Opcionalmente, o sistema pode também encaminhar uma mensagem ao administrador responsável, informando sobre a falha na entrega da notificação, possibilitando o acompanhamento e resolução do incidente.

## 4.14.13 UC12 – Visualizar Histórico de Reservas

Figura 19 – UC12

Nome do Caso de Uso	
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	
<b>Atores Secundários</b>	
<b>Resumo</b>	
<b>Pré-condições</b>	
<b>Pós-condições</b>	
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessa a opção "Histórico de Reservas"	1. Recupera todas as reservas passadas do usuário
	2. Exibe a lista com status, datas, horários e espaços
Cenário Alternativo	
Ações do ator	Ações do Sistema
1a. Aplica filtros por data ou status	1a. Aplica os filtros e atualiza a listagem
1b. Nenhum resultado encontrado	1b. Exibe mensagem "Nenhuma reserva encontrada para o filtro aplicado"
Cenário de Exceção	
Ações do ator	Ações do Sistema
1x. Acessa a funcionalidade, mas há falha de carregamento	1x. Exibe mensagem de erro técnico e orienta tentar novamente mais tarde

O caso de uso UC12 – Visualizar Histórico de Reservas tem como objetivo permitir que o usuário consulte todas as reservas feitas anteriormente na plataforma. Essa funcionalidade proporciona um acompanhamento detalhado das atividades passadas, permitindo ao usuário revisar dados relevantes como status, horários e espaços utilizados.

O Usuário é o ator principal, sendo o Website o ator secundário responsável por fornecer a interface necessária para o acesso ao histórico de dados.

Como pré-condição, é necessário que o usuário esteja devidamente autenticado no sistema. A partir disso, o fluxo principal inicia-se quando o usuário

acessa a opção "Histórico de Reservas". O sistema, então, recupera todas as reservas passadas vinculadas àquele usuário e exibe uma listagem contendo informações detalhadas, como status, datas, horários e os respectivos espaços utilizados. Como pós-condição, essas informações são disponibilizadas para consulta de forma organizada e clara.

No cenário alternativo, o usuário pode optar por aplicar filtros de busca com base em data ou status da reserva. O sistema aplica os filtros selecionados e atualiza a listagem conforme os critérios definidos. Caso nenhuma reserva seja encontrada com os parâmetros fornecidos, o sistema exibe a mensagem: "Nenhuma reserva encontrada para o filtro aplicado", incentivando o refinamento da pesquisa.

No cenário de exceção, pode ocorrer falha técnica no carregamento da funcionalidade. Nesse caso, o sistema apresenta uma mensagem de erro técnico e orienta o usuário a tentar novamente mais tarde, garantindo que o processo possa ser retomado posteriormente com segurança e estabilidade.

#### 4.15 Arquitetura da Solução

A arquitetura da solução proposta foi concebida com base no modelo cliente-servidor, buscando garantir uma estrutura organizada, escalável e de fácil manutenção (PRESSMAN; MAXIM, 2016). A aplicação será composta por três camadas principais: apresentação, lógica de negócios e persistência de dados. Essa separação de responsabilidades favorece a clareza na implementação das funcionalidades, o reaproveitamento de componentes e a evolução contínua da plataforma, conforme os princípios da arquitetura em camadas descritos por Sommerville (2011).

A interação entre os usuários e o sistema se dará por meio de navegadores modernos, por uma interface desenvolvida com foco em responsividade (CAMPOS, 2017), garantindo plena usabilidade em diferentes tamanhos de tela, com prioridade para dispositivos móveis, como smartphones e tablets. A camada de apresentação (frontend), desenvolvida em React com TypeScript, é orientada à construção de interfaces reativas e componentes reutilizáveis, o que contribui para a manutenibilidade e a escalabilidade do sistema.

O acesso à plataforma será feito tanto por administradores quanto por usuários comuns, cada qual com permissões específicas, em conformidade com o modelo de controle de acesso baseado em papéis (RBAC) recomendado por Ferreira e Araújo (2022). A lógica da aplicação (backend), implementada com base em princípios de orientação a objetos, será responsável por processar as ações do usuário, validar dados, aplicar regras de negócio e intermediar a comunicação com o banco de dados. O uso de REST APIs garante desacoplamento entre as camadas e favorece futuras integrações externas (FIELDING, 2000).

A camada de persistência será construída com PostgreSQL, um sistema gerenciador de banco de dados relacional robusto e confiável, adequado para aplicações que exigem consistência, integridade e capacidade de crescimento (DATE, 2019). A modelagem do banco de dados contemplará entidades como usuários, espaços, reservas e cancelamentos, com uso de chaves primárias e estrangeiras para garantir os relacionamentos corretos entre os dados. O uso do Prisma ORM contribuirá para a

abstração da lógica de acesso aos dados, reduzindo a complexidade e aumentando a segurança e produtividade no desenvolvimento.

Para garantir a escalabilidade, a arquitetura suporta replicação horizontal do backend, além de práticas como separação de leitura e escrita no banco de dados, caching com Redis para dados de acesso frequente e balanceamento de carga em ambientes com múltiplas instâncias (TANENBAUM; VAN STEEN, 2007). O sistema foi planejado para comportar aumento gradual de usuários e dados sem comprometimento da performance, sendo passível de implantação em infraestrutura em nuvem com escalonamento automático (auto-scaling), conforme necessário.

No que diz respeito à confiabilidade, a aplicação implementa mecanismos de logging, tratamento centralizado de exceções, testes automatizados e monitoramento contínuo de serviços. Esses aspectos são fundamentais para garantir alta disponibilidade e rápida recuperação diante de falhas (BASS; CLEMENTS; KAZMAN, 2012). O uso de versionamento via Git, pipelines de integração contínua (CI/CD) e testes unitários/integrados contribui ainda mais para a estabilidade do sistema.

Com relação à segurança, a arquitetura prevê a adoção de mecanismos robustos de autenticação e controle de acesso, com validação de credenciais e diferenciação de permissões conforme o perfil do usuário. As senhas dos usuários são armazenadas com uso de hashing seguro (como bcrypt), evitando exposição direta em caso de vazamento de dados. Todas as informações sensíveis são tratadas conforme os princípios estabelecidos pela Lei Geral de Proteção de Dados (BRASIL, 2018), incluindo limitação no uso de dados pessoais, armazenamento seguro e rastreabilidade das ações realizadas na plataforma (SILVA et al., 2020). A comunicação entre cliente e servidor é protegida com criptografia TLS via protocolo HTTPS.

Por fim, a solução foi projetada com foco em extensibilidade e integração com outros sistemas. A estrutura adotada permitirá, por exemplo, a futura incorporação de APIs externas, envio de notificações por e-mail, integração com ferramentas como Google Calendar ou Microsoft Bookings (MICROSOFT, 2024), entre outras funcionalidades adicionais. Dessa forma, o sistema não apenas atende às necessidades

iniciais de controle de espaços e reservas, mas também está preparado para acompanhar a evolução de demandas futuras da organização.

#### 4.15.1 Estratégias de Segurança

A segurança da informação é um dos pilares essenciais na construção de aplicações web, sendo responsável por garantir a confidencialidade, integridade e disponibilidade dos dados e serviços oferecidos (TANENBAUM; WETHERALL, 2011). No contexto do sistema LinkSpace, foram adotadas estratégias que contemplam boas práticas consolidadas no desenvolvimento seguro de software, com ênfase na proteção de dados pessoais, prevenção de vulnerabilidades e controle de acesso robusto.

A autenticação de usuários será realizada por meio de tokens JWT (JSON Web Token), permitindo sessões seguras e escaláveis, sem a necessidade de armazenamento de estado no servidor (FERREIRA; ARAÚJO, 2022). O token é assinado digitalmente, o que evita adulterações, e tem validade temporária definida, o que reduz riscos em casos de interceptação. A aplicação adota o modelo de autorização baseada em papéis (RBAC), restringindo o acesso às funcionalidades com base no perfil do usuário (administrador ou usuário comum), alinhando-se às diretrizes do OWASP para sistemas com múltiplos níveis de permissão (OWASP, 2023).

As senhas dos usuários são armazenadas utilizando algoritmos de hash criptográfico como o bcrypt, amplamente recomendado por sua resistência a ataques de força bruta e rainbow tables, especialmente por incorporar *salting* e fator de custo configurável (MAZZA, 2020). Além disso, são aplicadas políticas de senha forte durante o cadastro e redefinição, incentivando o uso de caracteres diversos e comprimento mínimo adequado.

Quanto à proteção contra vulnerabilidades comuns, a aplicação segue os princípios definidos pelo projeto OWASP Top 10, mitigando riscos como injeção de código (SQL Injection), cross-site scripting (XSS), e falsificação de requisição entre sites (CSRF). As entradas do usuário passam por processos de sanitização e validação, e o backend define regras claras de tratamento para evitar execução de código malicioso ou acesso indevido a dados.

Para proteção do tráfego de dados, todas as comunicações entre cliente e servidor são realizadas por meio de HTTPS (Hypertext Transfer Protocol Secure), utilizando certificados válidos e criptografia TLS (Transport Layer Security), o que assegura que os dados trafeguem de forma cifrada e não sejam interceptados por terceiros (BRANDÃO; REZENDE, 2021).

Adicionalmente, o sistema contempla:

- Rate limiting, para prevenir ataques de força bruta e negação de serviço (DoS);
- Logs de auditoria com rastreamento de ações críticas (login, edição e exclusão de dados), garantindo transparência e rastreabilidade conforme previsto pela Lei Geral de Proteção de Dados – LGPD (BRASIL, 2018);
- Separação entre variáveis sensíveis e o código-fonte, utilizando variáveis de ambiente (.env) para armazenar chaves e senhas de uso interno (Google Developers, 2024).

Por fim, cabe ressaltar que a segurança é tratada como um processo contínuo. O sistema será submetido a testes automatizados e revisões periódicas de código com foco em segurança (code reviews), promovendo uma cultura de desenvolvimento seguro desde as fases iniciais até a manutenção evolutiva da aplicação (SOMMERVILLE, 2011).



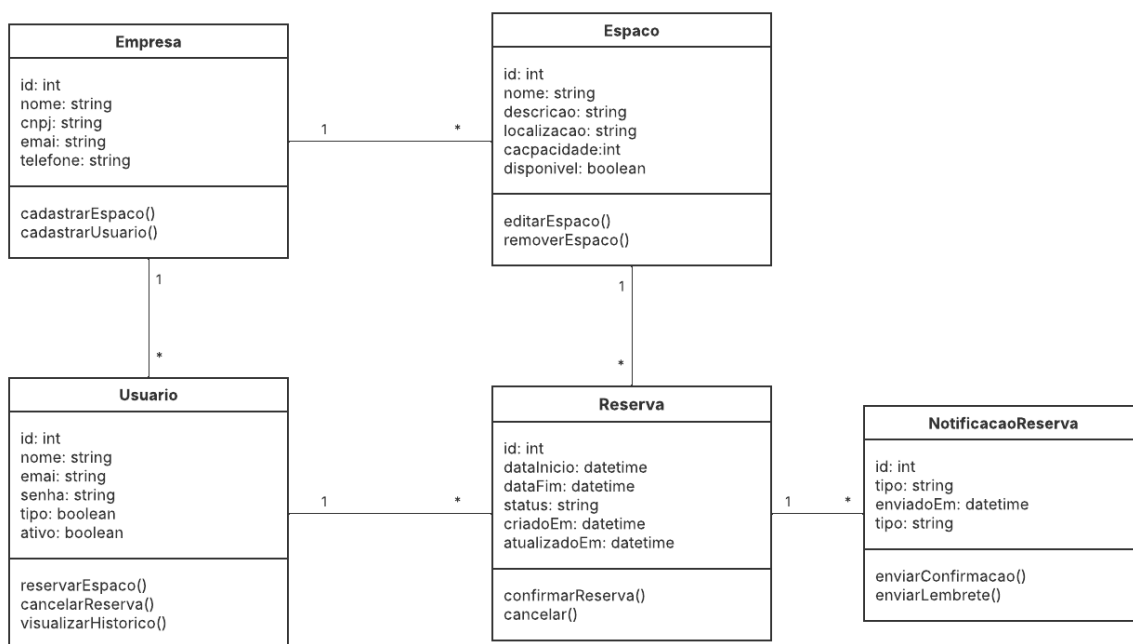
## 4.16 Diagrama de classe e diagramas de sequência elaborados segundo a documentação dos casos de uso

### 4.16.1 Diagrama de classe

O diagrama de classe é uma das principais representações da UML (Unified Modeling Language) e tem como objetivo descrever a estrutura estática de um sistema, detalhando as classes que o compõem, seus atributos, métodos (operações) e os relacionamentos entre elas. É uma ferramenta fundamental na modelagem orientada a objetos, pois permite representar a arquitetura lógica da aplicação antes da implementação do código, facilitando o entendimento, o planejamento e a manutenção do sistema.

No contexto da solução desenvolvida para o sistema de gestão e reserva de espaços compartilhados, o diagrama de classe apresentado organiza as principais entidades envolvidas na operação do sistema, destacando suas responsabilidades e os vínculos que estabelecem entre si, conforme imagem a seguir.

Figura 20 – Diagrama de Classe



A classe Empresa representa a entidade responsável pela administração dos espaços e usuários. Ela possui atributos como id, nome, cnpj, email e telefone, e métodos que permitem o cadastro de espaços e usuários, além da visualização do histórico de atividades.

A classe Usuario está associada diretamente à empresa e representa os indivíduos que utilizam o sistema, podendo ser administradores ou usuários comuns (identificados por meio do atributo tipo). Ela armazena dados como nome, email, senha e ativo, e possui métodos para ações administrativas e operacionais, como cancelamento de reservas e edição ou remoção do próprio cadastro.

A classe Espaco contempla os ambientes disponíveis para reserva. Entre seus atributos estão nome, descricao, localizacao, capacidade e disponivel. Seus métodos permitem a edição, remoção, visualização e reserva dos espaços.

A classe Reserva é o núcleo operacional do sistema, sendo responsável por registrar as informações relacionadas às reservas realizadas pelos usuários. Seus atributos incluem dataInicio, dataFim, status, criadoEm e atualizadoEm. Esta classe está associada a um usuário e a um espaço, representando a intermediação entre os dois. Possui também métodos como confirmarReserva() e cancelar(), que refletem as ações possíveis sobre uma reserva.

Por fim, a classe NotificacaoReserva é responsável por representar o envio de confirmações e alertas ao usuário após a efetivação de uma reserva. Ela possui atributos como tipo, enviadoEm e um método enviarConfirmacao(). Essa classe está associada à reserva e serve como suporte à comunicação do sistema com o usuário.

Os relacionamentos entre as classes seguem uma lógica coerente com o domínio do problema. Uma empresa pode ter vários usuários e espaços, enquanto cada reserva está associada a um único usuário e a um único espaço. Além disso, cada reserva pode gerar uma ou mais notificações ao usuário.

Esse diagrama de classes representa de forma clara e estruturada a lógica do sistema, permitindo que se compreenda facilmente os componentes principais, suas

responsabilidades e as relações existentes entre eles. Ele serve como base para o desenvolvimento do sistema, além de auxiliar na documentação e comunicação técnica entre os membros da equipe.

O sistema de reservas permite que um usuário autenticado visualize espaços disponíveis e realize uma reserva para um período específico. O processo inclui a verificação obrigatória da disponibilidade do espaço, garantindo que não haja conflitos de horários. Se disponível, a reserva é registrada e o espaço é marcado como ocupado; caso contrário, o sistema sugere outros horários. O diagrama de classes mostra quatro entidades principais: Usuário, Reserva, Espaço e Administrador. O usuário faz reservas, o espaço pode ser reservado e verificado quanto à disponibilidade, a reserva armazena os dados do agendamento, e o administrador gerencia o sistema. O diagrama de sequência ilustra o passo a passo do processo de reserva, desde o login até a confirmação por e-mail. O sistema garante a integridade das informações e responde com mensagens claras em caso de erro ou indisponibilidade. Esses diagramas descrevem tanto a estrutura quanto o comportamento do sistema de forma clara e funcional.

#### **4.17 Diagrama de Sequência**

O diagrama de sequência é um dos principais diagramas comportamentais da UML (Unified Modeling Language) e tem como finalidade representar graficamente a ordem cronológica das interações entre os diversos objetos e componentes do sistema durante a execução de um determinado cenário. Por meio desse diagrama, é possível visualizar de forma clara e sequencial como os atores interagem com o sistema, quais mensagens são trocadas entre os elementos e em que ordem essas interações ocorrem.

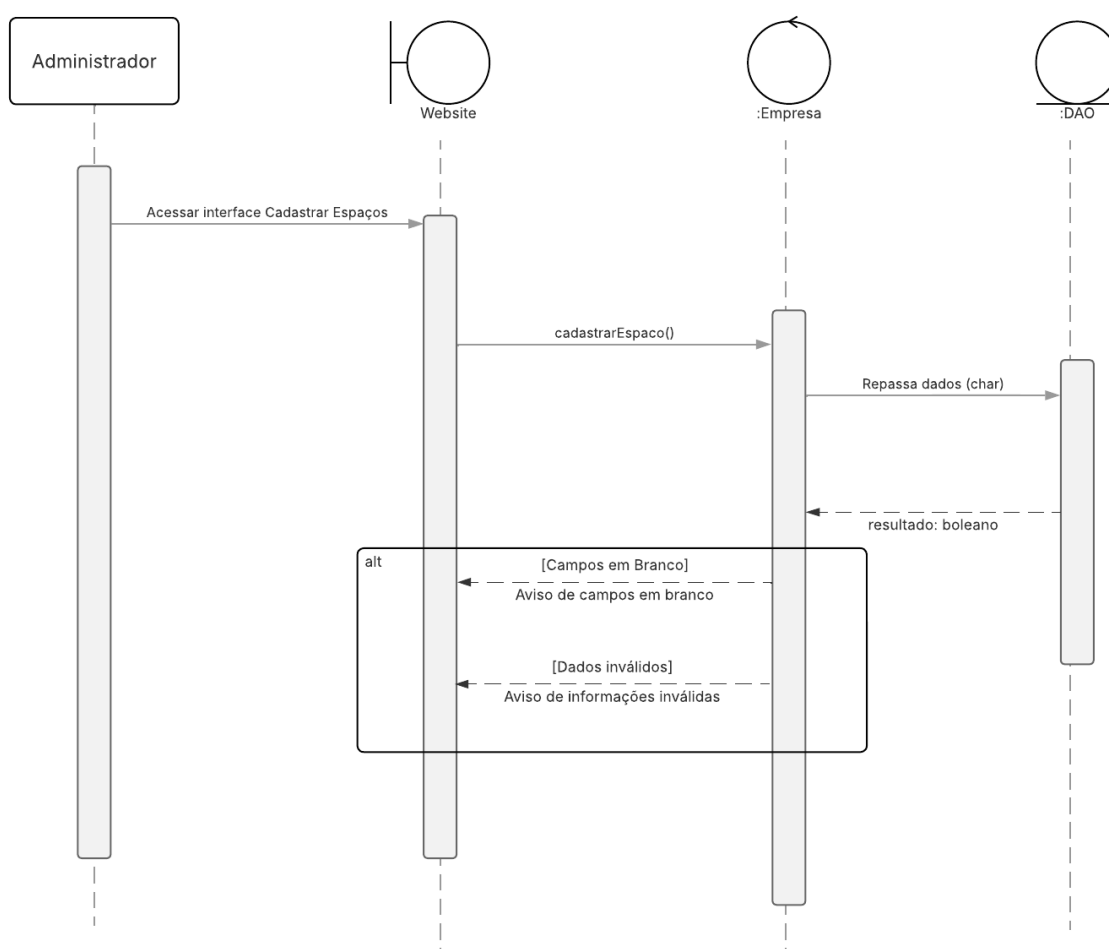
No contexto deste projeto, os diagramas de sequência foram desenvolvidos com base direta na documentação dos casos de uso, servindo como um complemento visual para cada um dos fluxos principais definidos. Cada caso de uso documentado (UC01 a UC12) possui um respectivo diagrama de sequência, que descreve o comportamento dinâmico do sistema frente à execução da funcionalidade associada.

Dessa forma, os diagramas contribuem significativamente para o entendimento do sistema por desenvolvedores, analistas e demais stakeholders, pois traduzem os requisitos funcionais em um modelo visual de fácil interpretação.

Além de reforçarem o alinhamento com os fluxos descritos nos casos de uso, os diagramas de sequência também possibilitam identificar variações, ramificações de fluxo e exceções tratadas no sistema, garantindo uma visão completa e técnica do comportamento da aplicação.

A seguir, são apresentados os diagramas de sequência correspondentes aos casos de uso UC01 a UC12:

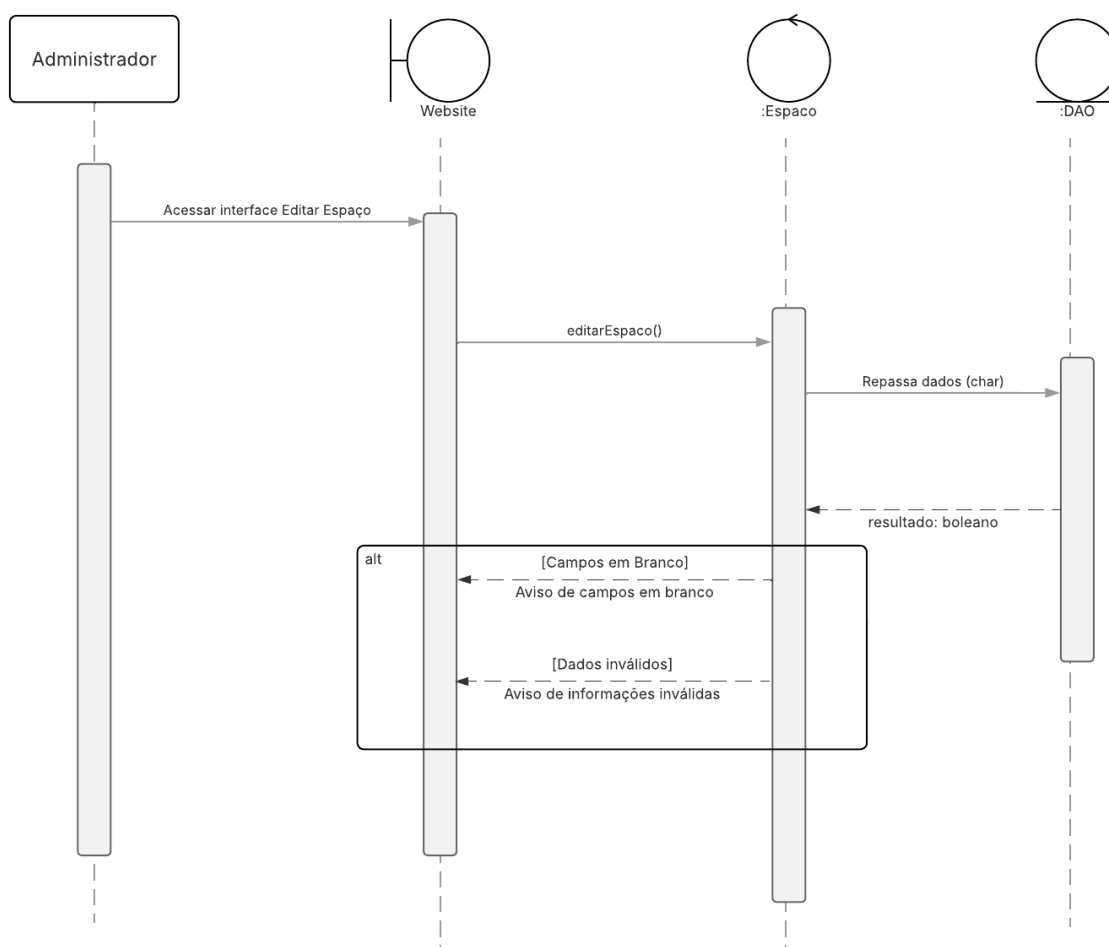
Figura 21 – UC01 – Cadastrar Espaço



Conforme a Figura 21 acima, o diagrama de sequência representa o processo de cadastro de espaços pelo administrador. O fluxo inicia-se com o acesso à interface

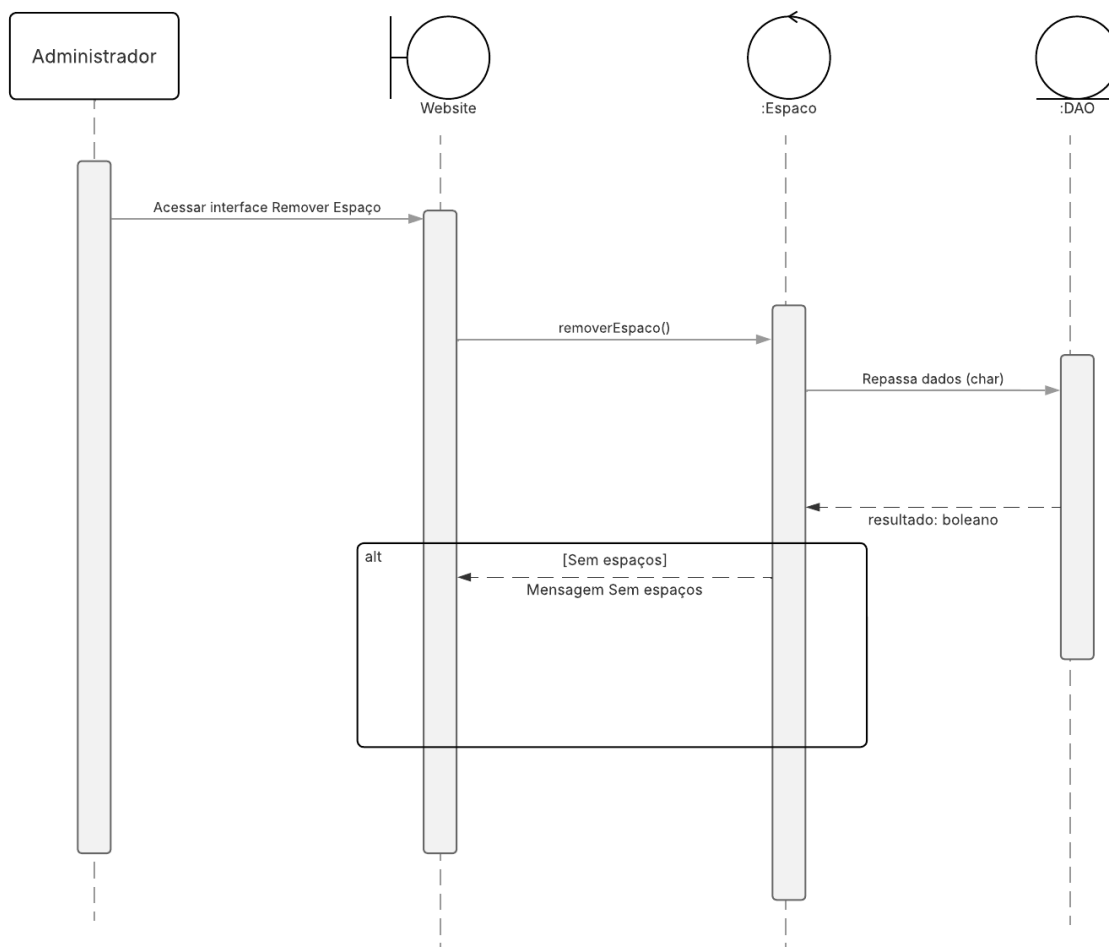
de cadastro e envio dos dados ao sistema, que os repassa para a camada de negócio e, em seguida, para o DAO responsável pelo banco de dados. O sistema valida as informações e, caso existam campos em branco ou dados inválidos, retorna mensagens de erro ao usuário, impedindo o cadastro até a correção.

Figura 22 – UC02 – Editar Espaço



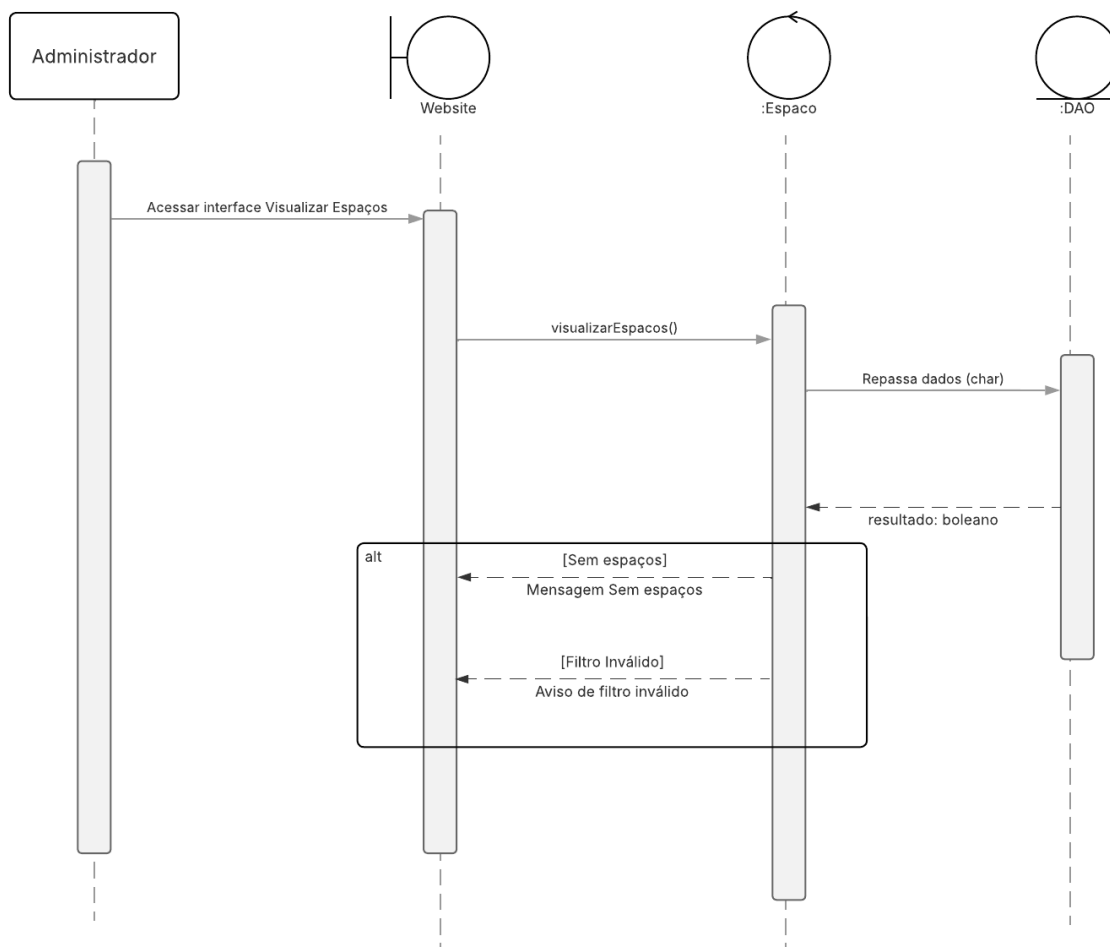
Conforme a Figura 22 acima, o diagrama de sequência descreve o processo de edição de um espaço pelo administrador. Após acessar a interface de edição, os dados atualizados são enviados ao sistema, que os valida e repassa para a camada de negócio e, posteriormente, ao DAO para atualização no banco. Caso existam campos em branco ou dados inválidos, o sistema exibe mensagens de aviso, impedindo a finalização da edição até que os erros sejam corrigidos.

Figura 23 – Remover Espaço



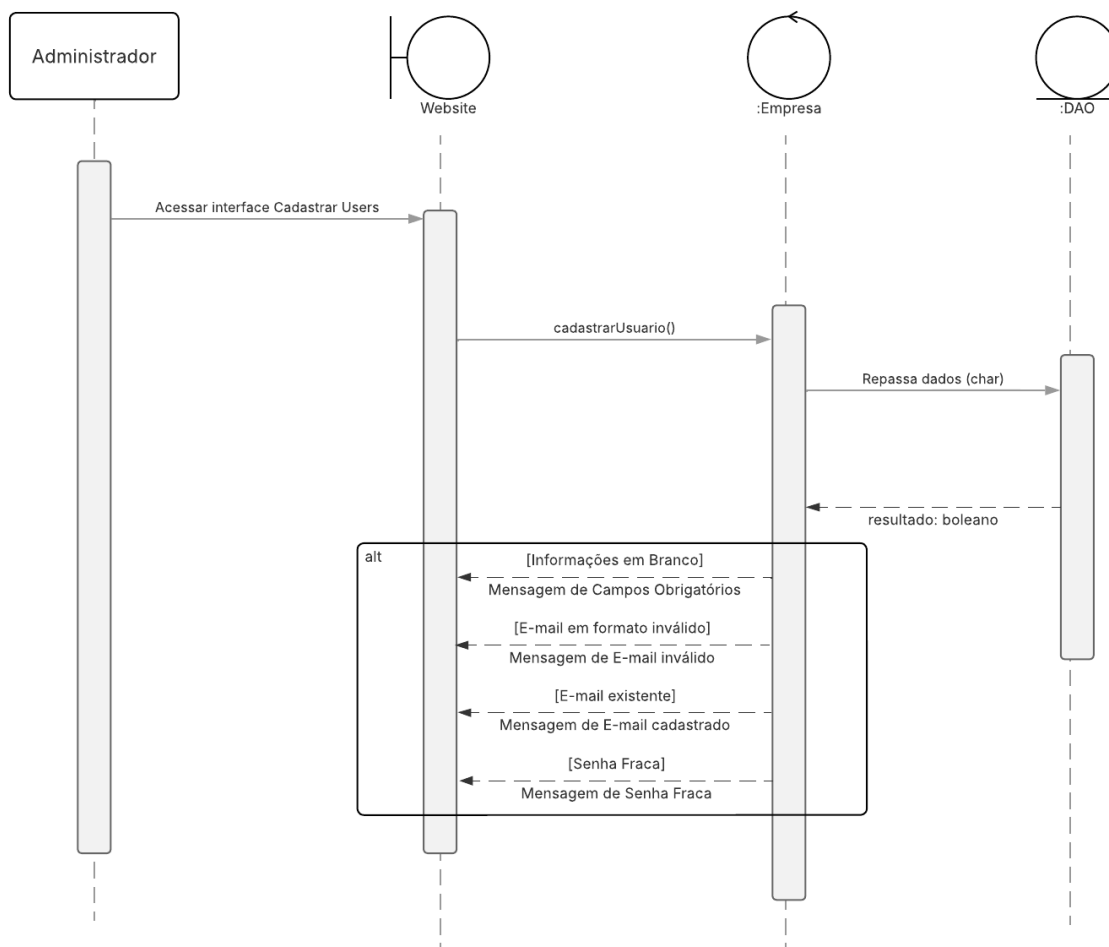
Conforme a Figura 23 acima, o diagrama de sequência mostra o processo de remoção de um espaço por parte do administrador. Após acessar a interface de remoção, o sistema envia a requisição para a camada de negócio, que repassa os dados ao DAO para exclusão no banco de dados. Caso não haja espaços disponíveis para remoção, o sistema exibe uma mensagem informando essa condição e interrompe a operação.

Figura 24 – UC04 – Visualizar Espaços



Conforme a Figura 24 acima, o diagrama de sequência ilustra o processo de visualização de espaços por parte do administrador. Após acessar a interface correspondente, o sistema realiza a requisição à camada de negócio, que consulta os dados no DAO. Caso não existam espaços cadastrados ou o filtro aplicado seja inválido, o sistema apresenta mensagens específicas para cada situação, informando a ausência de registros ou a necessidade de correção do filtro.

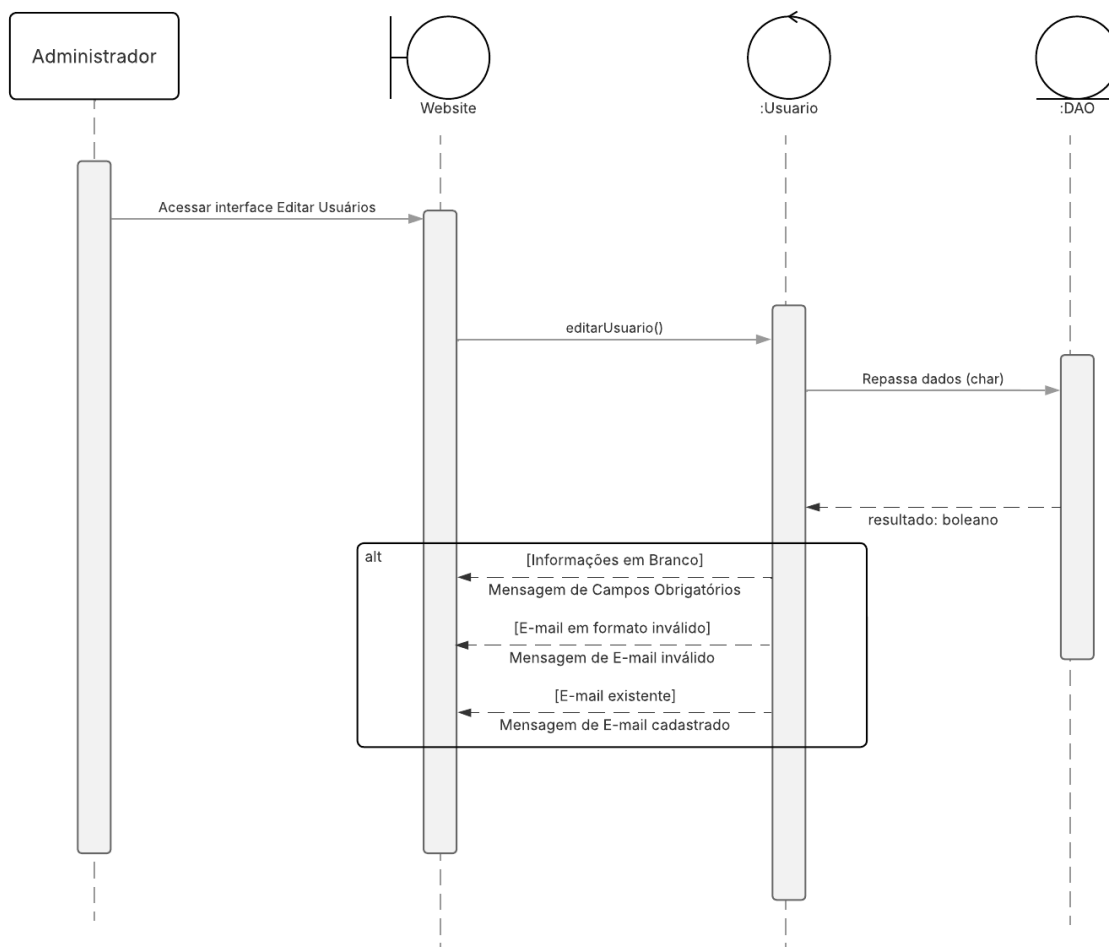
Figura 25 – UC05 – Cadastrar Usuário



Conforme a Figura 25 acima, o diagrama de sequência apresenta o fluxo de cadastro de usuários realizado pelo administrador. Após acessar a interface, os dados do novo usuário são enviados ao sistema, que os valida e encaminha para a camada de negócio e, posteriormente, ao DAO. O sistema verifica diversos cenários de erro, como campos obrigatórios em branco, e-mail inválido, e-mail já cadastrado e senha fraca, retornando mensagens específicas para que o administrador corrija as informações antes de concluir o cadastro.

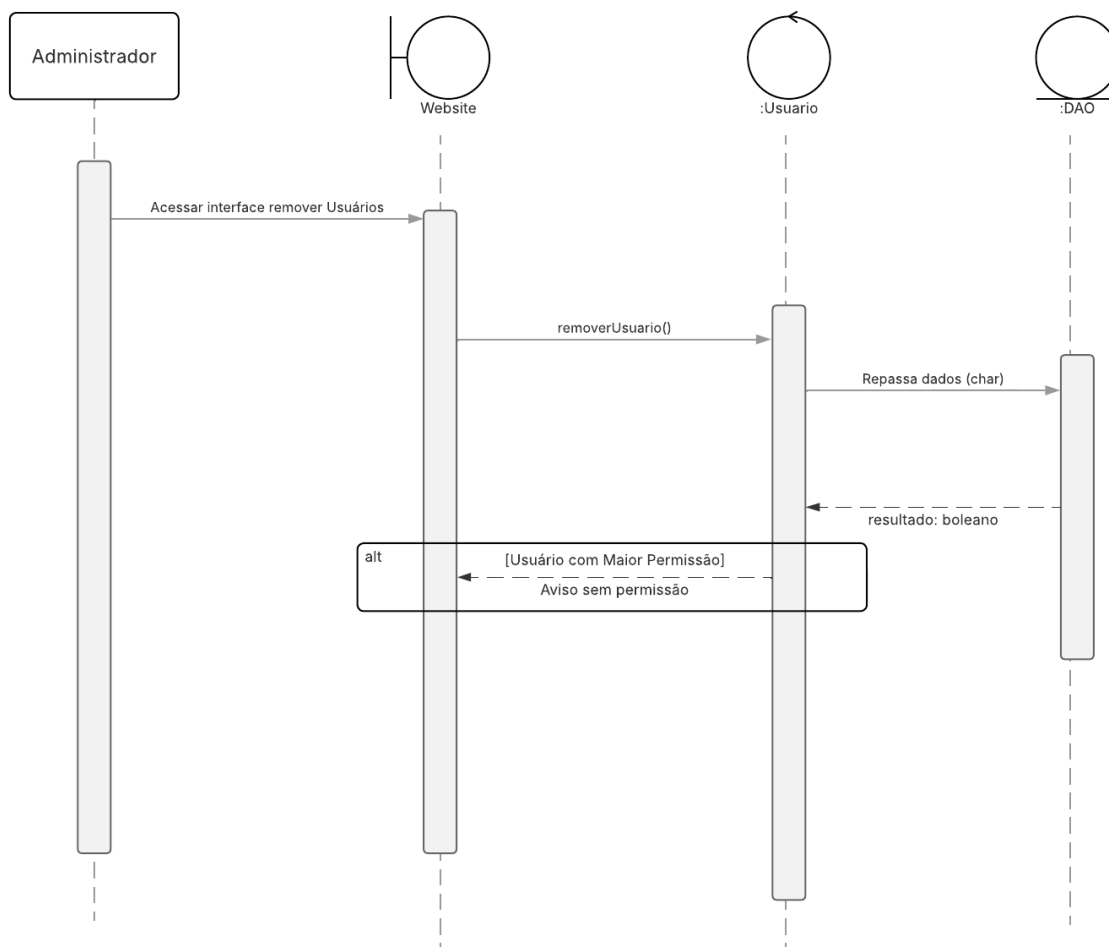


Figura 26 – UC06 - Editar Usuário



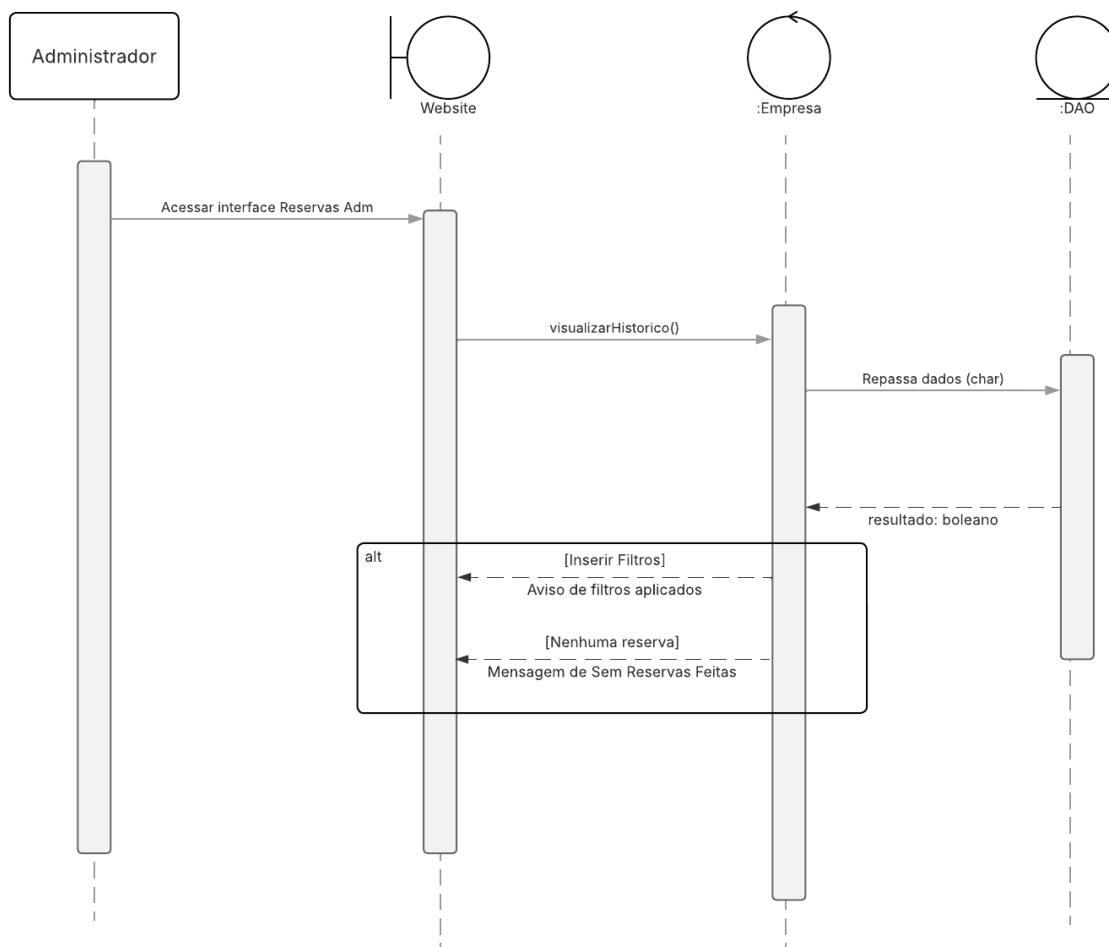
Conforme a Figura 26 acima, o diagrama de sequência representa o processo de edição de dados de um usuário pelo administrador. Após acessar a interface de edição, as informações são enviadas ao sistema, que repassa os dados para a camada de negócio e, posteriormente, ao DAO. O sistema realiza validações e, caso existam campos obrigatórios em branco, e-mail em formato inválido ou e-mail já cadastrado, são exibidas mensagens específicas informando os erros ao administrador.

Figura 27 – UC07 – Remover Usuário



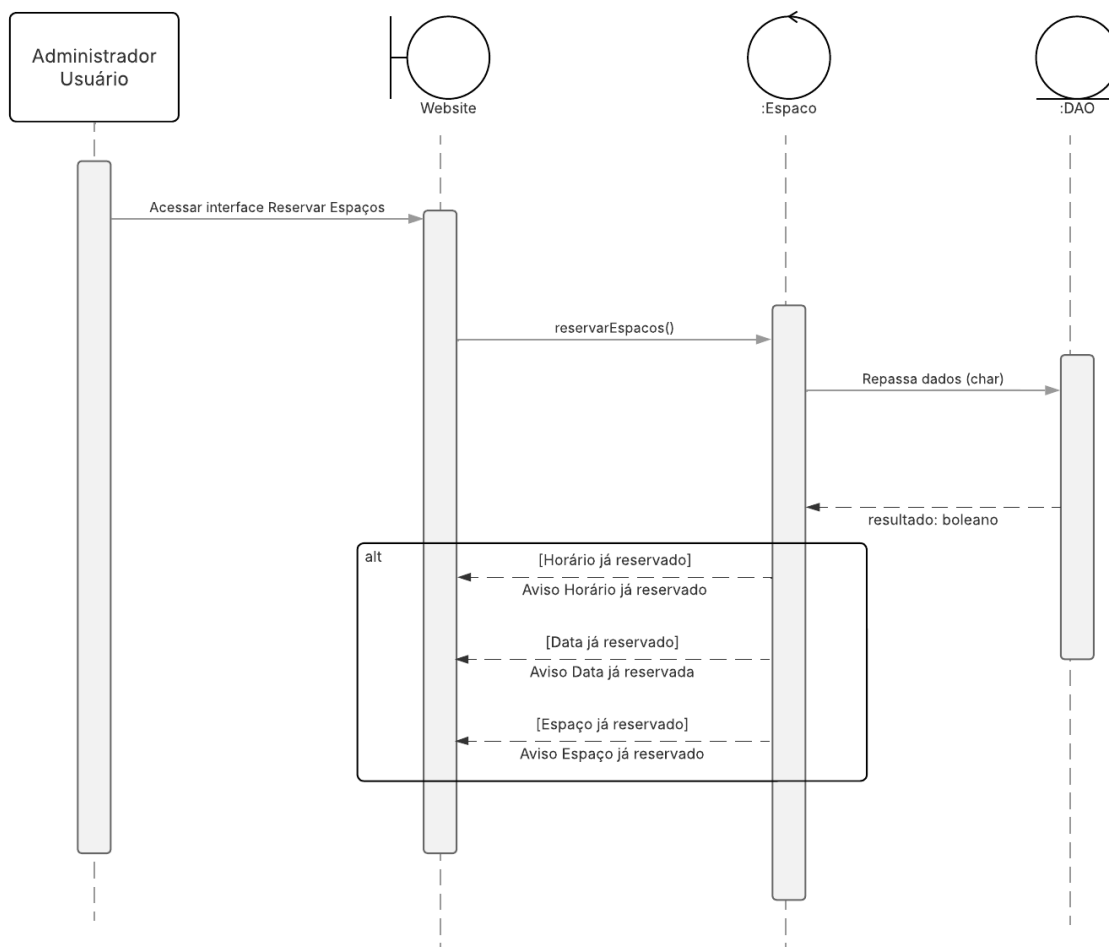
A Figura 27 ilustra o processo de remoção de usuários pelo administrador. Após acessar a interface, o sistema envia a solicitação à camada de negócio, que repassa os dados ao DAO para efetuar a exclusão. O diagrama também contempla a verificação de permissões: caso o usuário selecionado possua um nível de acesso superior ou igual ao do administrador responsável, o sistema exibe uma mensagem informando que a ação não é permitida.

Figura 28 – UC08 – Visualizar Reservas



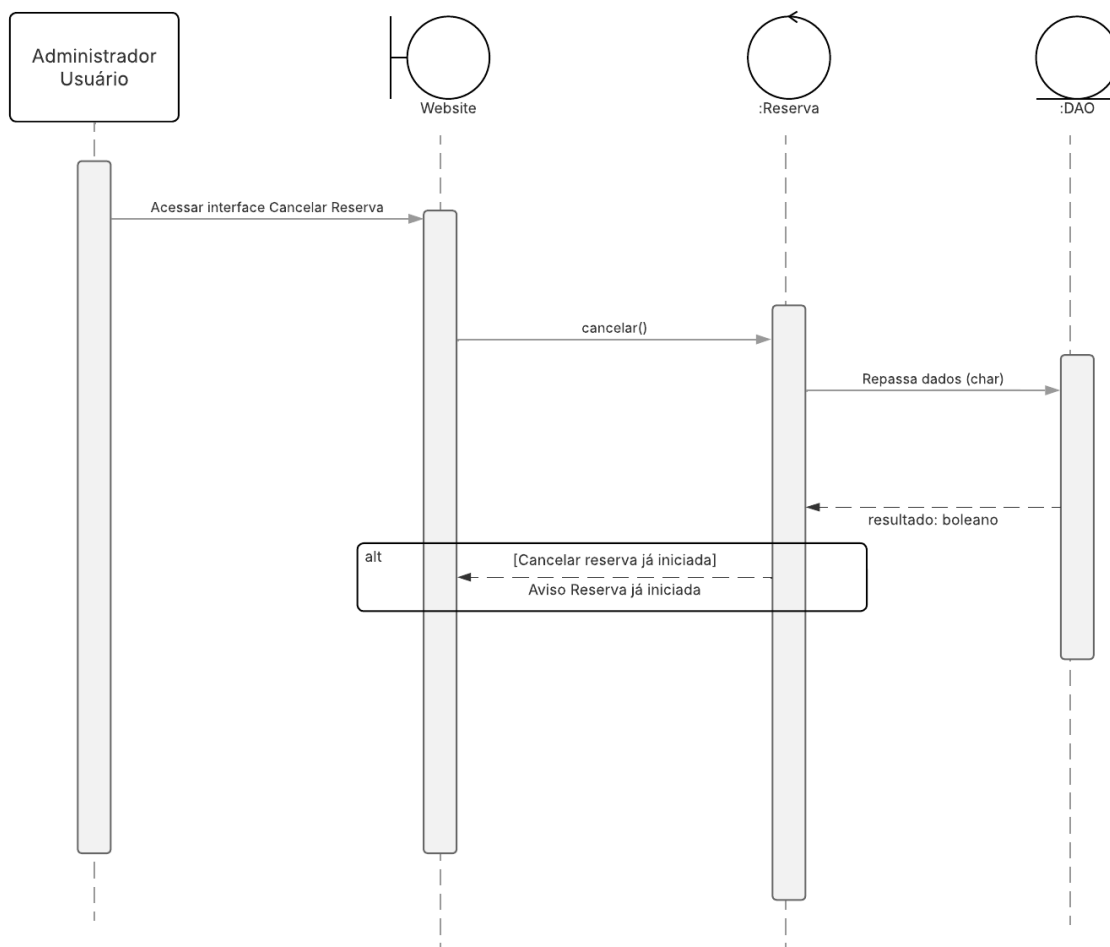
A Figura 28 representa o fluxo de visualização de reservas realizado pelo administrador. Após acessar a interface, o sistema solicita os dados à camada de negócio, que os repassa ao DAO. O retorno contém o histórico das reservas filtradas. Caso nenhum filtro seja aplicado ou nenhuma reserva esteja registrada, o sistema exibe mensagens específicas informando sobre os filtros utilizados ou a ausência de registros.

Figura 29 – UC09 - Reservar Espaços



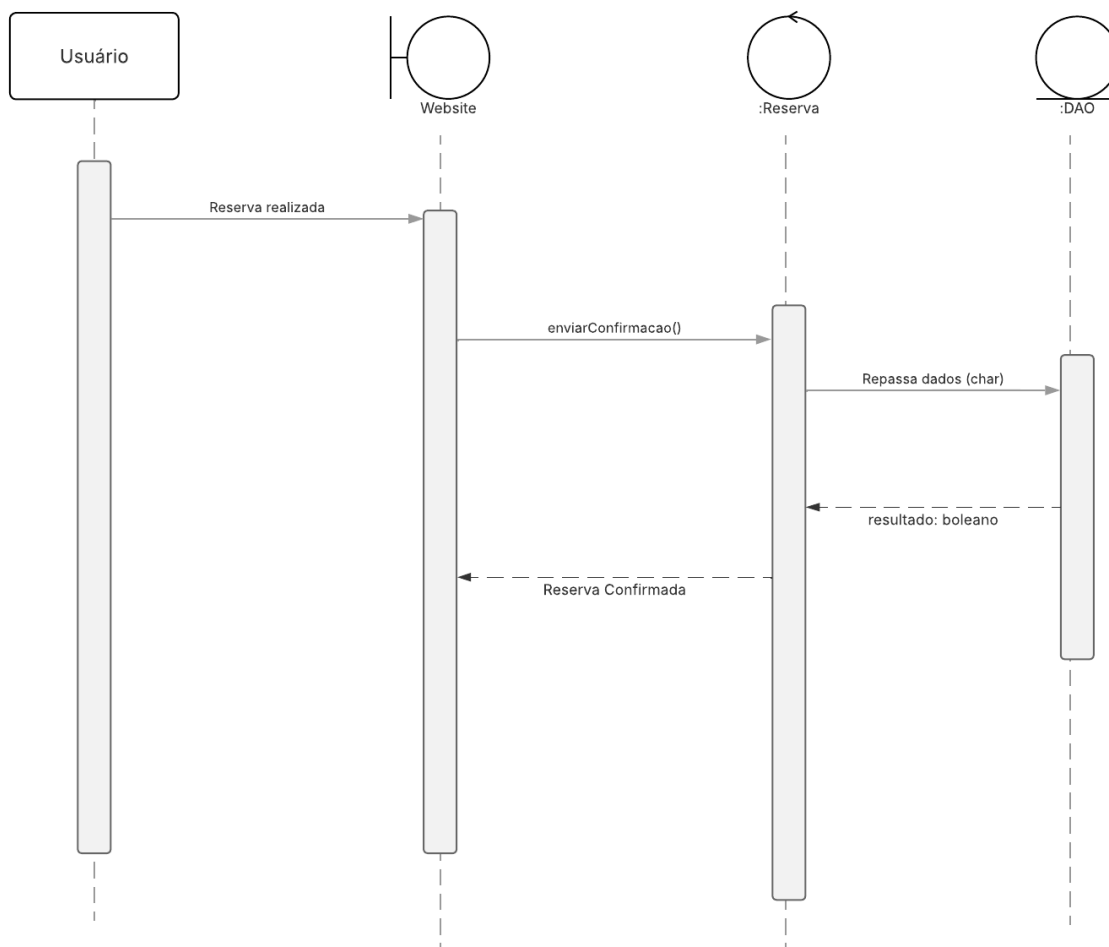
Na Figura 29, observa-se o fluxo de reserva de espaços realizado por administradores ou usuários comuns. O processo tem início com o acesso à interface de reserva, seguido pelo envio das informações ao sistema, que repassa os dados à camada de negócio e ao DAO. O sistema verifica possíveis conflitos e, se o horário, a data ou o espaço já estiverem reservados, são exibidas mensagens específicas informando a indisponibilidade para que o usuário selecione outra opção.

Figura 30 – UC10 – Cancelar Reserva



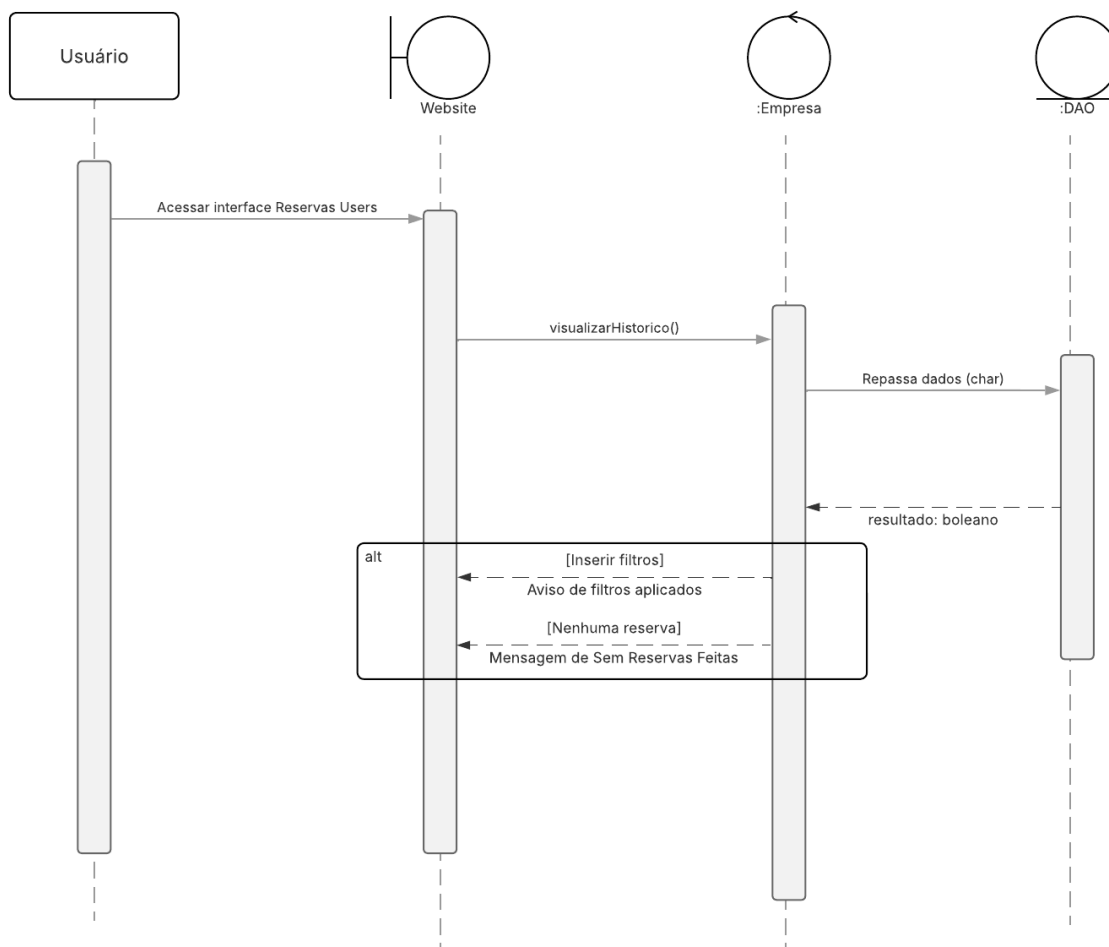
Através da Figura 30, é possível compreender o fluxo de cancelamento de reservas por parte de administradores ou usuários. O processo se inicia com o acesso à interface de cancelamento e envio da solicitação ao sistema, que verifica, por meio da lógica de negócio e do DAO, se a reserva ainda pode ser cancelada. Caso a reserva já tenha sido iniciada, o sistema retorna uma mensagem informando que o cancelamento não é mais permitido.

Figura 31 – UC11 – Receber Confirmação de Reserva



Como apresentado na Figura 31, o diagrama de sequência demonstra o processo de confirmação automática de uma reserva realizada por um usuário. Após a efetivação da reserva, o sistema aciona a lógica de negócio para confirmar a operação, enviando os dados ao DAO. Com o retorno positivo, o sistema notifica o usuário com uma mensagem de confirmação da reserva concluída com sucesso.

Figura 32 – UC12 – Visualizar Histórico de Reservas



A Figura 32 descreve o fluxo em que o usuário acessa o histórico de reservas realizadas. Após entrar na interface apropriada, o sistema aciona a lógica de negócio para consultar os dados junto ao DAO. O retorno depende da existência de registros e da aplicação de filtros. Se não houver reservas feitas ou os filtros estiverem incorretos, o sistema informa o usuário por meio de mensagens específicas.

#### 4.18 Projeto lógico de banco de dados MER/DER

O projeto lógico do banco de dados representa uma etapa fundamental no desenvolvimento da solução, pois define de forma precisa como os dados serão estruturados, organizados e relacionados no ambiente relacional. Por meio do Diagrama Entidade-Relacionamento (DER), é possível identificar as entidades principais do sistema, os atributos que as compõem, suas chaves primárias e estrangeiras, além dos relacionamentos que ocorrem entre elas. Essa modelagem reflete diretamente os requisitos funcionais extraídos dos casos de uso e serve como base para a criação do banco de dados físico.

O modelo apresentado contempla cinco entidades principais: Empresa, Usuário, Espaço, Reserva e NotificaçãoReserva. A entidade Empresa armazena informações institucionais como nome, CNPJ, e-mail e telefone, e relaciona-se com as entidades Usuário e Espaço, representando que cada empresa pode cadastrar múltiplos usuários e espaços. A entidade Usuário, por sua vez, possui atributos como nome, e-mail, senha, tipo de usuário (administrador ou comum) e ativo, além de estar vinculada a uma empresa, o que reforça a estrutura multiempresa da aplicação.

A entidade Espaço contém os dados dos ambientes que podem ser reservados, incluindo nome, descrição, localização, capacidade e status de disponibilidade. Cada espaço pertence a uma empresa, estabelecendo uma ligação direta com a entidade Empresa por meio de chave estrangeira.

A entidade Reserva é central para o funcionamento do sistema, conectando usuários e espaços e armazenando informações como data e hora de início e fim da reserva, status da operação e timestamps de criação e atualização. Trata-se de uma entidade associativa que reflete o processo de uso dos espaços disponíveis, conforme previsto nos casos de uso UC09 (Reservar Espaços) e UC10 (Cancelar Reserva).

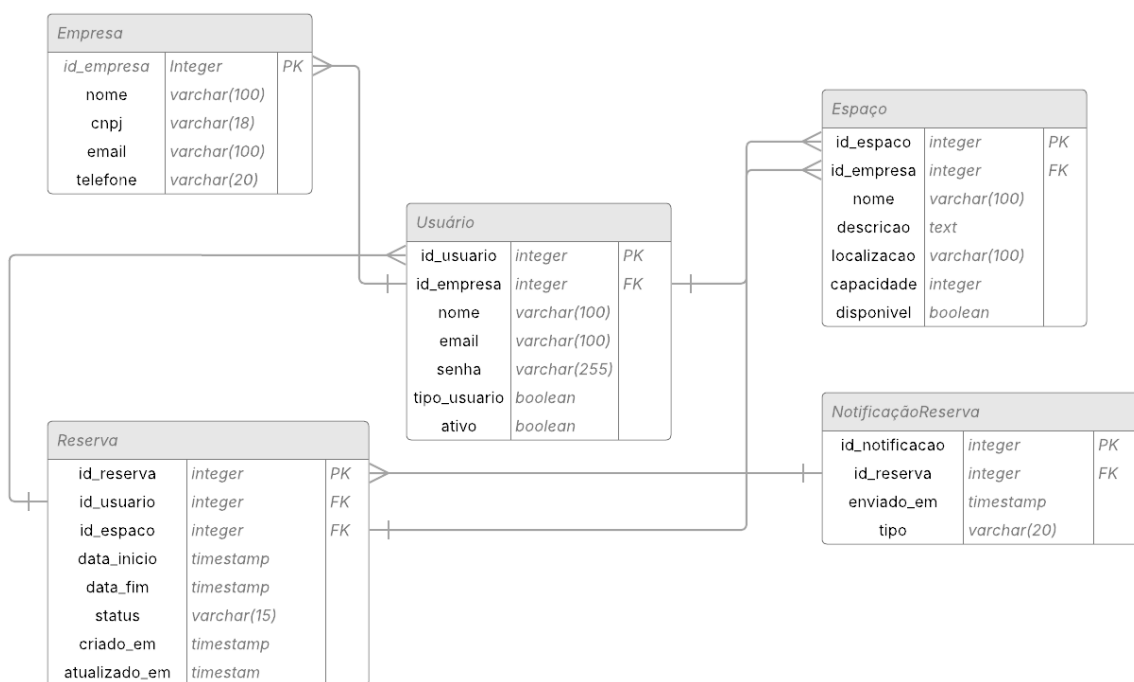
Por fim, a entidade NotificaçãoReserva registra os alertas ou confirmações enviadas ao usuário após a efetivação de uma reserva. Ela contém o tipo da notificação, a data de envio e a referência à reserva correspondente, reforçando a rastreabilidade do processo e o suporte ao caso de uso UC11 (Receber Confirmação



de Reserva).

A imagem a seguir ilustra o Diagrama Entidade-Relacionamento (DER) da solução, com a modelagem relacional proposta:

Figura 33 – Diagrama de Banco de Dados



Esse projeto lógico garante a consistência e integridade das informações, permitindo consultas otimizadas, controle adequado de relacionamentos e suporte total às funcionalidades do sistema. Ele está diretamente alinhado com os fluxos definidos nos casos de uso e diagramas de sequência, consolidando-se como um dos principais pilares técnicos da aplicação.

#### 4.19 Modelo CRC

O modelo CRC (Classe – Responsabilidade – Colaboração) é uma técnica utilizada na análise e design orientado a objetos que auxilia na definição do comportamento e das interações entre as classes de um sistema. Essa abordagem propõe a descrição de cada classe com foco em duas dimensões principais: suas responsabilidades, que representam as funções e dados que ela deve manter ou gerenciar, e suas colaborações, que indicam com quais outras classes ela interage para cumprir essas responsabilidades.

No projeto da solução de gestão e reserva de espaços compartilhados, a modelagem CRC foi utilizada como ferramenta complementar para consolidar a compreensão das principais entidades do sistema e seu comportamento no fluxo de funcionamento da aplicação. A seguir, apresenta-se a síntese da modelagem CRC das principais classes:

##### 4.19.1 CRC - Empresa

Tabela 15 – CRC (Empresa)

NOME DE CLASSE	EMPRESA
RESPONSABILIDADES	COLABORAÇÕES
ID	
NOME	
CNPJ	
EMAIL	
TELEFONE	

A classe Empresa representa a entidade responsável por organizar e gerenciar os espaços e usuários vinculados. Suas responsabilidades estão voltadas à

identificação e informações institucionais, como nome, CNPJ, e-mail e telefone. Embora não possua interações diretas complexas com outras classes, a empresa é a origem das relações hierárquicas com as classes Usuário e Espaço, às quais fornece o vínculo organizacional.

#### 4.19.2 CRC - Usuário

Tabela 16 – CRC (Usuário)

NOME DE CLASSE	USUÁRIO
RESPONSABILIDADES	COLABORAÇÕES
ID	receber o ID da empresa
NOME	
EMAIL	
SENHA	
TIPO	receber o tipo de usuário pela empresa "usuário" ou "administrador"
ATIVO	receber o status de atividade pela empresa

A classe Usuário possui como responsabilidade armazenar as informações pessoais e de acesso dos indivíduos cadastrados na plataforma, como nome, e-mail, senha, tipo (usuário comum ou administrador) e status de atividade. Colabora com a classe Empresa, da qual recebe o identificador e os parâmetros iniciais de configuração (tipos e status). Esta classe interage diretamente com outras funcionalidades, como a realização de reservas e o recebimento de notificações.

## 4.19.3 CRC - Espaço

Tabela 17 – CRC (Espaço)

NOME DE CLASSE	ESPAÇO
RESPONSABILIDADES	COLABORAÇÕES
ID	receber o id pela empresa
NOME	receber o nome pelo "administrador"
DESCRIÇÃO	receber uma descrição pelo "administrador"
LOCALIZAÇÃO	receber pelo "administrador" a definição de uma localização para o espaço
CAPACIDADE	receber pelo "administrador" a capacidade de espaços disponíveis nesta localização
DISPONIVEL	receber pelo "administrador" o status de "disponibilidade" do local a ser reservado

A classe Espaço representa os ambientes físicos disponíveis para reserva. Sua principal responsabilidade é manter os dados associados à identificação, descrição, localização, capacidade e status de disponibilidade. Suas colaborações ocorrem principalmente com a classe Administrador (subtipo de Usuário), que fornece as informações necessárias para a criação e manutenção dos espaços. Também colabora com a classe Empresa, da qual herda a referência organizacional.

## 4.19.4 CRC - Reserva

Tabela 18 – CRC (Reserva)

NOME DE CLASSE	RESERVA
RESPONSABILIDADES	COLABORAÇÕES
ID	receber o ID pela empresa
DATA INICIO	receber a data pelo "usuário" após a reserva
DATA FIM	receber a data pelo "usuário" após a reserva
STATUS	
CRIADO EM	receber a atualização pelo "usuário" após a reserva
ATUALIZADO EM	

A classe Reserva centraliza os dados relacionados às ações de agendamento de uso dos espaços. Suas responsabilidades incluem o armazenamento das datas de início e fim, status da reserva, e os registros de criação e atualização. Colabora diretamente com a classe Usuário, responsável por iniciar o processo de reserva, e com a classe Espaço, que fornece a disponibilidade. Além disso, serve como base para a geração de notificações por meio da classe NotificaçãoReserva.

## 4.19.5 CRC – Notificação

Tabela 19 – CRC (Notificação)

NOME DE CLASSE	NOTIFICAÇÃO
RESPONSABILIDADES	COLABORAÇÕES
ID	receber o ID da notificação pela empresa
TIPO	enviar as informações da reserva para o sistema
ENVIADO	o sistema envia uma notificação para o usuário e para o administrador informando os dados da reserva

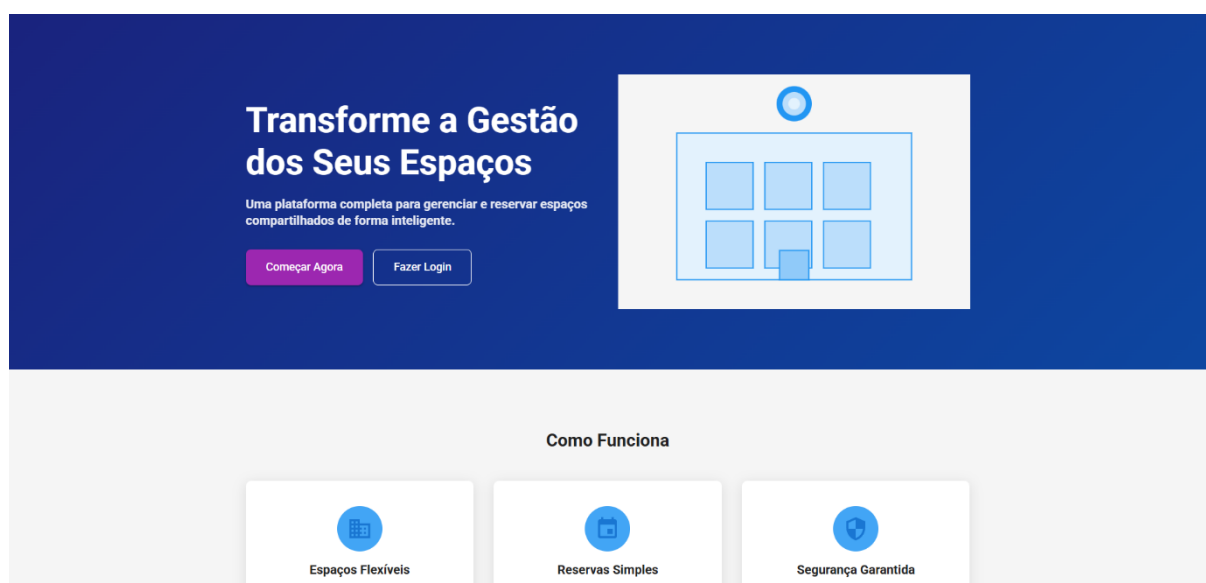
A classe NotificaçãoReserva é responsável por registrar e disparar notificações referentes às ações de reserva. Suas responsabilidades incluem armazenar o tipo de notificação, a data de envio e o vínculo com a reserva correspondente. Colabora com o sistema, que utiliza essas informações para comunicar os usuários (e administradores) a respeito do status de suas reservas, funcionando como um elo de comunicação interna do processo.

A modelagem CRC contribui para o refinamento das responsabilidades de cada classe, promovendo uma arquitetura mais coesa, com menor acoplamento e maior clareza nas colaborações entre os componentes do sistema. Essa estrutura, alinhada com os diagramas de classe e casos de uso, reforça a integridade conceitual da solução desenvolvida.

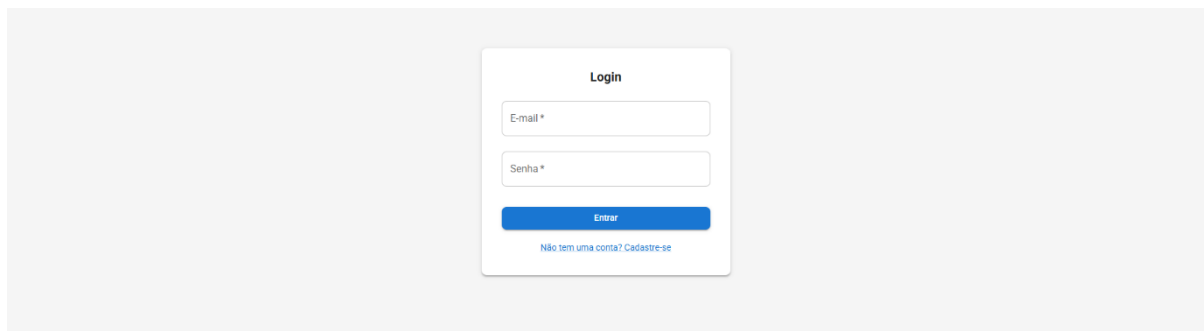
#### 4.20 Protótipo do projeto (de alta fidelidade)

O protótipo da solução desenvolvida para o sistema LinkSpace foi projetado com foco na experiência do usuário e na eficiência da navegação entre as funcionalidades principais da plataforma. A interface segue uma estrutura limpa e moderna, com organização em painéis laterais e seções bem definidas, garantindo fácil acesso às opções do sistema por parte de administradores e usuários.

Figura 34 – Página Inicial

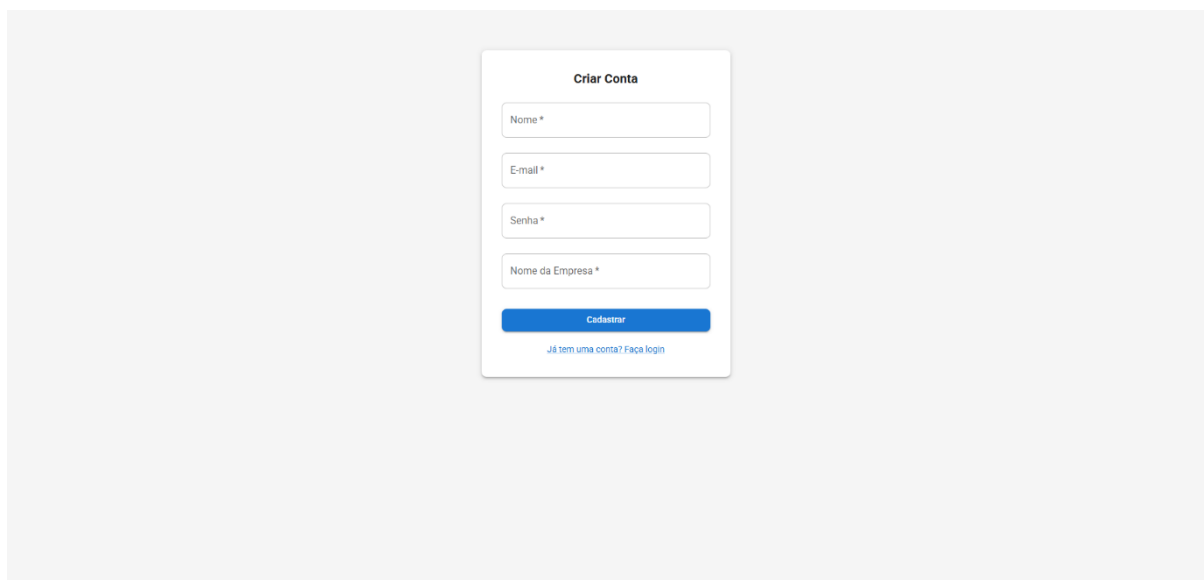


A página inicial foi elaborada para transmitir de forma imediata a proposta da aplicação: gestão e reservas de espaços compartilhados. Nela, o usuário encontra chamadas claras para ação, com botões de destaque que permitem iniciar rapidamente o cadastro ou efetuar login. O design prioriza simplicidade, evitando sobrecarga de informações, ao mesmo tempo em que reforça a identidade visual do sistema.

*Figura 35 – Página de Login*

A imagem mostra a interface de login do sistema. No topo do formulário, o título "Login" está centralizado. Abaixo dele, há dois campos de entrada: "E-mail \*" e "Senha \*", ambos com bordas arredondadas e ícones de lupa para pesquisa. Um botão azul com o texto "Entrar" em branco está posicionado abaixo dos campos. Na base do formulário, há um link azul que diz "Não tem uma conta? Cadastre-se".

A tela de login representa o ponto de entrada no sistema, contemplando os dois perfis existentes: administrador e usuário comum. O formulário apresenta campos de e-mail e senha com validações básicas de preenchimento, reforçando boas práticas de usabilidade. O layout busca minimizar erros, oferecendo mensagens de feedback claras quando as credenciais são inválidas. Essa etapa garante a segurança do acesso e o controle sobre os recursos do LinkSpace.

*Figura 36 – Página Criar Conta*

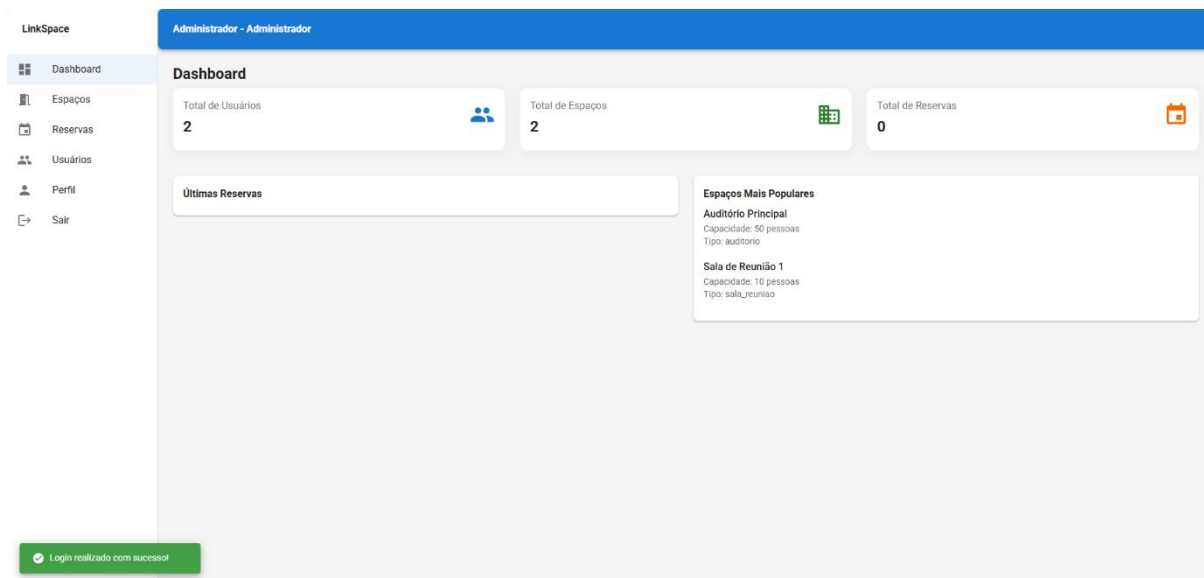
A imagem mostra a interface para criar uma nova conta. O formulário é intitulado "Criar Conta" no topo. Ele contém quatro campos de entrada: "Nome \*", "E-mail \*", "Senha \*" e "Nome da Empresa \*", todos com bordas arredondadas. Um botão azul com o texto "Cadastrar" em branco está localizado abaixo dos campos. Na base do formulário, há um link azul que diz "Já tem uma conta? Faça login".

O processo de criação de conta foi estruturado para ser simples e direto, coletando apenas informações essenciais: nome, e-mail, senha e empresa. Essa decisão está alinhada ao objetivo de reduzir barreiras de entrada e estimular novos cadastros. O layout privilegia campos bem espaçados, com botões de ação destacados, permitindo que mesmo usuários com pouca familiaridade tecnológica



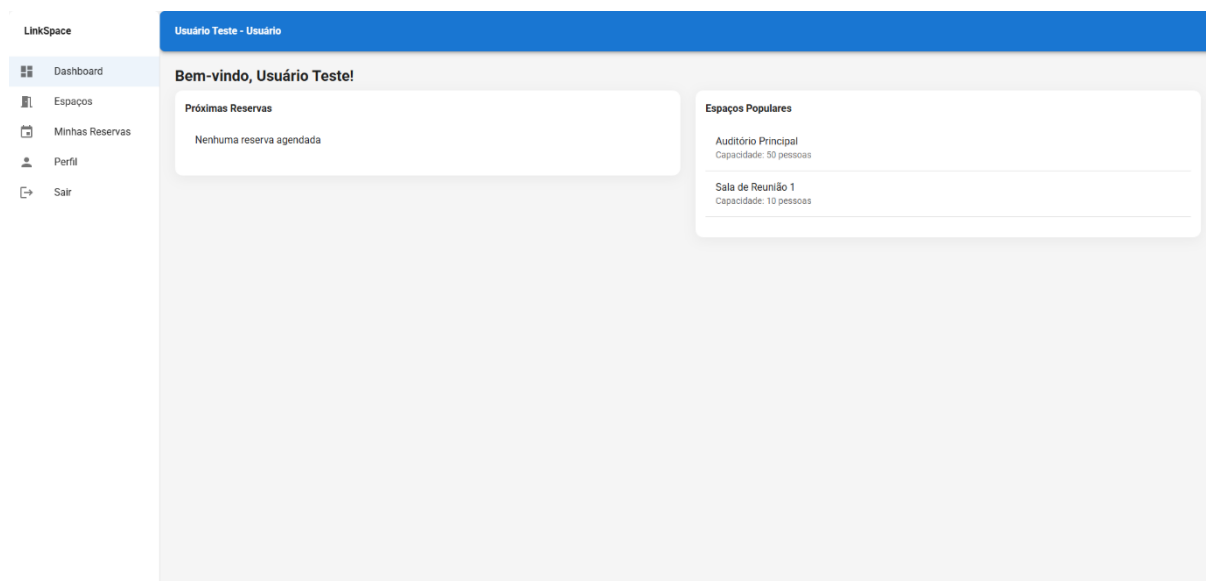
consigam concluir o processo sem dificuldades.

*Figura 37 – Dashboard do Administrador*



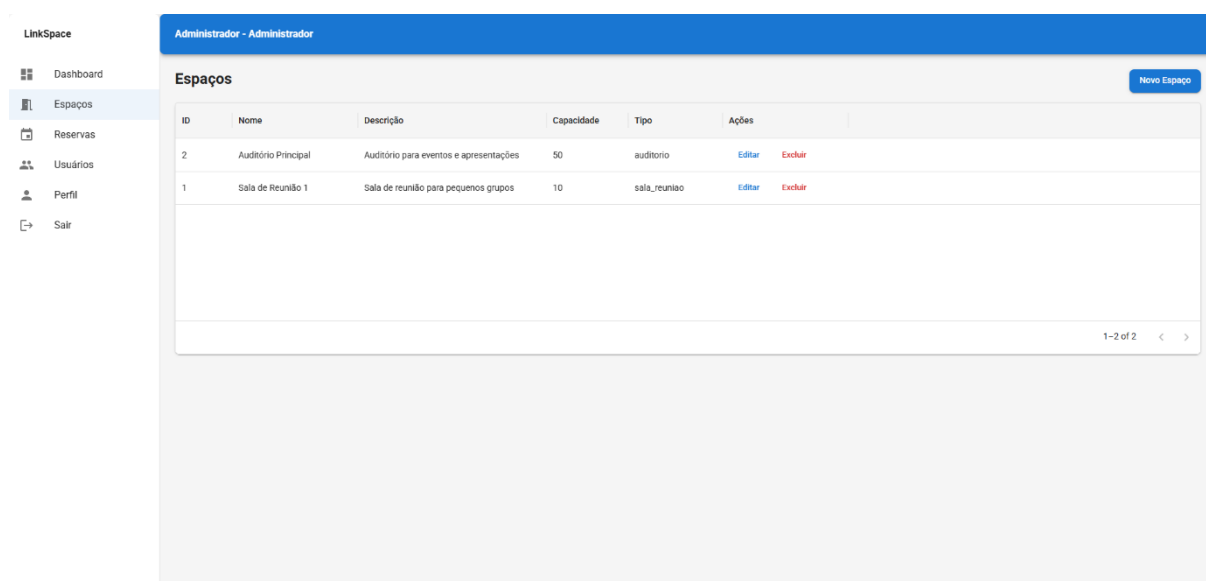
O painel do administrador centraliza indicadores fundamentais do sistema, como número de usuários, espaços cadastrados e reservas realizadas. Além disso, exibe os espaços mais utilizados, permitindo identificar padrões de uso. Essa visão panorâmica apoia o administrador na tomada de decisões estratégicas e no acompanhamento do sistema em tempo real. O layout privilegia blocos organizados e de fácil leitura, garantindo que as informações críticas estejam sempre visíveis de forma clara.

Figura 38 – Dashboard do Usuário Comum



O painel do usuário comum foi projetado com foco na experiência individual, destacando as próximas reservas e os espaços mais populares. Essa abordagem facilita a tomada de decisão e incentiva o engajamento no uso da plataforma. O texto de boas-vindas personalizado reforça o caráter amigável do sistema. A interface equilibra informações funcionais e simplicidade, contribuindo para uma experiência positiva e intuitiva.

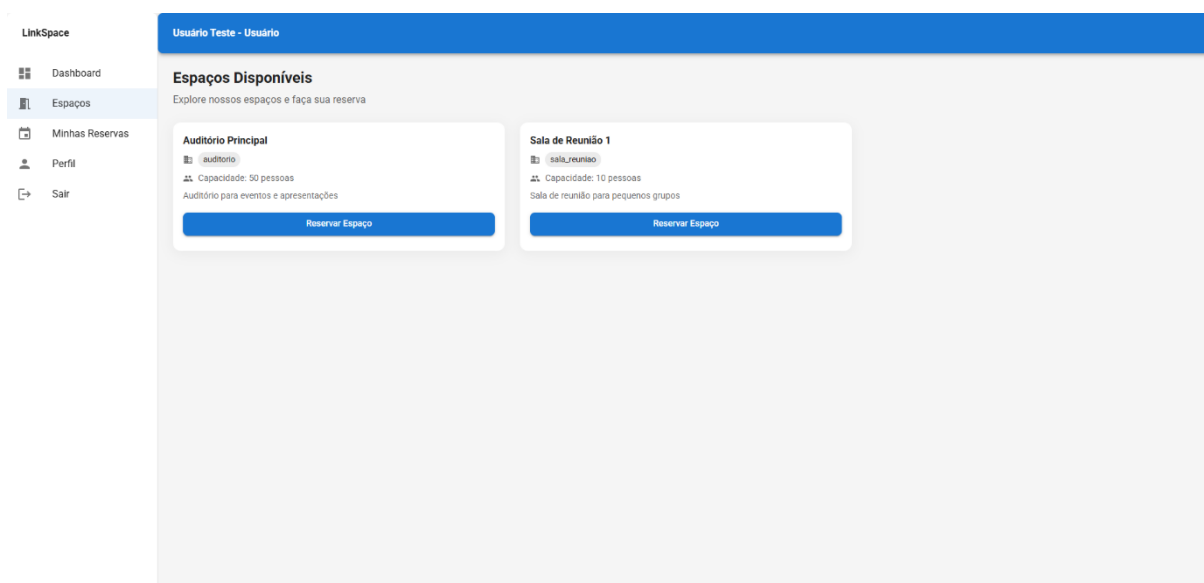
Figura 39 – Página de Espaços (Administrador)



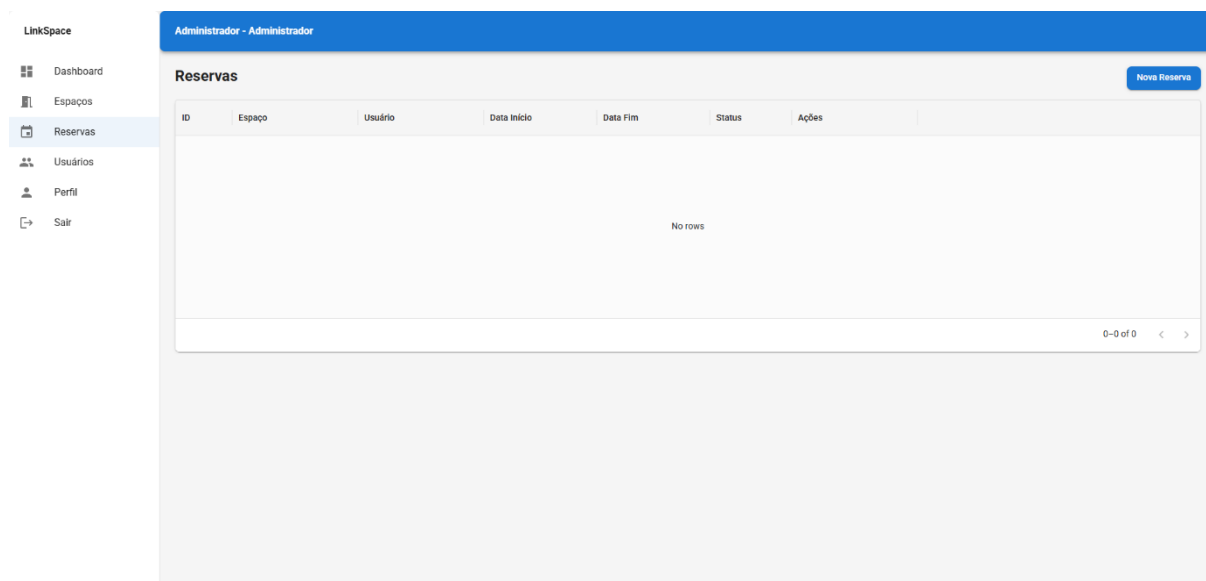
Essa tela possibilita ao administrador cadastrar, editar ou excluir espaços

disponíveis. O uso de um DataGrid torna as informações organizadas em colunas (nome, capacidade, tipo e ações), facilitando o gerenciamento de dados em escala. Além disso, os botões de edição e exclusão tornam a navegação mais prática, permitindo ajustes imediatos. Essa interface garante eficiência operacional ao administrador, sem abrir mão da clareza visual.

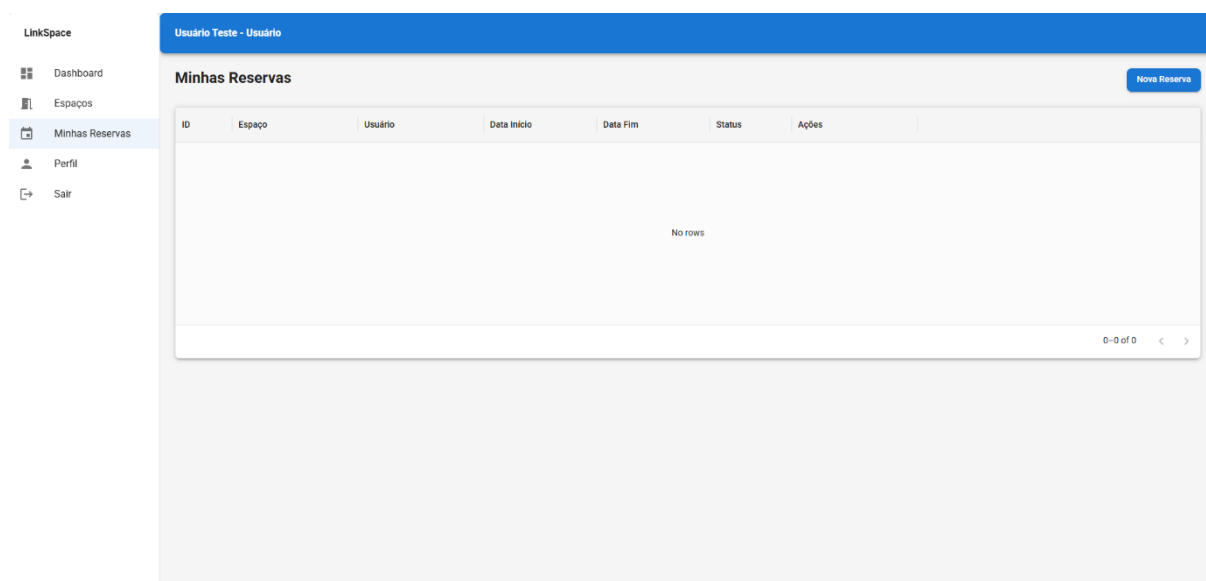
*Figura 40 – Página de Espaços (Usuário Comum)*



Para o usuário comum, a tela de espaços foi desenhada com objetivo de facilitar a escolha do ambiente mais adequado às suas necessidades. Cada espaço é apresentado com informações relevantes, como nome, capacidade e descrição, acompanhadas de um botão de ação que permite realizar a reserva de forma direta. O layout prioriza legibilidade e objetividade, tornando a experiência de reserva simples e rápida.

*Figura 41 – Página de Reservas (Administrador)*

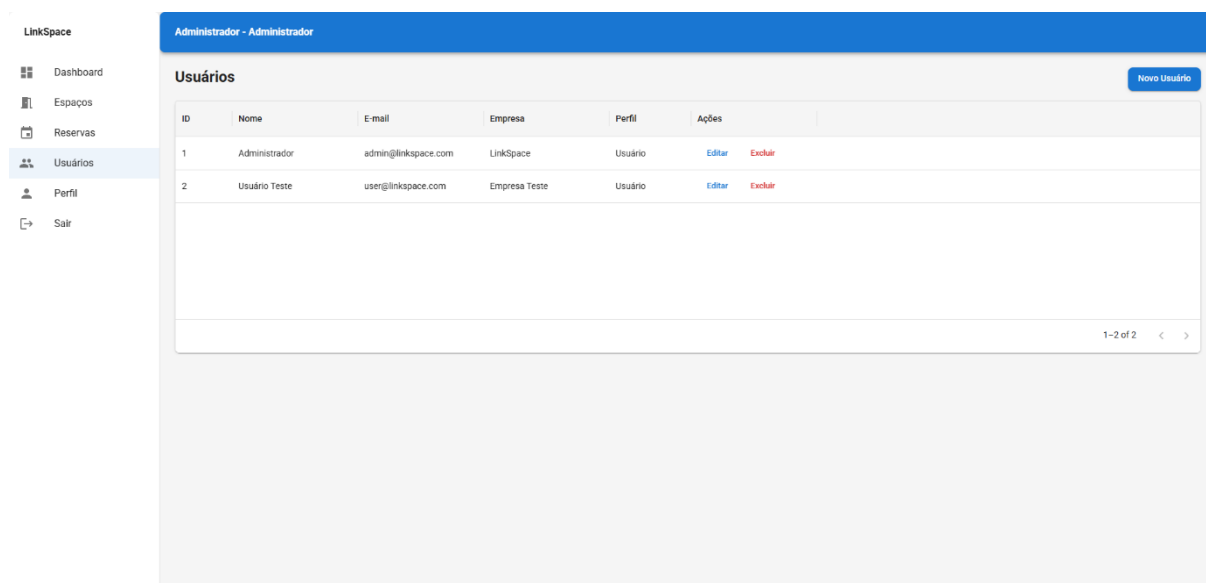
A área de reservas administrativas apresenta uma listagem detalhada das reservas realizadas no sistema. Essa tela permite ao administrador acompanhar a utilização dos espaços, garantindo maior controle sobre o uso dos recursos. A disposição em tabela facilita a identificação rápida de informações como datas, usuários e status das reservas, possibilitando intervenções ou ajustes quando necessário.

*Figura 42 – Página de Reservas (Usuário Comum)*

No perfil de usuário, a tela “Minhas Reservas” exibe todas as reservas associadas à conta logada. A proposta é garantir transparência e autonomia,

permitindo que o próprio usuário acompanhe seus agendamentos, visualize os detalhes e, quando necessário, cancele ou modifique sua reserva. O design segue a mesma consistência visual das demais páginas, mantendo familiaridade no uso.

*Figura 43 – Página de Usuários (Administrador)*



The screenshot displays the 'LinkSpace' user management interface for administrators. On the left is a sidebar with navigation links: Dashboard, Espaços, Reservas, **Usuários** (highlighted), Perfil, and Sair. The main content area has a blue header bar with 'Administrador - Administrador' and a 'Novo Usuário' button. Below this is a table titled 'Usuários' with columns: ID, Nome, E-mail, Empresa, Perfil, and Ações. The table contains two rows of user data. At the bottom right of the table is a pagination indicator '1-2 of 2' with navigation arrows.

ID	Nome	E-mail	Empresa	Perfil	Ações
1	Administrador	admin@linkspace.com	LinkSpace	Usuário	<a href="#">Editar</a> <a href="#">Excluir</a>
2	Usuário Teste	user@linkspace.com	Empresa Teste	Usuário	<a href="#">Editar</a> <a href="#">Excluir</a>

A tela de gestão de usuários oferece ao administrador a possibilidade de visualizar a lista completa de contas cadastradas, com informações como nome, e-mail e empresa vinculada. Além da consulta, é possível realizar edições e exclusões diretamente pela interface, assegurando a manutenção do controle sobre quem acessa a plataforma. Esse recurso é essencial para o gerenciamento seguro do sistema.

*Figura 44 – Página de Configurações (Administrador)*

Nesta tela, o administrador pode atualizar informações pessoais e redefinir credenciais de acesso. A interface foi projetada para oferecer autonomia na administração da conta, dispensando suporte externo em alterações simples. O design prioriza clareza e praticidade, dividindo em blocos os campos relacionados ao perfil e à alteração de senha.

*Figura 45 – Página de Configurações (Usuário Comum)*

A tela de configurações para usuários comuns segue o mesmo padrão visual da versão administrativa, mas adaptada às necessidades individuais. Nela, o usuário

pode atualizar informações de contato e modificar sua senha de forma prática. O objetivo é assegurar personalização da experiência e reforçar a autonomia do usuário dentro do sistema.

De forma geral, o protótipo demonstra uma aplicação bem estruturada, com navegação fluida e usabilidade alinhada às boas práticas de design de interfaces. O layout adotado favorece a interação dos usuários com o sistema, reduz a curva de aprendizado e contribui para o sucesso da aplicação em um ambiente corporativo de gestão e reservas de espaços compartilhados.

## 5 TESTES E RESULTADOS

A etapa de testes é um dos componentes centrais do ciclo de desenvolvimento de software, sendo responsável por avaliar a qualidade da solução implementada e verificar se ela atende, de forma consistente, aos requisitos funcionais e não funcionais previamente estabelecidos. Em projetos acadêmicos como este, os testes assumem dupla relevância: além de garantir que o sistema LinkSpace funciona conforme especificado, também fornecem evidências científicas e objetivas de que as decisões de arquitetura, tecnologias adotadas e critérios de projeto foram adequados para resolver o problema proposto.

Os testes cumprem, portanto, três papéis fundamentais:

**Verificação** – assegurar que cada funcionalidade implementada corresponde corretamente ao que foi especificado (por exemplo, o login de usuário deve autenticar apenas credenciais válidas).

**Validação** – confirmar que o sistema como um todo atende às necessidades reais de seus usuários, simulando cenários práticos de reserva e gestão de espaços.

**Qualidade contínua** – identificar defeitos, gargalos de desempenho ou problemas de interface que possam comprometer a experiência de uso, permitindo ajustes antes da entrega final.

Para estruturar a metodologia de testes, foi adotado o conceito da pirâmide de testes, amplamente referenciado na engenharia de software. Esse modelo sugere que a maior parte dos testes deve estar na base (unitários), com camadas intermediárias de integração e uma quantidade controlada de testes end-to-end (E2E). No caso do LinkSpace, tal abordagem permitiu um equilíbrio entre abrangência (testes em larga escala sobre pequenas funções) e realismo (cenários completos executados em navegador com Cypress).

Outro aspecto importante foi a inclusão de testes não funcionais, realizados por meio de auditorias automáticas (Lighthouse) e análises práticas de usabilidade e acessibilidade. Enquanto os testes funcionais se concentram na corretude de fluxos, os não funcionais avaliam dimensões de qualidade que influenciam diretamente a aceitação e satisfação do usuário final — como tempo de resposta, clareza visual, navegabilidade e segurança.



Além disso, adotou-se uma perspectiva de rastreabilidade entre requisitos e testes: cada caso de uso definido no projeto (por exemplo, UC01 – Login, UC03 – Criar Reserva) foi vinculado a um ou mais casos de teste. Esse alinhamento garantiu que não houvesse lacunas entre o que foi projetado e o que foi efetivamente verificado.

Por fim, cabe ressaltar que os testes foram conduzidos em um ambiente controlado, com dados sintéticos e mocks para simulação do backend. Essa estratégia foi necessária no contexto do TCC, mas não compromete a validade dos resultados, já que os cenários utilizados foram construídos com base em situações reais que o sistema enfrentaria em produção.

Dessa forma, esta seção apresenta a metodologia de testes aplicada ao LinkSpace, detalhando a estratégia adotada (5.1), o ambiente de execução (5.2), os dados e critérios de aceite (5.3), bem como os resultados práticos de testes funcionais, unitários, E2E, de interface e não funcionais (5.4 a 5.7).

## 5.1 Estratégia e escopo

A estratégia de testes do LinkSpace foi definida com base no conceito da pirâmide de testes, priorizando:

**Testes unitários**, responsáveis por validar funções auxiliares, como manipulação de datas e verificação de conflitos de reservas.

**Testes de integração**, aplicados para verificar a comunicação entre a camada de serviços e os componentes da aplicação.

**Testes end-to-end (E2E)**, que simulam a navegação do usuário no navegador, garantindo que os fluxos funcionais críticos se comportem conforme especificado.

**Testes não funcionais**, avaliando desempenho, acessibilidade e usabilidade.

O escopo incluiu tanto as funcionalidades principais — login/registro, gestão de espaços e reservas — quanto aspectos de qualidade mais amplos, como feedback visual da interface, clareza de mensagens de erro e validação de acessibilidade.

Além disso, foram definidos critérios de rastreabilidade: cada caso de teste foi

associado a um requisito funcional (por exemplo, UC01 – Login, UC03 – Criar Reserva), assegurando que não houvesse lacunas entre os objetivos do sistema e a verificação prática de seu funcionamento.

## 5.2 Ambiente de teste

Front-end: React com TypeScript e Material UI (MUI).

Navegador: Chrome (Versão 140.0.7339.128 64 bits) em 1920x1080 e 360x640 (mobile).

Back-end: serviços em memória e mocks para simular respostas da API (cadastro e consulta de espaços/reservas).

Autenticação: baseada em tokens JWT simulados e armazenados no *localStorage* durante os testes.

Para instrumentação, considerou-se Vitest nos testes unitários, Cypress para testes E2E em navegador e Lighthouse para auditorias de desempenho, usabilidade e acessibilidade. Todas as datas foram verificadas no fuso de São Paulo para evitar inconsistências em reservas.

## 5.3 Dados e critérios de aceite

Os dados de teste foram definidos de forma sintética e controlada, contemplando diferentes perfis de usuário e cenários de uso:

**Usuário administrador:** *admin@linkspace.com*

**Usuário comum:** *user@linkspace.com*

**Espaços de teste:** salas fictícias, com capacidades variando de 5 a 100 pessoas.

**Reservas de teste:** horários conflitantes e horários válidos, simulando tanto falhas esperadas (ex.: sobreposição de reservas) quanto operações corretas.

Os critérios de aceite foram estabelecidos com base em quatro dimensões:

1. **Funcionalidade:** cada operação deve retornar a resposta correta da interface e refletir a regra de negócio (ex.: reserva conflitante deve ser rejeitada).
2. **Interface:** os elementos de GUI devem ser exibidos de forma clara, com mensagens compreensíveis e feedback imediato após cada ação.
3. **Desempenho:** operações críticas (login, listagem de espaços, criação

de reservas) devem responder em até 300 ms no percentil 95 (p95).

4. **Segurança básica:** apenas usuários autenticados devem acessar rotas restritas; tokens inválidos ou expirados devem redirecionar para a tela de login.

A adoção desses critérios permitiu uniformizar a análise de resultados, transformando cada execução de teste em uma evidência objetiva de conformidade.

#### **5.4 Testes Funcionais (GUI e API)**

Os testes funcionais foram conduzidos com a ferramenta Cypress, simulando o uso real do sistema no navegador. Foram validados fluxos críticos para os dois perfis de usuário:

- **Administrador:** login, acesso ao menu de espaços, criação de um novo espaço e verificação de sua listagem.
- **Usuário** comum: login, acesso ao menu de reservas, criação de uma reserva e verificação da consistência de datas.

Ao todo, 18 cenários foram automatizados e executados, abrangendo autenticação, navegação, validação de campos obrigatórios e operações de CRUD.

Todos os testes foram concluídos com sucesso, demonstrando que o sistema atende aos critérios de aceite estabelecidos.

Figura 46 – Cypress Teste 01

```
Running: 01_login.cy.ts (1 of 3)
Browserslist: caniuse-lite is outdated. Please run:
  npx update-browserslist-db@latest
  Why you should do it regularly: https://github.com/browserslist/update-db#readme

Login e Navegação
  ✓ deve fazer login como admin e navegar para espaços (1710ms)
  ✓ deve fazer login como usuário e navegar para reservas (1255ms)
  ✓ deve redirecionar admin de /admin para /admin/spaces (1083ms)
  ✓ deve redirecionar usuário de /usuario para /usuario/reservations (1045ms)
  ✓ deve falhar login com credenciais inválidas (721ms)
  ✓ deve fazer logout corretamente (981ms)

6 passing (7s)

(Results)

Tests:      6
Passing:    6
Failing:    0
Pending:    0
Skipped:    0
Screenshots: 0
Video:      false
Duration:   6 seconds
Spec Ran:   01_login.cy.ts
```

A imagem mostra a suíte de testes de autenticação executada com o Cypress. São validados cenários de login do administrador e do usuário comum, incluindo redirecionamentos para as rotas canônicas após sucesso. Também há verificação de mensagens para credenciais inválidas. Todos os casos aparecem aprovados, evidenciando a confiabilidade do fluxo de login.

Figura 47 – Cypress Teste 02

```
Running: 02_admin_spaces_working.cy.ts (2 of 3)

Admin - Gestão de Espaços (Funcionais)
  ✓ deve exibir a lista de espaços existentes (1809ms)
  ✓ deve abrir o diálogo de novo espaço (1066ms)
  ✓ deve criar um novo espaço (7133ms)
  ✓ deve validar campos obrigatórios (1138ms)
  ✓ deve cancelar criação de espaço (4050ms)
  ✓ deve editar um espaço existente (5355ms)

6 passing (21s)

(Results)

Tests:      6
Passing:    6
Failing:    0
Pending:    0
Skipped:    0
Screenshots: 0
Video:      false
Duration:   20 seconds
Spec Ran:   02_admin_spaces_working.cy.ts
```

Esta tela apresenta os testes do perfil administrador para gestão de espaços. O roteiro automatiza a navegação até `/admin/spaces`, abre o formulário, cria um novo espaço e confirma sua exibição no grid. O Cypress valida elementos de interface (título, botões) e o resultado esperado na listagem. Os testes passaram integralmente, confirmando o CRUD essencial do módulo.

Figura 48 – Cypress Teste 03

```
Running: 03_user_reservations_working.cy.ts (3 of 3)

Usuário - Gestão de Reservas (Funcionais)
  ✓ deve exibir a lista de reservas (inicialmente vazia) (1576ms)
  ✓ deve abrir o diálogo de nova reserva (1107ms)
  ✓ deve criar uma nova reserva (8683ms)
  ✓ deve validar campos obrigatórios (1129ms)
  ✓ deve cancelar criação de reserva (4170ms)
  ✓ deve validar que data de fim é posterior à data de início (2550ms)

6 passing (19s)

(Results)

Tests:      6
Passing:    6
Failing:    0
Pending:    0
Skipped:    0
Screenshots: 0
Video:      false
Duration:   19 seconds
Spec Ran:   03_user_reservations_working.cy.ts
```

A imagem exibe a suíte do perfil usuário para minhas reservas. O fluxo cobre abertura da página `/usuario/reservations`, preenchimento dos campos (título/horários) e criação de uma reserva válida, além de checagens de consistência (ex.: impedir datas inválidas). A presença da nova reserva na listagem é verificada ao final. Todos os cenários foram aprovados, demonstrando conformidade com as regras de negócio.

Figura 49 – Cypress Resultado Final

(Run Finished)

Spec		Tests	Passing	Failing	Pending	Skipped
✓ 01_login.cy.ts	00:06	6	6	-	-	-
✓ 02_admin_spaces_working.cy.ts	00:20	6	6	-	-	-
✓ 03_user_reservations_working.cy.ts	00:19	6	6	-	-	-
✓ All specs passed!	00:47	18	18	-	-	-

O painel resume a execução total dos testes E2E no Cypress. Mostra a quantidade de specs e casos concluídos, com 100% de sucesso (todas as suítes em verde). Este resultado consolida que os fluxos críticos — login, gestão de espaços e criação de reservas — estão funcionando ponta a ponta conforme os critérios de aceite

definidos.

## 5.5 Testes unitários e de integração

Além dos testes funcionais, foram desenvolvidos testes unitários e de integração com o framework Vitest. Os testes cobriram funções auxiliares do sistema, como manipulação de datas, verificação de conflitos de horários e operações simuladas de reservas, conforme imagem a seguir.

*Figura 50 – Vitest Testes unitários*

```

✓ src/utils/__tests__/reservations.test.ts (26)
✓ src/utils/__tests__/datetime.test.ts (26)
✓ src/pages/Admin/Spaces.spec.tsx (4) 1182ms

Test Files  3 passed (3)
Tests       56 passed (56)
Start at    21:36:45
Duration    4.43s (transform 171ms, setup 463ms, collect 2.26s, tests 1.22s, environment 1.78s, prepare 428ms)

```

No total, 56 testes unitários/integrados foram executados, todos passando com sucesso, validando a robustez da lógica em camadas inferiores da aplicação.

## 5.6 Testes E2E (navegador)

Os testes end-to-end (E2E) constituem a camada mais próxima da experiência real do usuário. Foram implementados no Cypress, reproduzindo cenários completos de interação com o sistema, desde o login até a realização de operações críticas.

Diferentemente dos testes unitários e de integração, que validam componentes isolados ou funções específicas, os testes E2E validam o fluxo de ponta a ponta, garantindo que todas as camadas da aplicação (interface, lógica de navegação e integração com serviços/mocks) funcionem de forma coesa.

No caso do LinkSpace, foram construídos roteiros que simulam:

- Acesso de um administrador, criação de um espaço e verificação imediata da listagem.
- Acesso de um usuário comum, tentativa de criação de reserva com horários conflitantes (bloqueada pelo sistema) e, posteriormente, criação de uma reserva válida.

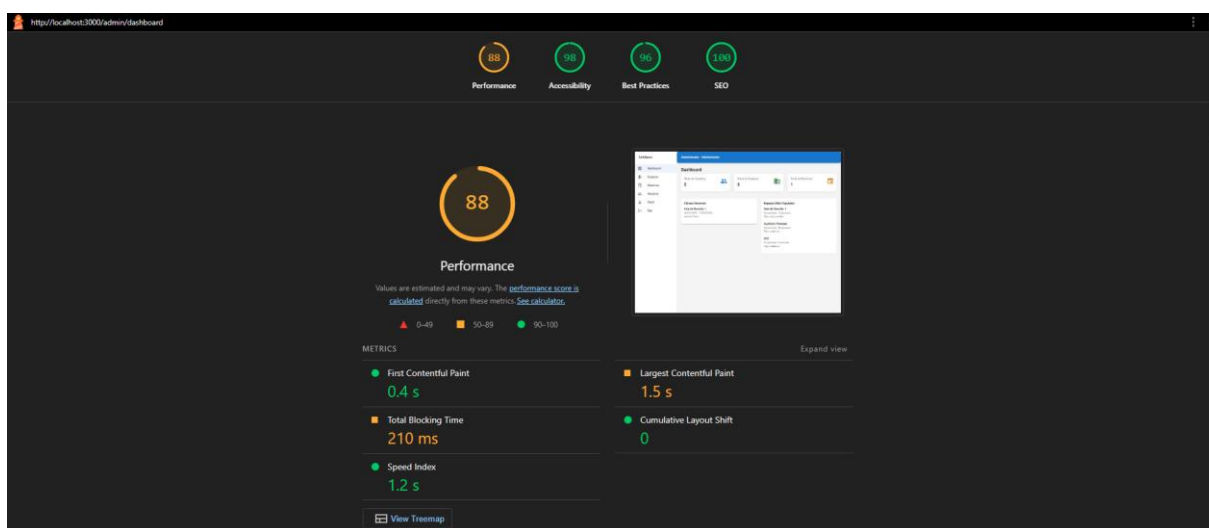
Esses testes permitem comprovar que os fluxos de negócio centrais — reserva de espaço e gestão de recursos — não apenas funcionam isoladamente, mas também se mantêm íntegros quando executados sob as condições reais de uso. O resultado positivo em todos os cenários indica que o sistema cumpre os requisitos de consistência e confiabilidade definidos no escopo do TCC.

## 5.7 Testes de interface e não funcionais

### 5.7.1 Desempenho

Para avaliar desempenho e boas práticas, foi utilizada a ferramenta Lighthouse sobre a página Dashboard (tanto do administrador quanto do usuário comum). Os resultados foram consistentes em ambas as versões, conforme Figura 51.

*Figura 51 – Lighthouse Teste*



As pontuações obtidas foram:

- Performance: 88
- Acessibilidade: 98
- Boas Práticas: 96
- SEO: 100

### 5.7.2 Usabilidade

A usabilidade foi avaliada a partir de dois enfoques:

1. **Teste prático com usuários:** uma pequena amostra (3 participantes,



não técnicos) foi convidada a executar tarefas simples: realizar login, localizar a área correspondente ao seu perfil e criar uma reserva/espço. Todos conseguiram concluir as tarefas em menos de dois minutos, sem solicitar auxílio. Esse resultado indica que a curva de aprendizado do sistema é baixa e que os elementos da interface são suficientemente intuitivos.

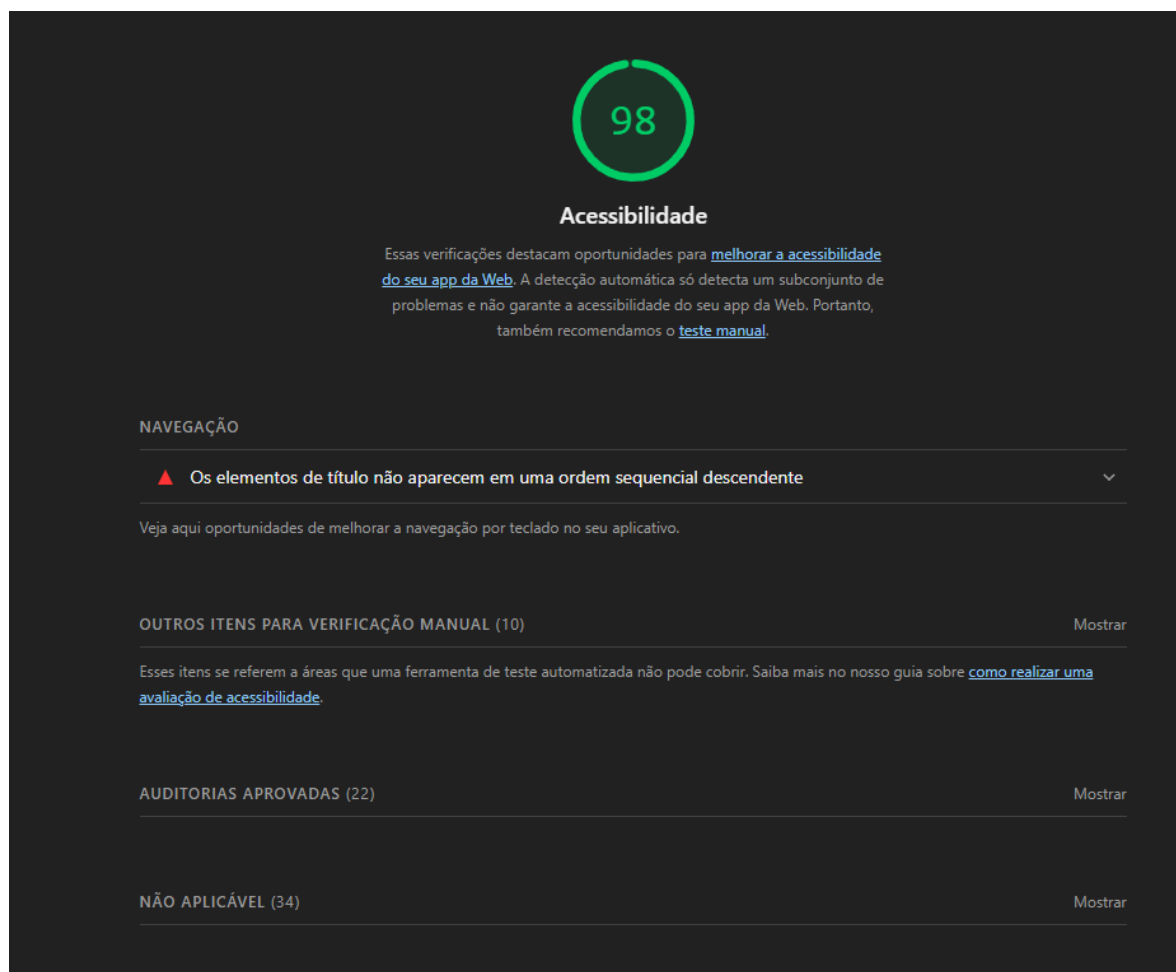
2. **Crterios heursticos:** a interface foi analisada à luz de princípios de Nielsen, como visibilidade do status do sistema (feedbacks claros após ações), correspondência entre o sistema e o mundo real (uso de termos comuns como “Espaços” e “Reservas”) e consistência (mesma estrutura de botões e formulários em diferentes páginas).

Apesar da aprovação geral, um ponto de melhoria identificado foi a ausência de tutoriais ou mensagens explicativas para usuários iniciantes, que poderiam enriquecer a experiência.

### 5.7.3 Acessibilidade

A avaliação de acessibilidade do LinkSpace foi conduzida com o auxílio do Lighthouse, que atribuiu pontuação de 98/100, de acordo com a imagem 10218, resultado considerado satisfatório para os parâmetros atuais de boas práticas.

Figura 52 – Lighthouse Acessibilidade



Observou-se a adoção consistente de elementos semânticos de HTML (por exemplo, `<header>`, `<main>`, `<button>`), o que favorece a interpretação da estrutura por leitores de tela. Também se verificou a presença de rótulos (labels) associados aos campos de formulário, medida essencial para garantir que tecnologias assistivas exponham corretamente o propósito de cada controle. Além disso, a navegabilidade por teclado mostrou-se funcional, permitindo a interação sem dependência exclusiva do mouse.

Apesar do desempenho globalmente positivo, a análise evidenciou oportunidades de aprimoramento. Em primeiro lugar, o contraste de cores de determinados botões não atinge a taxa mínima recomendada pelas Diretrizes de Acessibilidade para Conteúdo Web (WCAG), fixada em 4,5:1 para texto normal, o que pode dificultar a legibilidade em cenários de baixa acuidade visual. Em segundo lugar, identificou-se que algumas imagens de caráter estritamente decorativo não possuem

o atributo `alt=""`, condição que pode introduzir ruído informacional para usuários de leitores de tela ao anunciar conteúdo sem relevância semântica.

Tais ajustes, embora não inviabilizem o uso básico do sistema, são recomendados para ampliar a inclusão e a conformidade do LinkSpace. A correção do contraste cromático e a marcação adequada de imagens decorativas, aliadas à manutenção das práticas já implementadas (semântica HTML, rotulagem de formulários e suporte pleno ao teclado), tendem a consolidar a experiência de uso para diferentes perfis de usuário e a aproximar a aplicação de um padrão de acessibilidade exemplar.

#### 5.7.4 Segurança

Os testes de segurança tiveram caráter exploratório e se concentraram em três aspectos:

**Validação de autenticação:** confirmação de que apenas usuários autenticados acessam rotas restritas. O Cypress comprovou que, ao tentar visitar diretamente `/admin/spaces` ou `/usuario/reservations` sem login, o sistema redireciona para a tela de login.

**Expiração de sessão:** simulação da remoção manual do token de autenticação no `localStorage`, o que resultou em logout imediato na aplicação.

**Tratamento de entradas:** testes básicos de injeção em campos de formulário mostraram que caracteres especiais não quebram o layout nem são interpretados como código executável.

Ainda que essas verificações garantam uma camada inicial de segurança, não foram aplicados testes avançados, como varredura de vulnerabilidades ou pentests automatizados.

#### 5.7.5 Defeitos observados e ajustes

Durante a execução dos testes com o Cypress, foram identificados e corrigidos três pontos relevantes. Primeiro, constatou-se um redirecionamento incorreto após o login: usuários autenticados não eram encaminhados diretamente para suas áreas específicas (administradores para `/admin/spaces` e usuários para `/usuario/reservations`). A revisão da configuração de rotas solucionou a inconsistência.

Em seguida, verificou-se que as mensagens exibidas em tentativas de login com credenciais inválidas eram excessivamente genéricas (“Invalid credentials”). O conteúdo foi reformulado para tornar o feedback mais claro e acolhedor, favorecendo a compreensão do erro e a continuidade da tarefa. Por fim, observou-se uma inconsistência nas labels dos botões: alguns exibiam apenas “Salvar”, sem indicar o contexto da ação. O texto foi ajustado para formas específicas, como “Salvar Espaço” e “Salvar Reserva”, aprimorando a clareza e reduzindo ambiguidades.

#### 5.7.6 Limitações

As principais limitações observadas na fase de testes concentraram-se em quatro dimensões.

Em primeiro lugar, o uso de mocks foi necessário diante da ausência de um backend definitivo; como parte dos testes — sobretudo unitários e de integração — operou com dados em memória ou simulados, a validação ficou menos abrangente por não contemplar cenários reais de persistência em banco de dados.

Em segundo lugar, a amostra destinada à avaliação de usabilidade foi reduzida: apenas três usuários participaram da validação prática, o que limita a possibilidade de generalização dos resultados.

Em terceiro lugar, não foram conduzidos testes de carga; a inexistência de simulações de múltiplos acessos simultâneos impedem atestar o comportamento do sistema sob condições de alto tráfego.

Por fim, a cobertura de segurança foi parcial, restringindo-se a verificações básicas — como login, expiração de sessão e validação de entradas simples — e deixando de fora hipóteses de ataque mais sofisticadas. Apesar dessas limitações, os testes realizados foram suficientes para comprovar a viabilidade do sistema e para validar seus principais fluxos de uso.

#### 5.7.7 Trabalhos futuros

**Observabilidade e segurança:** rate limiting por IP, headers via Helmet, métricas e tracing para diagnóstico.

**Carga distribuída:** testes de estresse em ambiente com múltiplas instâncias e banco remoto para aferir escalabilidade.

**Experiência do usuário:** calendário/datepicker nas reservas, aprimoramento do foco visível e rótulos em ações secundárias.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

ABRAMOV, Dan; CLARK, Ryan Florence. React Router: Declarative Routing for React. Documentação Oficial, 2022. Disponível em: <https://reactrouter.com/en/main>. Acesso em: 09 maio 2025.

ALAVI, M.; LEIDNER, D. E. Review: Knowledge management and knowledge management systems: conceptual foundations and research issues. *MIS Quarterly*, v. 25, n. 1, p. 107–136, 2001.

ALMEIDA, João; SANTOS, Carla. Gestão de espaços em ambientes colaborativos: estudo de caso em coworkings brasileiros. *Revista de Inovação em Gestão*, v. 10, n. 2, p. 45–60, 2021.

ASRAR-UL-HAQ, M.; ANWAR, S. A systematic review of knowledge management and knowledge sharing: Trends, issues, and challenges. *Cogent Business & Management*, v. 3, n. 1, 2016.

AWS. Diferença entre modelo de dados lógico e físico. Disponível em: <https://aws.amazon.com/pt/compare/the-difference-between-logical-and-physical-data-model/>. Acesso em: 20 abr. 2025.

BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. *Software Architecture in Practice*. 3. ed. Boston: Addison-Wesley, 2013.

BENCHIMOL, Rafael. Espaços compartilhados e inovação urbana: repensando o uso da infraestrutura nas cidades inteligentes. *Revista Brasileira de Urbanismo*, v. 5, n. 1, p. 40–55, 2019.

BIBLIODEV. *Manual de Desenvolvimento Web com Django 4 e Vue.js: do Backend à Interface*. São Paulo: Casa do Código, 2022.

BOUCKEN, C.; REUSCHL, A. Knowledge sharing in coworking spaces: Practices, challenges and perspectives. *Journal of Business Research*, v. 103, p. 405–415, 2018.

BOUCKEN, F.; REUSCHL, A. Co-working spaces: A growing trend fostering entrepreneurial communities. *Urban Studies*, v. 55, n. 9, p. 1982–1999, 2018.

BRANDÃO, Rafael; REZENDE, Marcelo. *Segurança em aplicações web: fundamentos e práticas*. São Paulo: Novatec, 2021.

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). *Diário Oficial da União*, Brasília, DF, 15 ago. 2018. Disponível em: <[https://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2018/lei/l13709.htm](https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm)>. Acesso em: 20 abr. 2025.

BRASIL. Ministério da Economia. *Transformação digital nos serviços públicos: diretrizes para modernização da gestão interna*. Brasília: SEGES, 2020.

BROWN, Tim; WYATT, Jocelyn. Design thinking for social innovation. *Stanford Social Innovation Review*, v. 8, n. 1, p. 30–35, Winter 2010.

CAMPOS, Alyson Wegilles de Souza. *O uso do design responsivo no desenvolvimento de websites: aplicação dos métodos na construção de um website*. 2017. 61 f. Trabalho de Conclusão de Curso (Graduação em Design) – Universidade Federal de Pernambuco, Caruaru, 2017.

CASTELLS, Manuel. *A sociedade em rede*. 3. ed. São Paulo: Paz e Terra, 1999.

CGI.br – COMITÊ GESTOR DA INTERNET NO BRASIL. *Pesquisa TIC Domicílios 2023: uso das tecnologias de informação e comunicação nos domicílios brasileiros*. São Paulo: NIC.br, 2024.

CISNE, A. C. M.; ARASAKI, A. R.; SANTOS, J. C. Gestão do conhecimento: uma proposta de integração da inteligência competitiva. *Revista Gestão & Tecnologia*, v. 15, n. 2, p. 50–73, 2015.

COSTA, Mariana; FERREIRA, Tiago. Ambientes corporativos e reservas inteligentes: práticas de TI aplicadas à gestão de espaços. Caderno de Administração, v. 23, n. 1, p. 77–92, 2019.

COWORKING BRAZIL. Censo Coworking Brasil 2023. Disponível em: <https://coworkingbrasil.org/>. Acesso em: 05 maio 2025.

COWORKING RESOURCES. Key figures: coworking growth. 2023. Disponível em: <https://www.coworkingresources.org/blog/key-figures-coworking-growth>. Acesso em: 9 maio 2025.

COWORKING RESOURCES. The History of Coworking [Figura 2]. Disponível em: <https://www.coworkingresources.org/blog/history-of-coworking>. Acesso em: 23 abr. 2025.

DATE, C. J. Introdução a sistemas de banco de dados. 8. ed. Rio de Janeiro: Campus, 2004.

DATE-FNS. Modern JavaScript Date Utility Library. Documentação Oficial, 2024. Disponível em: <https://date-fns.org>. Acesso em: 09 maio 2025.

DAVENPORT, T. H.; PRUSAK, L. Conhecimento empresarial: como as organizações gerenciam o seu capital intelectual. 1. ed. Rio de Janeiro: Campus, 1998.

DESIGN COUNCIL (UK). The Double Diamond: a universally accepted depiction of the design process. London: Design Council, 2015. Disponível em: <https://www.designcouncil.org.uk>. Acesso em: 19 abr. 2025.

FERREIRA, Vinícius; ARAÚJO, Daniela. Segurança da informação em sistemas web: autenticação, autorização e criptografia. Revista de Tecnologia Aplicada, v. 17, n. 2, p. 35–48, 2022.

FERREIRA, Thayane Tamires de Souza; RAMOS, Flávia; TEIXEIRA, Jaqueline Nogueira. Tecnologias digitais no cuidado em saúde: desafios e possibilidades para a



qualificação do trabalho em enfermagem. *Revista Brasileira de Enfermagem*, v. 74, supl. 1, p. e20201385, 2021.

FLANAGAN, David. *JavaScript: o guia definitivo*. 7. ed. São Paulo: O'Reilly Media Brasil, 2021.

FREEMAN, Eric. *Use a cabeça! HTML e CSS*. 2. ed. São Paulo: Alta Books, 2020.

FREEMAN, Eric; ROBSON, Elisabeth. *Use a cabeça! JavaScript: uma introdução à programação orientada a objetos*. 2. ed. São Paulo: Alta Books, 2021.

G2; CAPTERRA; TRUSTRADIUS. *Análises comparativas de sistemas de reservas*. Acesso em: 20 abr. 2025.

GANDINI, A. The rise of coworking spaces: A literature review on the emerging phenomenon. *Geography Compass*, v. 9, n. 8, p. 512–521, 2015.

GARRETT, Jesse James. *The Elements of User Experience: User-Centered Design for the Web and Beyond*. Berkeley: New Riders, 2011.

GOOGLE. *Google Workspace Help: funcionalidades do Google Calendar*. Disponível em: <https://support.google.com/calendar>. Acesso em: 20 abr. 2025.

GRANT, R. M. Toward a knowledge-based theory of the firm. *Strategic Management Journal*, v. 17, p. 109–122, 2013.

GRINBERG, Miguel. *Desenvolvimento Web com Flask e Python: crie aplicações web usando Flask, PostgreSQL e Docker*. São Paulo: Novatec, 2019.

HSU, M. H.; CHANG, C. M. Examining interpersonal trust as a facilitator and uncertainty as an inhibitor of intra-organisational knowledge sharing. *Information Systems Journal*, v. 24, n. 2, p. 119–142, 2014.

HU, M. L. M.; LIN, C. Y. Y.; CHANG, C. H. Knowledge sharing and intellectual capital in the hotel industry. *International Journal of Hospitality Management*, v. 32, p. 78–86, 2013.

HUANG, Q.; DAVISON, R. M.; GU, J. The impact of trust, guanxi orientation and face on the intention to engage in interpersonal knowledge sharing in China. *Information Systems Journal*, v. 21, n. 6, p. 557–577, 2011.

IEEE Standard 829-2008 (IEEE Standard for Software Test Documentation).

IGLESIAS-PRADAS, Santiago; HERNÁNDEZ-GARCÍA, Ángel; FERNÁNDEZ-CARDADOR, Andrés. Social network sites in the workplace: A case study of a “Big Four” accounting firm. *Information Systems Management*, v. 34, n. 3, p. 268–283, 2017.

IM, G.; RAI, A. Knowledge sharing ambidexterity in long-term interorganizational relationships. *Management Science*, v. 54, n. 7, p. 1281–1296, 2008.

ISTQB Foundation Level Syllabus. International Software Testing Qualifications Board, 2018.

JULIAN, Mario de Oliveira et al. Design thinking: uma abordagem centrada no ser humano para a inovação. *Revista de Empreendedorismo, Inovação e Tecnologia*, v. 1, n. 1, p. 58–77, 2016.

KRUG, Steve. *Don't make me think: A common sense approach to web usability*. 2. ed. Berkeley: New Riders, 2005.

KRUG, Steve. *Não me faça pensar: uma abordagem de bom senso à usabilidade na web*. 3. ed. Rio de Janeiro: Alta Books, 2020.

LOBO, Renan. Comparativo entre Moment.js e date-fns: qual biblioteca de data escolher em 2023. Medium, 2023. Disponível em: <https://renanlobo.dev/date-fns-vs-momentjs>. Acesso em: 09 maio 2025.

LOEBBECKE, C.; VAN FENEMA, P. C.; POWELL, P. Enterprise platform logic: A study of mediating and constraining factors in the provision of platform-based solutions. *Journal of Strategic Information Systems*, v. 25, n. 1, p. 29–45, 2016.

MAIA, Camila do Couto. Criação de um website sobre tecnologias da informação e comunicação na área da saúde. 2023. 193 f. Dissertação (Mestrado Profissional em Gestão do Cuidado em Enfermagem) – Universidade Federal de Santa Catarina, Florianópolis, 2023.

MAIA, João. Gestão informatizada de TICs na saúde pública: um estudo de caso. São Paulo: Editora Tecnologia Pública, 2023.

MARCOTTE, Ethan. *Responsive Web Design*. New York: A Book Apart, 2014.

MARTINEZ-CONESA, I.; SOTO-ACOSTA, P.; CARAYANNIS, E. G. On the path towards open innovation: Assessing the role of knowledge management capability and environmental dynamism in SMEs. *Journal of Knowledge Management*, v. 21, n. 3, p. 553–570, 2017.

MAZZA, Felipe. *Web Security Essentials: Guia prático para desenvolvedores*. São Paulo: Casa do Código, 2020.

MICROSOFT. Documentação ASP.NET Core MVC. Disponível em: <https://learn.microsoft.com/pt-br/aspnet/core/mvc/overview?view=aspnetcore-9.0>. Acesso em: 20 abr. 2025.

MICROSOFT. Microsoft Bookings Documentation. Disponível em: <https://learn.microsoft.com/en-us/microsoft-365/bookings/>. Acesso em: 20 abr. 2025.

MORISSET, Bruno. Building new places of the creative economy: the rise of coworking spaces. In: *Proceedings of the 2nd Geography of Innovation International Conference*, Utrecht University, 2014. p. 1–23.

MOROZOV, Evgeny; BRIA, Francesca. *Rethinking the Smart City: Democratizing Urban Technology*. Rosa Luxemburg Stiftung, 2018.

MUI. Material UI – Responsive Design. Documentação Oficial, 2024. Disponível em: <https://mui.com/material-ui/customization/breakpoints/>. Acesso em: 09 maio 2025.

NIELSEN, Jakob. Usability Engineering. Boston: Academic Press, 1994.

NIELSEN, Jakob. Usability Engineering. San Diego: Morgan Kaufmann, 2021.

NORMAN, Donald A. O design do dia a dia. Rio de Janeiro: Rocco, 2013.

OECD – ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT. Digital Government Review of Brazil: towards the digital transformation of the public sector. Paris: OECD Publishing, 2018.

OLIVEIRA, Ana Paula; FONSECA, João Marcos. Ambientes colaborativos: a construção dos espaços compartilhados no século XXI. Revista Cidades e Territórios, v. 12, n. 2, p. 77–91, 2020.

OWASP. OWASP Top 10 – 2021: The Ten Most Critical Web Application Security Risks. Open Web Application Security Project, 2021. Disponível em: <https://owasp.org/Top10/>. Acesso em: 09 maio 2025.

PICCINI, Robson Sotoriva; BEHR, Ariel; SCHIAVI, Giovana Sordi. Compartilhamento de conhecimento entre empresas e suas consequências: um estudo em espaços de coworking. Ciência da Informação, Brasília, v. 53, n. 1, p. 1–28, jan./abr. 2024.

PRESSMAN, Roger S.; MAXIM, Bruce R. Engenharia de software: uma abordagem profissional. 9. ed. Porto Alegre: AMGH Editora, 2021.

ROOMZILLA. Roomzilla Overview. Disponível em: <https://www.roomzilla.net/>. Acesso em: 20 abr. 2025.

SANTOS, S. H. Desenvolvimento de uma plataforma web para monitoramento das mídias sociais das empresas juniores da Universidade Federal de Ouro Preto. João Monlevade: UFOP, 2019.

SCHNEIER, Bruce. *Secrets and lies: digital security in a networked world*. New York: John Wiley & Sons, 2000.

SILVA, Roberta; PEREIRA, Lucas. Arquiteturas em nuvem para sistemas de gestão institucional. *Revista Brasileira de Computação Aplicada*, v. 12, n. 1, p. 32–49, 2020.

SIMEONOVA, Boyka. Digital technologies and knowledge sharing: the case of a big four firm. *Journal of Knowledge Management*, v. 22, n. 1, p. 133–155, 2018.

SKEDDA. Skedda Features. Disponível em: <https://www.skedda.com/features>. Acesso em: 20 abr. 2025.

SOMMERVILLE, Ian. *Engenharia de software*. 9. ed. São Paulo: Pearson Education do Brasil, 2011.

SOMMERVILLE, Ian. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2019.

SOTO-ACOSTA, Pedro; CEGARRA-NAVARRO, Juan Gabriel; GARCIA-PEREZ, Alexeis. Analyzing the role of knowledge-intensive services in the management of innovation. *Journal of Business Research*, v. 70, p. 178–187, 2017.

SOUZA, Ricardo; RODRIGUES, Letícia. Tecnologia da informação no ensino superior: sistemas de apoio à gestão acadêmica. *Revista de Tecnologias Educacionais*, v. 7, n. 3, p. 98–112, 2020.

STEFFEN, Bruno; OLIVEIRA, E. G.; BALLE, A. R. Conhecimento compartilhado em ambientes colaborativos: fatores que influenciam e mecanismos de gestão. *Revista de Administração da UFSM*, v. 10, n. 2, p. 316–336, 2017.

TANENBAUM, Andrew S.; WETHERALL, David J. Redes de computadores. 5. ed. São Paulo: Pearson, 2011.

TIDWELL, Jenifer. Designing Interfaces: Patterns for Effective Interaction Design. 2. ed. Sebastopol: O'Reilly Media, 2011.

W3C – WORLD WIDE WEB CONSORTIUM. Media Queries Level 4. 2018. Disponível em: <https://www.w3.org/TR/mediaqueries-4/>. Acesso em: 09 maio 2025.

W3C – WORLD WIDE WEB CONSORTIUM. Web Content Accessibility Guidelines (WCAG) 2.1. 2018. Disponível em: <https://www.w3.org/TR/WCAG21/>. Acesso em: 09 maio 2025.

ZIMMERMANN, R. et al. O compartilhamento do conhecimento e a inovação organizacional: uma revisão sistemática da literatura. Revista de Administração Mackenzie, v. 19, n. 2, p. 1–34, 2018.

## **7 APÊNDICE (CASO NECESSÁRIO)**