

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

**BOOK COLLECTION: UMA REDE SOCIAL DE GERENCIAMENTO E
ORGANIZAÇÃO DE BIBLIOTECAS PESSOAIS**

DOUGLAS DA SILVA ARAÚJO

**Cajazeiras
2025**

**BOOK COLLECTION: UMA REDE SOCIAL DE GERENCIAMENTO E
ORGANIZAÇÃO DE BIBLIOTECAS PESSOAIS**

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

Orientador

Prof. PhD. Francisco Daladier Marques Júnior.

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

A663b

Araújo, Douglas da Silva.

Book collection : uma rede social de gerenciamento e organização de bibliotecas pessoais / Douglas da Silva Araújo. – Cajazeiras, 2025.
111f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2025.

Orientador: Prof. Dr. Francisco Daladier Marques Júnior.

1. Desenvolvimento de sistemas. 2. Rede social Book Collection.
3. Gestão de acervo. 4. Estante virtual. I. Instituto Federal da Paraíba.
II. Título.

IFPB/CZ

CDU: 004.4:028(043.2)



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

DOUGLAS DA SILVA ARAUJO

**BOOK COLLECTION: UMA REDE SOCIAL DE GERENCIAMENTO E ORGANIZAÇÃO DE
BIBLIOTECAS PESSOAIS**

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

Orientador

Prof. Dr. Francisco Daladier Marques Júnior

Aprovada em: **19 de Março de 2025.**

Prof. Dr. Francisco Daladier Marques Júnior - Orientador

Prof. Me. Francisco Paulo de Freitas Neto - Avaliador
IFPB - Campus Cajazeiras

Prof. Me. Diogo Dantas Moreira - Avaliador IFPB -
Campus Cajazeira

Documento assinado eletronicamente por:

- **Francisco Paulo de Freitas Neto**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 19/03/2025 10:53:03.
- **Francisco Daladier Marques Junior**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 20/03/2025 22:03:38.
- **Diogo Dantas Moreira**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 24/03/2025 19:36:43.

Este documento foi emitido pelo SUAP em 19/03/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 684330
Verificador: 17d3067332
Código de Autenticação:



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

Dedico este trabalho à minha família — meu pai, minha mãe e minhas irmãs — que sempre me apoiaram incondicionalmente ao longo de toda a minha trajetória acadêmica.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela força, saúde e sabedoria que me conduziram até a conclusão desta etapa tão importante da minha vida acadêmica. Sem Ele, nada disso seria possível.

Aos meus pais e irmãs, meu eterno agradecimento por todo o apoio, carinho e incentivo incondicionais ao longo dessa jornada. Vocês estiveram ao meu lado em cada desafio, e sem o suporte emocional de vocês, eu não teria conseguido superar os obstáculos e chegar até aqui.

Agradeço também ao meu orientador, Francisco Daladier Júnior, por sua dedicação, paciência e por acreditar na minha ideia. Seu apoio foi fundamental no desenvolvimento deste trabalho, e as valiosas orientações e sugestões foram essenciais para aprimorar este projeto.

Aos meus colegas de curso, agradeço pelas trocas de ideias, discussões enriquecedoras e pelo apoio em momentos de dúvida e desafio. Vocês foram uma importante fonte de motivação durante toda essa caminhada.

Aos professores, meu sincero agradecimento pelos ensinamentos que transcenderam as salas de aula e contribuíram significativamente para a construção deste trabalho. As conversas e orientações que tive com vocês foram decisivas para refinar as especificações do projeto e encontrar soluções para dúvidas enfrentadas.

Agradeço também ao IFPB, por oferecer um ambiente de aprendizado acolhedor e todas as oportunidades que me permitiram crescer tanto academicamente quanto pessoalmente.

Por fim, um agradecimento especial aos participantes do questionário público realizado durante a pesquisa. Suas contribuições foram valiosas para o desenvolvimento e compreensão das questões abordadas neste trabalho.

A todos, meu mais profundo agradecimento.

”Observem como os lírios crescem: eles não fiam, nem tecem. Porém, eu digo a vocês que nem mesmo o rei Salomão, em toda a sua glória, jamais se vestiu como um deles. Se Deus veste assim a erva do campo, que hoje existe e amanhã é queimada no forno, quanto mais ele fará por vocês, gente de pouca fé!”

Lucas 12:27-28

RESUMO

Em um mundo cada vez mais imerso na tecnologia, a literatura continua a desempenhar um papel crucial no desenvolvimento humano, intelectual e cultural. Embora as tecnologias modernas influenciam a maneira como a sociedade se relaciona com a leitura, elas também oferecem novas oportunidades para enriquecer essa interação. Neste contexto, o presente trabalho apresenta o sistema *Book Collection*, projetado para integrar o ambiente digital com o universo literário. O *Book Collection* tem como objetivo principal fornecer uma plataforma que permite aos usuários organizar e explorar novas leituras e histórias. Com funcionalidades voltadas para o gerenciamento de estantes virtuais, o sistema facilita a criação e manutenção de coleções de livros, adaptando-se aos interesses individuais dos usuários. Além disso, oferece uma camada social interativa, onde os usuários podem compartilhar publicações, enviar mensagens e acessar as coleções de outros membros, promovendo uma rica troca de informações e experiências literárias. Para tanto, em sua versão inicial, o sistema será desenvolvido para a Web, utilizando tecnologias amplamente adotadas no mundo moderno, como *Node.js*, *TypeScript*, *PostgreSQL*, *React JS*, *Google Books* e *Gemini AI*. O sistema não só proporciona uma estrutura para quem já é adepto da leitura, mas também atua como um incentivo para aqueles que ainda não estão familiarizados com o mundo dos livros. Através de uma interface intuitiva e recursos personalizados, o *Book Collection* busca engajar e expandir o interesse literário, oferecendo uma base de dados abrangente e ferramentas de personalização para enriquecer a experiência do usuário. Com uma abordagem inovadora, o *Book Collection* visa transformar a maneira como os leitores interagem com suas coleções e com a comunidade literária, tornando a leitura mais acessível e atraente no ambiente digital.

Palavras-chave: Tecnologia. Literatura. *Book Collection*. Sistema Web. Estantes virtuais. Gerenciamento de leituras. Camada social. Interação literária. Personalização.

ABSTRACT

In a world increasingly immersed in technology, literature continues to play a crucial role in human, intellectual and cultural development. While modern technologies influence the way society relates to reading, they also offer new opportunities to enrich this interaction. In this context, this paper presents the Book Collection system, designed to integrate the digital environment with the literary universe. Book Collection's main objective is to provide a platform that allows users to organize and explore new readings and stories. With functionalities aimed at managing virtual bookshelves, the system facilitates the creation and maintenance of book collections, adapting to users' individual interests. It also offers an interactive social layer, where users can share publications, send messages and access other members' collections, promoting a rich exchange of information and literary experiences. To this end, in its initial version, the system will be developed for the Web, using technologies widely adopted in the modern world, such as Node.js, TypeScript, PostgreSQL, React JS, Google Books and Gemini AI. The system not only provides a structure for those who are already adept at reading, but also acts as an incentive for those who are not yet familiar with the world of books. Through an intuitive interface and personalized features, Book Collection seeks to engage and expand literary interest, offering a comprehensive database and personalization tools to enrich the user experience. With an innovative approach, Book Collection aims to transform the way readers interact with their collections and with the literary community, making reading more accessible and attractive in the digital environment.

Keywords: Technology. Literature. Book Collection. Web system. Virtual bookshelves. Reading management. Social layer. Literary interaction. Personalization

LISTA DE FIGURAS

Figura 01 - Produtos do Backlog do sistema.....	29
Figura 02 - Diagrama de Fluxo.....	35
Figura 03 - Diagrama de Caso de Uso.....	55
Figura 04 - Diagrama de atividade.....	62
Figura 05 - Modelagem Conceitual dos Dados.....	63
Figura 06 - Modelagem Lógica dos Dados.....	65
Figura 07 - Diagrama de Camadas da Arquitetura.....	67
Figura 08 - Diagrama de Camadas da Camada de Aplicação.....	67
Figura 09 - Arquitetura de aplicações Web usando o padrão MVC.....	68
Figura 10 - Funcionamento da Arquitetura REST.....	69
Figura 11 - Correlação entre as arquiteturas do sistema.....	72
Figura 12 - Tela de Login.....	85
Figura 13 - Tela de Redefinição de Senha.....	86
Figura 14 - Tela Cadastro.....	87
Figura 15 - Tela Inicial.....	88
Figura 16 - Tela Inicial para novos Usuários.....	89
Figura 17 - Pesquisa de Interesses de Leitura.....	89
Figura 18 - Tela de Estante Virtual.....	90
Figura 19 - Tela de Coleção de Livros.....	91
Figura 20 - Tela de Perfil.....	92
Figura 21 - Publicação de usuário: Avaliações.....	93
Figura 22 - Publicação de usuário: Citações.....	93
Figura 23 - Publicação de usuário: Livros lidos.....	94
Figura 24 - Tela de edição de perfil.....	94
Figura 25 - Tela de Perfil de um Amigo.....	95
Figura 26 - Tela de Pesquisa de Livros.....	96
Figura 27 - Tela de Informações de um Livro.....	97
Figura 28 - Seleção de Coleção.....	98
Figura 29 - Funcionalidades de livros não presentes na coleção.....	99
Figura 30 - Funcionalidades de livros presentes na coleção.....	99
Figura 31 - Avaliar Livro.....	100
Figura 32 - Tela Citações.....	101
Figura 33 - Tela de Histórico de Gerenciamento de Leitura.....	102
Figura 34 - Janela Virtual: Diário de Leitura em andamento.....	103
Figura 35 - Janela Virtual: Visualização pública de um Diário de Leitura.....	103
Figura 36 - Janela Virtual: Atualização de Diário de Leitura.....	104
Figura 37 - Tela de Notificações.....	105
Figura 38 - Tela Inicial do Chapter.....	106
Figura 39 - Tela Conversa com um Amigo.....	106

LISTA DE ALGORITMOS

Algoritmo 01 - Exemplo de Teste Unitário.....	33
Algoritmo 02 - Consumo do Google Books API.....	79
Algoritmo 03 - Configuração do Gemini AI.....	82
Algoritmo 04 - Geração de Prompts.....	83

LISTA DE QUADROS

Quadro 01 - Análise das características dos sistemas analisados.....	24
Quadro 02 - Análise de pesquisa pública.....	27
Quadro 03 - Exemplo de Caso de Teste.....	32
Quadro 04 - Fluxo de execução e desenvolvimento.....	37
Quadro 05 - Requisitos Funcionais.....	42
Quadro 06 - Requisitos não funcionais.....	44
Quadro 07 - Caso de uso: Gerência de estante virtual.....	56
Quadro 08 - Caso de uso: Gerência de leitura.....	57
Quadro 09 - Caso de uso: Gerência de publicação.....	59
Quadro 10 - Caso de uso: Busca de livros.....	60

LISTA DE ABREVIATURAS E SIGLAS

ACID	Atomicidade, Consistência, Isolamento e Durabilidade
ADS	Análise e Desenvolvimento de Sistemas
API	<i>Application Programming Interface</i>
CRUD	<i>Create, Read, Update, Delete</i>
EDA	<i>Event-Driven Architecture</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IA	Inteligência Artificial
ISBN	<i>International Standard Book Number</i>
IFPB	Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
JWT	<i>JSON Web Token</i>
LLM	<i>Large Language Models</i>
MVC	<i>Model-View-Controller</i>
ORM	<i>Object-Relational Mapping</i>
PSP	<i>Personal Software Process</i>
REST	<i>Representational State Transfer</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	<i>Structured Query Language</i>
TCC	Trabalho de Conclusão de Curso
TDD	<i>Test-Driven Development</i>
UML	<i>Unified Modeling Language</i>
URI	<i>Uniform Resource Identifier</i>
UX	<i>User Experience</i>
UI	<i>User Interface</i>
Virtual DOM	<i>Document Object Model Virtual</i>
XP	<i>Extreme Programming</i>

Sumário

1 INTRODUÇÃO.....	14
1.1 Motivação.....	16
1.2 Problemática.....	16
1.3 Potencialidade de Comercialização.....	18
1.4 Objetivos.....	19
1.4.1 Objetivo geral.....	19
1.4.2 Objetivos específicos.....	20
1.5 Organização do Documento.....	20
2 METODOLOGIA.....	21
2.1 Organização do Planejamento de Desenvolvimento.....	21
2.1.1 Pesquisa de trabalho e Projetos relacionados.....	21
2.1.2 Análise de pesquisa pública.....	26
2.1.3 Organização dos requisitos.....	28
2.2 Desenvolvimento ágil e Fluxo de execução das atividades.....	30
2.2.1 Scrum Solo.....	30
2.2.2 Test-driven Development (TDD).....	31
2.2.3 Fluxo de Desenvolvimento.....	34
2.2.4 Fluxo de realização das atividades.....	35
2.3 Conclusão Metodológica.....	40
3 DESCRIÇÃO DO SISTEMA.....	41
3.1 Elicitação dos Requisitos.....	41
3.2 Tecnologias Utilizadas.....	45
3.2.1 React JS.....	45
3.2.2 Typescript.....	46
3.2.3 Prisma ORM.....	46
3.2.4 Express JS.....	47
3.2.5 Socket.io.....	48
3.2.6 Google Books API.....	49
3.2.7 Gemini AI.....	50
3.2.8 PostgreSQL.....	51
3.2.9 Redis.....	52
3.3 Modelagem.....	53
3.3.1 Modelagem do sistema.....	54
3.3.1.1 Diagrama de Caso de Uso.....	54
3.3.1.2 Caso de uso.....	55
3.3.1.3 Diagrama de atividade.....	61
3.3.2 Modelagem de dados.....	62
3.3.2.1 Modelagem Conceitual de dados.....	63
3.3.2.2 Modelagem Lógica de dados.....	64
3.3.3 Modelagem da Arquitetura.....	66
3.3.3.1 MVC: Model-View-Controller.....	68

3.3.3.2 REST - Representational State Transfer.....	69
3.3.3.3 EDA: Event-Driven Architecture.....	70
3.3.3.4 Correlação entre as arquiteturas.....	71
3.3.4 Conclusão da modelagem.....	73
3.4 Funcionalidades.....	73
3.4.1 Autenticação e gerenciamento de seção.....	73
3.4.2 Gestão de perfil e edição concorrente.....	74
3.4.3 Gerenciamento de recomendações literárias.....	74
3.4.4 Gerenciamento de estante virtual.....	75
3.4.5 Gerenciamento de camada social.....	76
3.4.5.1 Publicação de notas na comunidade.....	76
3.4.5.2 Envio de mensagens.....	76
3.4.5.3 Seguir e ser seguido.....	77
3.4.6 Gerenciamento de estados de leitura.....	77
3.4.7 Gerenciamento de comunicação com serviços externos.....	78
3.4.7.1 Comunicação e consumo com o Google Books API.....	78
3.4.7.2 Comunicação e consumo com o Google Gemini AI.....	80
3.4.8 Conclusão das funcionalidades.....	84
3.5 Interface Gráfica.....	84
3.5.1 Tela de Login e Cadastro.....	85
3.5.2 Tela Inicial.....	88
3.5.3 Tela de Estante e de Coleção de livros.....	90
3.5.4 Telas de Perfil.....	92
3.5.5 Telas de Livros.....	95
3.5.6 Tela de Citações.....	101
3.5.7 Telas de Gerenciamento de leitura.....	102
3.5.8 Tela de Notificações.....	104
3.5.9 Chapter.....	105
3.5.10 Conclusão sobre as Interfaces do Sistema.....	107
4 CONSIDERAÇÕES FINAIS.....	108
5 REFERÊNCIAS.....	110

1 INTRODUÇÃO

A leitura desempenha um papel crucial no desenvolvimento intelectual, cultural e emocional de indivíduos e sociedades ao longo da história. Por meio da leitura, temos acesso a novas ideias, perspectivas e conhecimentos, ampliando nossos horizontes e enriquecendo nossa compreensão do mundo ao redor. Como afirmado por Melo (2014) em seu estudo sobre a importância do hábito da leitura:

“ [...] Os benefícios da prática da leitura são diversos, desenvolve senso crítico e intelectual do ser humano, é um meio transformador da construção do conhecimento, trazendo o despertar metafísico do conhecimento do mundo. ” (Melo, 2014).

No entanto, no contexto contemporâneo, testemunha-se uma transformação significativa nos padrões de leitura, impulsionada pela crescente influência das tecnologias digitais e pela rápida disseminação de informação online. A 5ª edição da pesquisa Retratos da Leitura no Brasil, realizada pelo Instituto Pró-Livro, revelou que 48% dos brasileiros são classificados como não leitores, destacando a falta de tempo e o uso predominante de entretenimento digital, como internet, redes sociais e televisão, como os principais motivos para tal cenário (Instituto Pró-Livro, 2021). Entretanto, é importante ressaltar que, mesmo diante desse contexto, há oportunidades para explorar os meios tecnológicos a fim de tornar a leitura mais atrativa e dinâmica. Como afirmado por Yusof (2021) em seu estudo recente sobre os hábitos de leitura dos estudantes na era digital:

“ [...] Por exemplo, em vez de gastar tempo lendo em uma mesa de estudo ou na biblioteca, eles (alunos) podem simplesmente pegar o telefone e sentar e relaxar no sofá enquanto leem. Além disso, em vez de ir à biblioteca e procurar materiais nas prateleiras um por um, é mais fácil para eles encontrar os materiais existentes disponíveis em formato digital. Isso significa que a era digital muda totalmente o comportamento dos hábitos de leitura dos alunos.” (Yusof, 2021).

No contexto atual da sociedade digital, a gestão eficiente de informação tornou-se um aspecto fundamental em diversas áreas do conhecimento. No âmbito da literatura, a transição para o meio digital tem gerado uma demanda crescente por ferramentas que permitam aos leitores organizar, acessar e compartilhar suas leituras de maneira eficaz. Surge, então, a necessidade de sistemas que atendam as demandas específicas dos leitores ávidos por uma experiência mais personalizada e interativa.

Sistemas *web* como o ¹*Wattpad* e o ²*Skoob* têm proporcionado aos leitores a oportunidade de manter registros virtuais de suas bibliotecas, permitindo a criação e gerenciamento de listas de livros já lidos ou desejados. Contudo, ao analisar esses sistemas identificam-se algumas deficiências. Primeiramente, eles não possibilitam a gerência e a avaliação dos livros simultaneamente à leitura, o que limita a capacidade dos usuários de interagir com suas leituras em tempo real. Além disso, não oferecem um mecanismo eficiente para o compartilhamento de acervos entre usuários, impedindo que outros leitores tenham acesso às listas de livros, avaliações e notas publicadas.

Nesse cenário, o sistema *Book Collection* emerge como uma solução para a gestão de bibliotecas pessoais e interações em torno da literatura. Portanto, esse trabalho propõe o desenvolvimento e implementação do sistema *Book Collection*, uma plataforma *web* desenvolvida com o objetivo de proporcionar aos usuários uma experiência personalizada na organização e compartilhamento de suas leituras. Através deste sistema, os usuários podem cadastrar-se e ter acesso a uma série de funcionalidades projetadas para facilitar o controle e a interação em torno de suas coleções literárias.

O *Book Collection* permite que os usuários criem listas personalizadas de livros, incluindo obras já lidas e aquelas que desejam ler no futuro. Além disso, os usuários têm a possibilidade de avaliar os livros, compartilhar notas e interagir com outros perfis, formando assim uma comunidade de leitores engajados. Essa abordagem integrada de organização pessoal e interação social visa enriquecer a experiência de leitura dos usuários e promover a troca de conhecimento e recomendações literárias.

Diante da importância da leitura no desenvolvimento humano e o impacto das tecnologias neste cenário, este trabalho visa explorar e propor o desenvolvimento de uma solução para a gestão e interação em torno da literatura. Nas seções seguintes, serão abordadas a motivação e a problemática envolvidas por trás desse trabalho, apresentando o impacto e a transformação da leitura na era digital, ademais, a potencialidade de comercialização do produto, apresentando as possíveis fontes de renda e mercados interessados neste sistema, e, por fim, difundindo os objetivos gerais e específicos que orientam esse trabalho. Ao fazê-lo, espera-se contribuir com o entendimento profundo das necessidades dos leitores contemporâneos e da interação em torno da literatura em meio digital.

¹ Wattpad - Disponível na URL: <https://www.wattpad.com> e acessado em 05/03/2024.

² Skoob - Disponível na URL: <https://www.skoob.com.br> e acessado em 05/03/2024.

1.1 Motivação

No contexto atual, marcado pela crescente digitalização e rápida disseminação de informação *online*, a leitura continua a ser uma prática essencial para o desenvolvimento intelectual, cultural e emocional dos indivíduos e das sociedades. Como Paulo Freire tão eloquentemente disse: “A leitura do mundo precede a leitura da palavra” (Freire, 1989). Através da leitura, expandimos nossos horizontes, desenvolvemos nosso senso crítico e exploramos novas perspectivas sobre o mundo que nos cerca.

Por conseguinte, as tecnologias modernas, que agora ocupam um espaço central em nossas vidas diárias, introduziram desafios únicos na maneira como nos envolvemos com a leitura. A pesquisa Retratos da Leitura no Brasil, realizada pelo Instituto Pró-Livro, destaca a necessidade urgente de abordar os hábitos de leitura em face da predominância do entretenimento digital e a constante sensação de falta de tempo (Instituto Pró-Livro, 2021). A internet, embora ofereça acesso a vasta quantidade de informações, também pode ser uma fonte de distração e desvio de atenção.

No entanto, é crucial reconhecer que a era digital também oferece oportunidades únicas para revitalizar nossos hábitos de leitura, tornando-os mais acessíveis, interativos e personalizados. As novas tecnologias, como computadores, celulares e *tablets*, permitem o acesso a uma ampla variedade de conteúdos literários em diferentes formatos e plataformas. Ao explorar essas ferramentas digitais de maneira estratégica e consciente, os leitores podem expandir suas experiências de leitura e explorar novos mundos literários de maneira inovadora.

Nesse contexto, surge a oportunidade de desenvolver soluções que integrem a tecnologia digital ao processo de leitura, oferecendo ferramentas e plataformas que atendam as necessidades e expectativas dos leitores contemporâneos, bem como cobrindo as lacunas deixadas por soluções já existentes. É dentro deste contexto que o sistema *Book Collection* emerge como uma proposta promissora, proporcionando aos usuários uma experiência personalizada na organização e compartilhamento de suas leituras.

1.2 Problemática

No cenário atual, marcado pela crescente digitalização e rápida disseminação de informações *online*, surge uma problemática essencial que precisa ser enfrentada: a necessidade premente de repensar e reestruturar nossos hábitos de leitura diante do impacto tecnológico. Esta problemática se desdobra em duas vertentes cruciais: a necessidade de

organizar leituras em um ambiente digital e o impacto da tecnologia na maneira como nos envolvemos com a leitura.

A digitalização da leitura tem apresentado uma série de desafios para os leitores, incluindo: i) a falta de organização e estrutura em bibliotecas pessoais digitais; ii) a proliferação de distrações online, e; iii) a perda da experiência sensorial associada à leitura em formato físico; iv) a facilidade de acesso a uma ampla variedade de conteúdo *online* muitas vezes resulta em uma dispersão do foco e dificuldade de engajar-se profundamente com uma obra, e; v) a ausência de ferramentas eficazes para organizar, avaliar e compartilhar leituras limita a capacidade dos leitores de interagir significativamente com o material e de aproveitar plenamente o potencial das tecnologias digitais.

“Com o desenvolvimento das tecnologias, os textos se apropriaram de peculiaridades da *web*, como a hipertextualidade, a não-linearidade, a multimídia e a interatividade, permitindo que as pessoas fiquem à frente de novos modelos de textualidade oferecidos pelos ambientes digitais.” (Monteiro, 2021).

Sendo assim,

“Mergulhadas em um universo cada vez mais informacional e tecnológico, as pessoas passaram a demandar novas habilidades de leitura. [...] Diante desse contexto, o ato de ler foi ressignificado em um processo de mudança sociocultural que altera as relações sociais, os comportamentos e as formas de perceber e de se comunicar com o outro.” (Monteiro, 2021).

No entanto, a transição para o meio digital também oferece oportunidades sem precedentes para repensar e revitalizar os hábitos de leitura. As novas tecnologias permitem o acesso a uma vasta gama de conteúdo literário em formatos diversos, possibilitando uma experiência de leitura mais acessível, interativa e personalizada. A digitalização da leitura também cria espaço para o desenvolvimento de soluções inovadoras, como o sistema em questão, *Book Collection*, que visa fornecer aos leitores uma plataforma para organizar, acessar e compartilhar suas leituras de maneira eficaz e significativa.

Diante desses desafios e oportunidades, torna-se evidente a importância de explorar soluções que contribuam para promover uma experiência de leitura enriquecedora e significativa na era digital. Ao investigar como os hábitos de leitura estão sendo moldados pela tecnologia e como podemos aproveitar ao máximo o potencial das plataformas digitais, espera-se contribuir para uma compreensão mais abrangente e eficaz das questões que permeiam o cenário contemporâneo da leitura.

1.3 Potencialidade de Comercialização

No cenário dinâmico e digital, a gestão eficiente de bibliotecas pessoais e as interações em torno da literatura tornam-se cada vez mais relevantes. Em resposta a essa demanda, o sistema *Book Collection* emerge como uma proposta projetada para proporcionar aos usuários uma experiência personalizada na organização e compartilhamento de suas leituras, objetivando transformar a forma como os leitores interagem com suas coleções literárias em um ambiente *web* e digital.

Sendo assim, o sistema *Book Collection* oferece uma série de potencialidades comerciais que podem ser exploradas em diversas áreas de aplicação. Desde a formação de parcerias estratégicas até a monetização por meio de publicidade. Assim, a plataforma possui uma ampla gama de oportunidades para o desenvolvimento de um negócio lucrativo no mercado editorial e educacional.

É importante ressaltar que, entre as tecnologias usadas no desenvolvimento do *Book Collection*, conforme descrito no capítulo de *Descrição do Sistema*, está a utilização de uma Interface de Programação de Aplicação (API) externa responsável por gerar e apresentar informações sobre os livros, o ³*Google Books API*. De acordo com os termos de uso dessa API, não é permitido cobrar dos usuários pelo uso do aplicativo sem uma permissão expressa do Google. Conforme descrito na documentação oficial da API, na seção de Termos de Serviço:

“Você não pode cobrar dos usuários nenhuma taxa pelo uso de seu aplicativo, a menos que tenha firmado um contrato separado com o Google ou obtido uma permissão por escrito do Google.”

Portanto, a monetização do sistema não incluirá a oferta de serviços *premium* diretamente aos usuários ou qualquer outro serviço, que exija uma interação monetária por parte do usuário, para o uso dos serviços prestados pela API do *Google*. Em vez disso, o *Book Collection* pode estabelecer parcerias estratégicas com editoras e livrarias, oferecendo promoções exclusivas, descontos em livros recomendados pelos usuários e até mesmo a possibilidade de venda direta de obras por meio da plataforma. Para esta última funcionalidade, seria necessário desenvolver um sistema de *e-commerce* integrado, incluindo um catálogo de produtos, carrinho de compras, processamento de pagamentos e gestão de pedidos, garantindo uma experiência de compra segura e eficiente para os usuários.

³ Google LLC. **Google Books APIs**. Disponível em: <https://developers.google.com/books>. Acesso em: 28/04/2024.

Vale ressaltar que a funcionalidade de venda direta de livros não está incluída nos requisitos atuais do sistema, mas pode ser considerada para desenvolvimento e integração futura, dependendo da evolução do projeto e das necessidades dos usuários.

Outra oportunidade seria a monetização por meio de publicidade e *marketing* de conteúdo. Ao analisar os hábitos de leitura e as preferências dos usuários, o sistema poderia oferecer oportunidades de publicidade direcionada e *marketing* de conteúdo para editoras, autores e outras partes interessadas no mercado editorial. Essas parcerias seriam benéficas tanto para a plataforma, quanto para as editoras, aumentando a visibilidade e as vendas de livros.

Por outro lado, o sistema *Book Collection* pode ser comercializado como uma solução inovadora para empresas e desenvolvedores que buscam oferecer aos usuários uma experiência personalizada na gestão de bibliotecas pessoais e interações em torno da literatura. Empresas de desenvolvimento de software, provedores de serviços tecnológicos e desenvolvedores independentes podem se interessar em adquirir ou licenciar o sistema para integração em suas plataformas existentes ou para o desenvolvimento de novos produtos.

Além disso, o *Book Collection* também poderia ser adaptado para uso em ambientes educacionais, permitindo que escolas, universidades e instituições acadêmicas gerenciem bibliotecas virtuais, acompanhem o progresso de leitura dos alunos e promovam a interação e colaboração entre estudantes e professores. Essa seria uma oportunidade de expansão significativa para a plataforma, oferecendo um valor adicional para instituições de ensino e estudantes.

Em resumo, o sistema *Book Collection* não apenas atende às necessidades dos leitores contemporâneos, mas também oferece um horizonte promissor de oportunidades comerciais em diversas áreas de aplicação.

1.4 Objetivos

1.4.1 Objetivo geral

Desenvolver um sistema *web* que proporcione aos usuários uma plataforma intuitiva e eficaz para organizar suas leituras, tanto de livros já lidos, quanto de obras que desejam ler, ao mesmo tempo em que facilita a interação e o compartilhamento de recomendações entre os usuários.

1.4.2 Objetivos específicos

São objetivos específicos deste trabalho:

- Promover um sistema organizacional de leitura;
- Implementar um sistema de avaliação de livros;
- Desenvolver uma comunidade de leitores engajados;
- Incentivar a leitura ativa.

1.5 Organização do Documento

Este documento está organizado em quatro capítulos, cada um desempenhando um papel fundamental na apresentação e desenvolvimento do sistema proposto. No segundo capítulo, será detalhado a metodologia que será utilizada no processo de desenvolvimento do sistema, abordando o processo de desenvolvimento e o fluxo das atividades realizadas. Em seguida, no terceiro capítulo, será apresentada uma descrição detalhada do sistema desenvolvido, incluindo diagramas, arquitetura, tecnologias utilizadas, telas do sistema, funcionalidades, entre outros aspectos relevantes. E, por último, o quarto capítulo, se abstém as considerações finais, a conclusão e progressão dos próximos passos deste projeto.

2 METODOLOGIA

Este capítulo detalha informações sobre a metodologia usada para o desenvolvimento do sistema *Book Collection*, apresentando informações sobre a metodologia ágil usada em seu desenvolvimento, sobre o fluxo de execução e as atividades realizadas em torno do planejamento de desenvolvimento do sistema.

2.1 Organização do Planejamento de Desenvolvimento

Esta seção aborda os métodos e recursos utilizados no desenvolvimento da solução proposta, considerando a problemática discutida anteriormente. Dado que a leitura desempenha um papel crucial na sociedade contemporânea, a criação de um sistema *web* que facilite a organização e a disseminação de informações literárias exige uma análise minuciosa, tanto do campo literário, quanto das ferramentas digitais atualmente disponíveis.

Para definir os requisitos funcionais e não-funcionais do sistema *Book Collection*, foi fundamental conduzir uma pesquisa detalhada dos sistemas existentes no mercado. Além disso, um questionário público foi elaborado e distribuído para alcançar tanto leitores experientes, quanto pessoas que ainda não têm o hábito da leitura. O objetivo foi entender seus padrões de consumo literário e explorar possíveis soluções que incentivem a leitura.

A seguir, serão apresentadas três subseções: a primeira aborda a pesquisa de trabalhos e projetos relacionados, a segunda analisa os resultados obtidos na pesquisa pública e a terceira aborda a organização dos requisitos coletados. Juntas, essas seções fornecem a base teórica e prática para o desenvolvimento do *Book Collection*.

2.1.1 Pesquisa de trabalho e Projetos relacionados

No contexto atual de desenvolvimento, a convergência entre sistemas *web* e o universo literário tem despertado grande interesse entre empresas e desenvolvedores. Portanto, o objetivo principal desta subseção é apresentar uma análise de alguns sistemas *web* que são utilizados para a organização de bibliotecas pessoais. A motivação por trás dessa análise é identificar as funcionalidades disponíveis nesses sistemas, comparando-as com as funcionalidades básicas desejadas para o desenvolvimento do *Book Collection*. A partir dessa análise, é possível identificar os diferenciais que o sistema a ser desenvolvido deve apresentar em relação aos sistemas já existentes, além de definir referências de *design* para a elaboração da solução.

Os sistemas analisados incluem o *Skoob*, o ⁴*GoodReads*, o ⁵*LibraryThing*, o *Kindle* ⁶e o *Wattpad*. O *Skoob* é uma rede social brasileira desenvolvida para leitores, lançada em 2009, permite a criação e gerenciamento de bibliotecas pessoais, compartilhamento de resenhas, avaliação de livros e interação com outros usuários. O *GoodReads*, fundado em 2006 e adquirido pela Amazon em 2013, oferece recursos semelhantes, permitindo a organização de leituras em bibliotecas pessoais, a avaliação de livros e a interação entre usuários. O *LibraryThing*, por sua vez, é uma plataforma lançada em 2005, que permite aos usuários catalogar e organizar suas coleções de livros, incluindo recursos como etiquetagem, avaliações e resenhas. E por último, o *Wattpad*, também lançado em 2006, apresenta funcionalidades de gerenciamento de leitura, avaliação e interação entre usuários, como comum nos sistemas analisados, porém seu principal diferencial concentra-se na ideia que o sistema permite que usuários publiquem seus escritos e possam ler livros na própria plataforma, promovendo uma interação maior entre os usuários desses sistemas.

Embora o *Kindle* seja uma das plataformas de leitura digital mais populares, ele não foi incluído na análise principal por não se caracterizar como uma rede social literária. Lançado pela Amazon em 2007, o *Kindle* se destaca pela sua biblioteca virtual, permitindo a compra, o armazenamento e a leitura de livros digitais com recursos como marcação de trechos e personalização de fontes. Apesar de oferecer funcionalidades como compartilhamento de anotações e destaques, sua interação entre usuários é limitada quando comparada a redes sociais literárias como *Skoob* e *GoodReads*.

Para a análise dos sistemas mencionados, as buscas foram realizadas utilizando fontes confiáveis como artigos acadêmicos, sites especializados e páginas oficiais das plataformas. As principais palavras-chave usadas durante a pesquisa foram: "*rede social para leitores*", "*biblioteca virtual*", "*avaliação de livros*", "*plataforma de leitura*", "*interação entre leitores*", e os próprios nomes das plataformas. As buscas ocorreram em bancos de dados acadêmicos como *Google Scholar*⁷, além de visitas a páginas como *Wikipedia*⁸, resenhas e blogs especializados, com o intuito de garantir uma visão ampla e precisa sobre os serviços oferecidos por cada plataforma.

As funcionalidades analisadas nesses sistemas compreendem uma série de características essenciais previstas para o sistema *Book Collection*, entre as quais:

1. **Acervo digital:** Engloba a disponibilidade de livros, informações sobre eles e a precisão na busca por títulos específicos;

⁴ GoodReads - Disponível na URL: <https://www.goodreads.com> e acessado em 08/04/2024.

⁵ LibraryThing- Disponível na URL: <https://br.librarything.com> e acessado em 08/04/2024.

⁶ Kindle - Disponível na URL: <https://www.amazon.com.br> e acessado em 15/02/2025.

⁷ Google Scholar - Disponível na URL: <https://www.google.com> e acessado em 08/04/2024.

⁸ Wikipedia - Disponível na URL: <https://pt.wikipedia.org> e acessado em 08/04/2024.

2. **Interação social:** Diz respeito à presença de recursos que promovem interação entre usuários, como salas de bate-papo, acesso a perfis e compartilhamento de conteúdo, destacando-se pela facilidade e eficácia dessa interação;

3. **Avaliação de livros:** Relaciona-se à capacidade de os usuários avaliarem livros e/ou leituras já realizadas, proporcionando um meio para expressar opiniões e recomendações sobre obras literárias;

4. **Publicação de postagens:** Refere-se à capacidade dos usuários de criar e compartilhar conteúdo, como textos, citações e imagens, tornando-os visíveis para outros usuários da plataforma;

5. **Recomendação de leitura:** Envolvendo a implementação de sistemas que sugerem livros com base nos interesses e histórico de leitura do usuário, facilitando a descoberta de novas obras;

6. **Gerenciamento de leitura:** Aborda ferramentas que auxiliam os usuários na organização de suas leituras, permitindo marcar o progresso de leitura e gerenciar marcadores no sistema;

7. **Organização de bibliotecas:** Engloba a capacidade de os usuários criarem e gerenciarem suas bibliotecas pessoais, permitindo a classificação dos livros em categorias como "Já lidos", "Lendo" e "Desejo ler", facilitando o acesso e a organização das obras conforme as preferências individuais.

8. **Segurança de Direitos Autorais:** Refere-se às medidas adotadas pelas plataformas para garantir o respeito aos direitos de propriedade intelectual, prevenindo a reprodução não autorizada de conteúdos protegidos. Isso inclui políticas de publicação, controle sobre trechos compartilhados, prevenção de plágio e mecanismos para remoção de materiais que infrinjam as normas de *copyright*.

A abordagem para avaliação e comparação segue as definições a seguir:

1. **Inexistente:** A característica não está presente no sistema;
2. **Baixo:** A característica está presente, mas de forma básica e com poucos recursos;
3. **Médio:** A característica está presente, mas pode faltar algum recurso para ser completa;

4. **Alto:** A característica está bem implementada, com todos os recursos necessários.

Neste contexto, após analisar os sistemas selecionados e avaliar a presença e usabilidade das funcionalidades em cada aplicação, foi elaborada o seguinte quadro comparativo:

Quadro 01 - Análise das características dos sistemas analisados.

Características	Skoob	Goodreads	LibraryThing	Wattpad	Book Collection
Acervo Digital	Alto	Alto	Alto	Baixo	Alto
Interação Social	Médio	Médio	Baixo	Alto	Alto
Avaliação de livros	Alto	Alto	Baixo	Baixo	Alto
Publicação de posts	Inexistente	Inexistente	Inexistente	Alto	Alto
Recomendação de leituras	Alto	Alto	Médio	Médio	Alto
Gerenciamento de leitura	Baixo	Inexistente	Inexistente	Médio	Alto
Organização de bibliotecas	Alto	Médio	Médio	Médio	Alto
Segurança de Direitos Autorais	Médio	Alto	Médio	Baixo	Alto

Fonte: Elaborado pelo o autor.

Com base na análise do quadro apresentado, é possível inferir a comparação das funcionalidades e o nível de implementação em cada sistema avaliado.

Inicialmente, o *Skoob* destaca-se por seu excelente acervo digital, proporcionando uma eficiente busca e amplas informações sobre os livros. No entanto, sua interação social e gerenciamento de leitura apresentam deficiências, e há ausência da funcionalidade de publicação de postagens. Em relação à segurança dos direitos autorais, o *Skoob* possui nível médio, pois evita o compartilhamento de livros completos, mas permite resenhas e interações que podem incluir conteúdos protegidos sem moderação rigorosa.

Similarmente, o *Goodreads* também possui um acervo digital robusto, mas enfrenta desafios na interação entre usuários e na ausência de gerenciamento de leitura e publicação de postagens. No que se refere à segurança dos direitos autorais, sua classificação é alta, pois faz parte do ecossistema da *Amazon* e segue diretrizes rígidas de proteção de conteúdo, evitando violações.

Já o *LibraryThing* oferece um bom acervo digital, embora a pesquisa não seja tão intuitiva, e suas funcionalidades sociais e de gerenciamento de leitura também careçam de desenvolvimento. No entanto, ele apresenta segurança de direitos autorais em nível médio, pois não permite compartilhamento de obras, mas disponibiliza trechos e informações detalhadas sem mecanismos avançados de controle.

Por fim, o *Wattpad* se destaca pela capacidade de produção de obras na plataforma, promovendo interação entre os usuários, mas enfrenta limitações nas outras funcionalidades e na diversidade do acervo. No entanto, sua segurança de direitos autorais é baixa, pois, apesar de possuir algumas diretrizes, há maior risco de publicação de conteúdos não autorizados pelos próprios usuários, tornando a fiscalização menos eficaz.

No *Book Collection*, a segurança de direitos autorais é uma preocupação central, sendo garantida por meio de diversas medidas que asseguram a conformidade com as normas vigentes. Em primeiro lugar, o sistema não armazena ou distribui livros diretamente, mas obtém todas as informações a partir da *Google Books API*, um serviço oficial que segue diretrizes rigorosas de proteção autoral. Dessa forma, os dados exibidos no sistema são provenientes de fontes autorizadas, garantindo que não haja violação de direitos.

Além disso, o *Book Collection* limita-se a oferecer funcionalidades voltadas à organização pessoal da leitura, permitindo que os usuários adicionem livros às suas coleções, façam avaliações e registrem seu progresso de leitura. Não há a possibilidade de *upload*, compartilhamento ou distribuição de livros completos, o que elimina qualquer risco de uso indevido de obras protegidas.

Por fim, ao restringir suas funcionalidades à catalogação e avaliação de livros, sem permitir a publicação de conteúdo próprio por parte dos usuários, o sistema reforça seu compromisso com a segurança dos direitos autorais. Dessa maneira, o *Book Collection* pode ser classificado com um nível alto de proteção, garantindo que sua proposta de gestão de leitura seja realizada dentro dos parâmetros legais.

Em suma, cada um dos sistemas analisados possuem pontos fortes e áreas a serem melhoradas. O desenvolvimento do *Book Collection* busca integrar as melhores características dessas plataformas, priorizando a eficiência na busca de livros, a interação social intuitiva, o gerenciamento simplificado de leitura e a promoção de uma comunidade

engajada de leitores. Este processo visa oferecer aos usuários uma experiência completa e enriquecedora no universo da leitura online.

2.1.2 Análise de pesquisa pública

A pesquisa foi conduzida utilizando o ⁹*Google Forms*, uma ferramenta gratuita e amplamente reconhecida para a criação de formulários e questionários online. O *Google Forms* é uma plataforma versátil, que permite a coleta e análise de dados de maneira eficiente, sendo ideal para pesquisas que necessitam alcançar um público amplo e diversificado. Esta ferramenta foi escolhida por sua facilidade de uso, tanto para o pesquisador, quanto para os participantes, além de sua capacidade de compilar e organizar as respostas de forma automatizada.

O questionário foi direcionado, tanto a leitores frequentes, quanto a não leitores, com o objetivo de entender os padrões de leitura na sociedade. Foram exploradas questões relacionadas a como as pessoas leem, quanto tempo dedicam à leitura e quais fatores impactam esse hábito, sejam eles positivos ou negativos. A pesquisa também buscou identificar os principais sistemas e plataformas literárias que os participantes utilizam, destacando as funcionalidades que eles mais valorizam, aquelas que sentem falta e as necessidades que ainda não foram plenamente atendidas por esses sistemas.

Além de coletar informações sobre os hábitos e preferências dos usuários, a pesquisa incluiu a apresentação de protótipos de telas do sistema *Book Collection*, elaborados inicialmente no *Figma*¹⁰. Os participantes foram convidados a avaliar o *design* e a usabilidade dessas telas, expressando suas opiniões sobre a interface proposta e o potencial interesse em utilizar o sistema. As críticas e sugestões fornecidas pelos usuários foram fundamentais para compreender as expectativas em relação à experiência do usuário e o *layout* da aplicação.

Para melhor compreender os hábitos de leitura dos usuários e identificar oportunidades de aprimoramento no desenvolvimento do *Book Collection*, os dados coletados foram organizados e analisados de forma sistemática. A pesquisa buscou avaliar desde o perfil dos participantes até suas preferências em relação a funcionalidades e usabilidade de plataformas literárias. Os dados analisados foram gerados a partir da pesquisa pública, cujas respostas foram registradas em uma planilha no *Google Sheets*¹¹ para posterior avaliação. O Quadro 02 a seguir apresenta um resumo dos principais resultados obtidos, destacando as informações mais relevantes para a concepção e aprimoramento do sistema.

⁹ Google Forms. **Pesquisa pública realizada.** Disponível em: <https://forms.gle/9uDVxcKE88hoXYUz6>.

¹⁰ Figma. **Prototipação do sistema.** Disponível em: [https://www.figma.com/design/Book-Collection-\(TCC\)](https://www.figma.com/design/Book-Collection-(TCC)).

¹¹ Google Sheets. **Dados da pesquisa realizada.** Disponível em: <https://docs.google.com/spreadsheets>

Quadro 02 - Análise de pesquisa pública.

Categoria	Resultados Obtidos
Perfil dos participantes	<p>Com base na quantidade de livros lidos nos últimos seis meses, os entrevistados foram categorizados em três perfis de leitura:</p> <ul style="list-style-type: none"> ● Leitores frequentes (37,4%): aqueles que leram entre 6 e mais de 20 livros; ● Leitores ocasionais (38,5%): aqueles que leram entre 3 e 5 livros; ● Não leitores (23,1%): aqueles que leram no máximo 2 livros ou nenhum.
Tempo médio de leitura	<p>Com base na frequência de leitura declarada pelos entrevistados, observou-se a seguinte distribuição:</p> <ul style="list-style-type: none"> ● 38,5% leem <i>diariamente</i>; ● 26,9% leem <i>semanalmente</i>; ● 15,4% leem <i>raramente</i>;
Principais obstáculos à leitura	<p>Entre os principais obstáculos à leitura apontados pelos entrevistados — com a possibilidade de seleção múltipla — destacam-se:</p> <ul style="list-style-type: none"> ● 76,9% mencionam a <i>falta de tempo</i> como o principal impedimento; ● 30,8% indicam a <i>preferência por outras formas de entretenimento</i> ou a <i>dificuldade em encontrar livros de interesse</i>.
Principais plataformas utilizadas	<p>Entre as principais plataformas utilizadas pelos entrevistados — com a possibilidade de seleção múltipla — destacam-se:</p> <ul style="list-style-type: none"> ● 46,2% não usam nenhuma plataforma de leitura; ● 38,8% usam preferencialmente o <i>Skoob</i> e o <i>Kindle</i>;
Funcionalidades analisadas	<p>Entre as funcionalidades mais valorizadas nas plataformas utilizadas pelos entrevistados — considerando a possibilidade de múltiplas escolhas — destacam-se:</p> <ul style="list-style-type: none"> ● 46,2% se interessam por <i>leitura de e-books</i>; ● 30,8% se interessam pelas <i>avaliações e resenhas de livros</i>; ● 26,9% se interessam pela <i>organização de estantes virtuais</i> e pelo <i>acompanhamento do progresso de leitura</i>.
Funcionalidades ausentes	<p>Entre as funcionalidades apontadas pelos entrevistados como ausentes ou passíveis de aprimoramento — considerando a possibilidade de múltiplas escolhas — destacam-se:</p> <ul style="list-style-type: none"> ● 38,5% indicaram a ausência de <i>ferramentas para anotação e marcação durante a leitura</i>; ● 30,8% indicaram a presença de uma <i>interface mais</i>

	<p><i>amigável e intuitiva;</i></p> <ul style="list-style-type: none"> • 26,9% indicaram uma melhoria no <i>sistema de recomendações de livros</i>.
--	--

Fonte: Elaborado pelo o autor.

Além da análise preliminar, apresentada no quadro acima, pode-se realizar uma análise resumida sobre o interesse apresentado pelos entrevistados ao serem questionados sobre as funcionalidades do sistema. Assim, a maioria dos participantes demonstrou alto interesse na plataforma, destacando funcionalidades como organização da estante virtual, avaliações e resenhas e interações sociais. Um grupo menor expressou interesse moderado, apontando que utilizariam algumas funcionalidades, mas não todas. Poucos participantes mostraram baixo interesse, indicando que não interagiriam com certos recursos, como a publicação de notas de comunidade.

Além disso, destaca-se que a interface do sistema foi bem recebida, com a maior parte dos participantes considerando-a intuitiva, bem distribuída e funcional. Muitos destacaram a facilidade de navegação, o *design* agradável e a boa organização das funcionalidades. Algumas sugestões de melhoria foram mencionadas, como maior personalização da estante virtual e do perfil. Apenas uma pequena parcela apontou pontos críticos, sugerindo ajustes em cores, *layout* ou funcionalidades específicas.

Os resultados desta pesquisa foram de extrema importância para o desenvolvimento das próximas etapas do sistema *Book Collection*. Eles não apenas forneceram *insights* sobre os principais interesses e necessidades dos futuros usuários, mas também ajudaram a identificar as áreas que necessitam de maior atenção para melhorar a usabilidade e a funcionalidade do sistema. Em resumo, os dados coletados moldaram de maneira significativa as decisões de *design*, funcionalidade e abordagem do projeto, garantindo que o produto final seja o mais alinhado possível com as expectativas e demandas do público-alvo.

2.1.3 Organização dos requisitos

Com base na pesquisa e estudo realizados, foi possível elaborar uma série de requisitos, conforme apresentado no capítulo de *Descrição do Sistema*. Esses requisitos não apenas estabelecem as relações com os sistemas previamente analisados, destacando as semelhanças e diferenças entre o sistema proposto e as tecnologias disponíveis no mercado, mas também aproveitam os dados coletados através da pesquisa pública. Com uma base estrutural sólida e uma listagem primordial dos requisitos, procedeu-se à documentação desses elementos.

A gestão dos requisitos segue utilizando a ferramenta de organização *Jira*¹². Desenvolvido pela *Atlassian*. O *Jira* é uma ferramenta poderosa para gerenciamento de projetos e rastreamento de tarefas, especialmente eficaz na gestão de requisitos de *software*. Sua utilização permite uma organização clara e estruturada dos requisitos, facilitando a comunicação entre os membros de uma equipe ágil e assegurando que todos os aspectos do projeto sejam monitorados e gerenciados eficientemente. O *Jira* também oferece recursos como rastreamento de progresso, gerenciamento de prioridades e integração com outras ferramentas de desenvolvimento, o que contribui significativamente para a agilidade e eficiência do processo de desenvolvimento do sistema *Book Collection*. Segundo a documentação oficial da ferramenta:

“É a fonte única de informações de todo o ciclo de vida de desenvolvimento, permitindo que as equipes autônomas tenham o contexto necessário para que ajam com rapidez e permaneçam alinhadas à meta de negócios mais importante.”

Dessa forma, antes do início do desenvolvimento do sistema, os requisitos foram coletados e organizados no *backlog* do projeto. Essa etapa permitiu estruturar de maneira clara todas as demandas do sistema, possibilitando uma visão abrangente do escopo e das prioridades do desenvolvimento. No total, foram definidos 58 itens, todos devidamente concluídos ao final do projeto. A Figura 01 a seguir ilustra parte desse *backlog*, demonstrando como os requisitos foram distribuídos e acompanhados ao longo do ciclo de desenvolvimento.

Figura 01 - Produtos do Backlog do sistema

Tipo	Chave	Resumo	Responsável	Relator	Prioridade	Status	Resolução	Criado	Atualiz
☑	BC-58	BFF: Desenvolver classe de conexão com a api da API	DA Douglas da Silva	DA D...	Medium	CONCLUÍDO	Itens concluídos	16 de nov. de 2024	29 de jan. de 2025
☑	BC-57	BFF: Resolução dos possíveis problemas entre o Front-end e o Back-end do sistema	DA Douglas da Silva	DA D...	Medium	CONCLUÍDO	Itens concluídos	16 de nov. de 2024	29 de jan. de 2025
☑	BC-56	BFF: Desenvolver método para salvar dados nos cookies da aplicação	DA Douglas da Silva	DA D...	Medium	CONCLUÍDO	Itens concluídos	16 de nov. de 2024	29 de jan. de 2025
☑	BC-55	BFF: Desenvolver método de acesso e programação de tempo real para o chat e informações de busca de perfil	DA Douglas da Silva	DA D...	Medium	CONCLUÍDO	Itens concluídos	16 de nov. de 2024	29 de jan. de 2025
☑	BC-54	BFF: Desenvolver contexto para salvar dados das coleções de usuário	DA Douglas da Silva	DA D...	Medium	CONCLUÍDO	Itens concluídos	16 de nov. de 2024	29 de jan. de 2025
☑	BC-53	BFF: Desenvolver contexto para salvar dados de um usuário	DA Douglas da Silva	DA D...	Medium	CONCLUÍDO	Itens concluídos	16 de nov. de 2024	29 de jan. de 2025
☑	BC-52	BFF: Desenvolver contexto para autenticação de um usuário	DA Douglas da Silva	DA D...	Medium	CONCLUÍDO	Itens concluídos	16 de nov. de 2024	29 de jan. de 2025
☑	BC-51	BFF: Desenvolver Classe para os serviços vindos da API	DA Douglas da Silva	DA D...	Medium	CONCLUÍDO	Itens concluídos	16 de nov. de 2024	29 de jan. de 2025
☑	BC-50	API: Documentar testes unitários	DA Douglas da Silva	DA D...	Medium	CONCLUÍDO	Itens concluídos	31 de out. de 2024	16 de nov. de 2024
☑	BC-49	API: Desenvolver funcionalidade de um "User" seguir e	DA Douglas da Silva	DA D...	Medium	CONCLUÍDO	Itens concluídos	23 de out. de 2024	28 de out. de 2024

Fonte: Elaborado pelo o autor.

¹² Jira - Disponível em: <https://www.atlassian.com>. Acessado dia: 28/06/2024.

Ao adotar o *Jira*, o sistema *Book Collection* se beneficiou de uma abordagem estruturada e eficiente na gestão de requisitos, o que é essencial para o sucesso do projeto. A ferramenta não só melhora a organização e clareza das informações, mas também assegura que os objetivos do projeto sejam cumpridos de maneira consistente e alinhada com as metas de negócios estabelecidas.

2.2 Desenvolvimento ágil e Fluxo de execução das atividades

Esta seção irá difundir a metodologia ágil escolhida para o desenvolvimento do sistema *Book Collection*, além de elucidar sobre o plano inicial de etapas e metas que geriram este cenário. Desse modo, para o desenvolvimento do sistema, foi adotada uma abordagem ágil baseada no *Scrum Solo*¹³, adaptada para projetos conduzidos por um único desenvolvedor. Esta metodologia combina os princípios do *Scrum* com práticas individuais de organização e produtividade. Além disso, foi integrado ao desenvolvimento orientado por testes, do inglês *Test-Driven Development* (TDD), garantindo que o código seja continuamente validado por meio de testes automatizados. A seguir, serão detalhados ambos os métodos e sua aplicação neste projeto.

2.2.1 Scrum Solo

O *Scrum* é uma metodologia ágil de gerenciamento de projetos amplamente utilizada, criada por Ken Schwaber e Jeff Sutherland nos anos 1990. Caracteriza-se por sua abordagem interativa e colaborativa, focada na entrega contínua de valor. O *Scrum* promove a inspeção e adaptação constantes, permitindo que as equipes ajustem seus processos de acordo com o progresso e resultados obtidos. A metodologia divide o desenvolvimento em *sprints* curtos, onde o trabalho é planejado, executado, revisado e melhorado continuamente. Como afirmado por Schwaber & Sutherland (2014), "a inspeção habilita a adaptação", destacando que os eventos do *Scrum* são projetados para provocar melhorias.

No entanto, para projetos conduzidos por um único desenvolvedor, como é o caso do sistema *Book Collection*, o *Scrum* tradicional precisa ser adaptado. Surge então o *Scrum Solo*, uma variação que combina os princípios do *Scrum* com as boas práticas do *Personal Software Process* (PSP), que visa a gestão e desenvolvimento de *software* de qualidade. Assim, o *Scrum Solo* ajusta cerimônias e práticas do *Scrum* tradicional, para o contexto individual, mantendo a organização, o foco e a melhoria contínua.

¹³ Scrum Solo. Disponível em: <https://engenhariasoftware.files.wordpress.com/2016/04/scrum-solo.pdf>. Acesso em: 28/07/2024.

No *Scrum Solo*, o desenvolvedor mantém um *Product Backlog* com todas as funcionalidades e requisitos priorizados. Cada *sprint*, com duração de uma a quatro semanas, envolve a seleção de itens do *backlog* para desenvolvimento, sendo seguido de revisões e retrospectivas regulares. Embora o trabalho seja realizado individualmente, o *Scrum Solo* oferece uma estrutura clara e disciplinada, permitindo ao desenvolvedor monitorar o progresso, identificar melhorias e ajustar prioridades de forma eficiente.

A aplicação do *Scrum Solo* no desenvolvimento do sistema *Book Collection* traz várias vantagens. Primeiramente, a metodologia promove a disciplina e o foco, permitindo um progresso constante e eficiente. Além disso, o *Scrum Solo* permite ao desenvolvedor ajustar prioridades e adaptar-se rapidamente a mudanças ou novos requisitos, garantindo que o projeto permaneça alinhado com as necessidades dos usuários. A prática de revisões e retrospectivas regulares incentiva a reflexão sobre o processo e a busca por melhorias contínuas, resultando em um produto de alta qualidade.

Ao desenvolver o sistema *Book Collection*, usando o *Scrum Solo*, assegura-se uma abordagem estruturada e eficiente, facilitando a entrega de um sistema funcional e bem projetado que atenda às necessidades dos usuários de forma eficaz e inovadora.

2.2.2 Test-driven Development (TDD)

O *Test-Driven Development* (TDD) é uma prática derivada das metodologias ágeis, como o *eXtreme Programming* (XP), que coloca os testes como elemento central no processo de desenvolvimento de software. O conceito básico do TDD é a criação de testes automatizados antes mesmo de implementar o código, o que facilita o entendimento do projeto. Assim, segundo Pressman (2011), “os componentes individuais são testados para garantir que operem corretamente.”

Uma das principais características do TDD é o ciclo de desenvolvimento “*Red-Green-Refactor*”, onde o desenvolvedor começa escrevendo um teste que falha (*Red*), depois implementa o código que faz o teste passar (*Green*) e, por fim, refatora o código para melhorá-lo sem quebrar os testes (*Refactor*). Essa metodologia permite que o código seja testado continuamente, garantindo que as funcionalidades implementadas atendam aos requisitos especificados de forma precisa e segura.

No contexto do *Book Collection*, o uso do *Jest*¹⁴, uma popular ferramenta de testes para *JavaScript* e *TypeScript*, se alinha diretamente com a metodologia TDD. O *Jest* oferece um ambiente eficiente para testes unitários e de integração, permitindo simular diferentes cenários e validar o comportamento de funcionalidades. Ao implementar o *Jest*, o *Book*

¹⁴ Jest. Disponível em: <https://jestjs.io/pt-BR/>. Acesso em: 05/09/2024.

Collection se beneficia de uma infraestrutura que garante qualidade e previne regressões no código, facilitando o desenvolvimento contínuo.

Seguindo essa abordagem, os testes do *Book Collection* foram conduzidos em conjunto com o desenvolvimento do código, seguindo os princípios do TDD. O processo era estruturado da seguinte forma: antes do início da implementação, eram planejados os casos de teste, definindo os comportamentos esperados da funcionalidade, as validações necessárias e os possíveis cenários de erro. O Quadro 03, exemplifica como esses casos eram organizados, garantindo que cada função atendesse aos critérios estabelecidos.

Quadro 03 - Exemplo de Caso de Teste

Identificadores	Caso de Teste: Criação de usuário
Funcionalidade	Criar uma instância de um usuário.
Resumo	Criar uma instância de um usuário, preenchendo suas informações corretamente e salvando os dados no banco de dados.
Pré-condições	<ol style="list-style-type: none"> 1. A base estrutural da API e o modelo de banco de dados já devem ter sido construídos; 2. Funcionalidades, como: Validação dos campos e serviços da entidade, já devem ter sido desenvolvidos.
Procedimentos	<ol style="list-style-type: none"> 1. Informar os campos necessários, corretamente, para a criação de um usuário, (ex: Nome, e-mail, senha...); 2. O sistema deve validar as informações para garantir que estão corretas, bem como garantir que não são informações nulas; 3. O sistema deve validar, que as informações tidas como “únicas”, como o Nome de Perfil ou o Email do usuário, não esteja sendo usado por outro perfil; 4. O sistema deve, caso as informações estejam corretas, salvar os dados do usuário no banco de dados principal.
Pós-condições	<ol style="list-style-type: none"> 1. Ao criar um novo usuário, o sistema deve responder à requisição com o código de <i>status HTTP 201 (Created)</i>, indicando que a criação foi bem-sucedida. Esse status sinaliza que um novo recurso foi gerado com sucesso no servidor.
Resultado encontrado	

Fonte: Elaborado pelo o autor.

Após a definição dos casos de teste, para cada *sprint* de desenvolvimento, a implementação do código era realizada de forma meticulosa, seguindo o que havia sido previamente planejado, bem como, se necessário, realizando alterações. Durante essa etapa, também eram aplicados princípios de boas práticas, como o código limpo, do inglês *Clean Code*, e padrões de projeto adequados para garantir a legibilidade e a manutenção do código.

Uma vez implementada a funcionalidade, os testes unitários eram executados para validar seu comportamento. O Algoritmo 01, apresenta um exemplo desses testes, demonstrando como o *Jest* foi utilizado para verificar a correta execução das funções e prevenir possíveis falhas antes da entrega final.

Algoritmo 01 - Exemplo de Teste Unitário

```
1  it("Deve ser possível criar um usuário", async () => {
2    //Criando objeto da request
3    const req = {
4      body: {
5        email: "testBookCollection@gmail.com",
6        password: "123456",
7        profileName: "7-BC",
8        userName: "User Book Collection"
9      }
10   } as Request;
11
12   //Criando objeto da response
13   const res = {
14     status: jest.fn().mockReturnThis(),
15     json: jest.fn(),
16   } as unknown as Response;
17
18   //Procedimento
19   await userController.createUser(req, res);
20
21   //Resultado
22   expect(prismaMock.user.findUnique).toHaveBeenCalledWith({
23     where: { email: 'testBookCollection@gmail.com' }
24   });
25   expect(prismaMock.user.findUnique).toHaveBeenCalledWith({
26     where: { profileName: "7-BK" }
27   });
28   expect(prismaMock.user.create).toHaveBeenCalled();
29   expect(res.status).toHaveBeenCalledWith(201);
30 });
31
```

Fonte: Elaborado pelo o autor.

Após a execução dos testes unitários, o código passava por uma reavaliação, permitindo a identificação e correção de possíveis erros, além de assegurar seu funcionamento ideal. Esse processo contínuo de validação contribuiu para a robustez do sistema, garantindo que cada funcionalidade fosse implementada com qualidade desde o início. Embora a metodologia exigisse um esforço inicial metucioso, a integração dos testes ao fluxo de

desenvolvimento resultou em um processo mais eficiente e confiável, reduzindo retrabalho e facilitando a manutenção do código.

2.2.3 Fluxo de Desenvolvimento

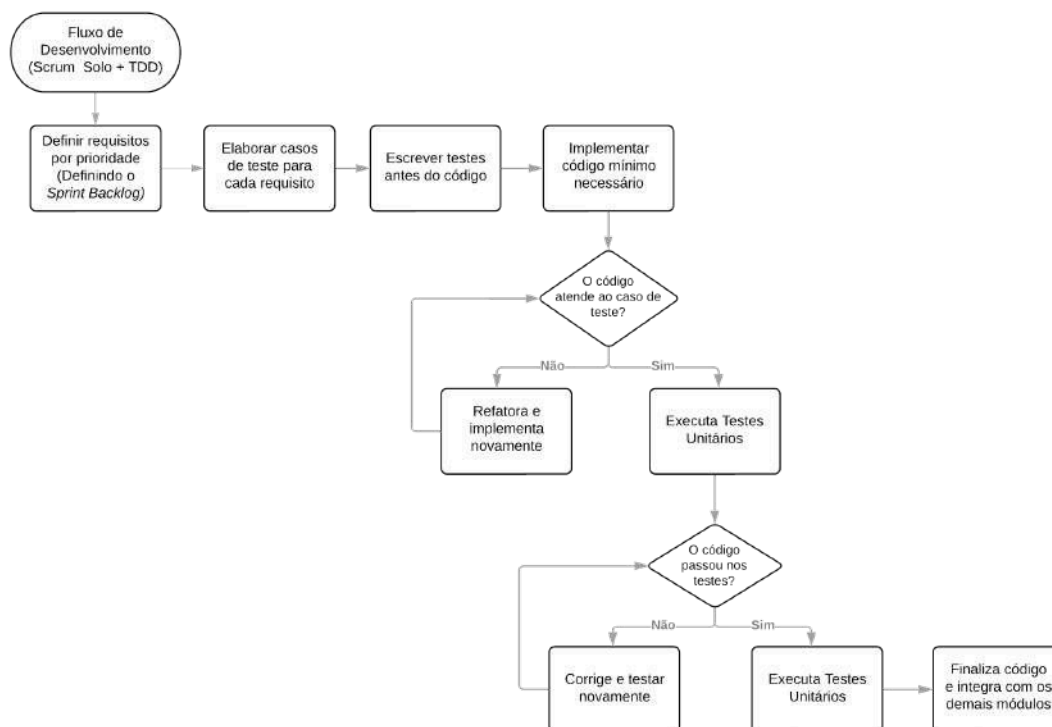
O fluxo de desenvolvimento do sistema, conforme apresentado nas subseções anteriores, baseou-se na integração entre as metodologias ágeis *Scrum Solo* e TDD, aliada ao acompanhamento dos requisitos previamente coletados e organizados na plataforma de gestão *Jira*. Dessa forma, o processo seguiu um ciclo estruturado de *sprints* de duas semanas, com etapas bem definidas para garantir a qualidade e a eficiência do desenvolvimento.

Inicialmente, eram selecionados os requisitos a serem implementados no ciclo, formando o *sprint backlog*. A priorização desses requisitos seguia critérios estratégicos, assegurando que as funcionalidades mais essenciais fossem desenvolvidas primeiro. Após essa definição, iniciava-se a fase de maturação do código, na qual casos de teste e testes unitários eram elaborados antes da implementação, conforme descrito na subseção sobre TDD.

Com os testes previamente estabelecidos, o processo de implementação era conduzido de forma iterativa, garantindo que cada nova funcionalidade fosse validada continuamente. Assim, o código era refinado e testado ao longo do *sprint*, assegurando sua estabilidade e aderência aos requisitos definidos.

A Figura 02 ilustra o fluxo de desenvolvimento adotado no projeto, evidenciando a relação entre o *Scrum Solo* e o TDD. No modelo aplicado, as iterações seguiram ciclos incrementais de duas semanas, nos quais os requisitos eram priorizados e convertidos em casos de teste antes da implementação do código. A fase de testes era conduzida de forma rigorosa, com execuções contínuas e refatorações sempre que necessário, garantindo a qualidade e a integridade do *software* desde o início.

Figura 02 - Diagrama de Fluxo



Fonte: Elaborado pelo o autor.

A adoção do TDD, aliada ao *Scrum Solo*, tornou o desenvolvimento do *Book Collection* ainda mais estruturado e seguro. Além de assegurar que cada nova funcionalidade fosse devidamente testada, o TDD incentivou uma cultura de código limpo e bem organizado, essencial para um sistema que busca oferecer uma experiência confiável e consistente aos usuários.

Dessa forma, o fluxo de desenvolvimento ágil seguiu ciclos incrementais e iterativos, com *sprints* de duas semanas. O *Scrum Solo* forneceu a estrutura para planejamento e execução, enquanto o TDD garantiu a qualidade do código desde o início, com fases claras de pesquisa, *design*, reavaliação, implementação, testes e validação. A integração dessas abordagens promoveu um desenvolvimento eficiente e confiável para o projeto.

2.2.4 Fluxo de realização das atividades

O plano de execução do projeto segue um fluxo organizado e estruturado, dividido em etapas que abrangem desde a pesquisa inicial até a conclusão e entrega do sistema. Cada fase foi cuidadosamente planejada para garantir uma progressão lógica e eficiente do trabalho. O processo inclui a definição de requisitos, seleção de tecnologias, desenvolvimento do *backend* e *frontend*, integração de sistemas, implementação de

funcionalidades específicas e testes do sistema para garantir a qualidade do produto final. Ao longo de todo o período de pesquisa e desenvolvimento, cada etapa recebeu a devida atenção, permitindo uma abordagem sistemática e focada na obtenção dos resultados desejados.

Desse modo, apresenta-se a seguir a definição de cada fase do desenvolvimento do sistema, conforme discutido anteriormente, bem como os objetivos e resultados de cada uma delas.

- **Pesquisa e levantamento dos requisitos:** A primeira fase do desenvolvimento do projeto envolve o estudo e revisão das informações sobre o tema e as tecnologias que serão utilizadas no desenvolvimento do sistema. O objetivo desta fase é identificar e levantar os requisitos do sistema, definir as funcionalidades básicas e selecionar as principais tecnologias a serem usadas.
- **Design e prototipação:** A segunda fase foca na criação de protótipos das diferentes partes do projeto. Nesta etapa, serão desenvolvidos protótipos para a arquitetura do banco de dados, arquitetura da API, arquitetura do sistema e *design* das interfaces da aplicação.
- **Reavaliação:** A terceira fase desempenha um papel fundamental, pois tem como principal função avaliar os produtos inicialmente desenvolvidos e realizar a refatoração necessária, permitindo a continuidade eficiente das próximas etapas do desenvolvimento.
- **Implementação:** A quarta fase consiste na implementação das atividades e protótipos definidos nas fases anteriores. Nessa etapa, o foco será a construção das funcionalidades e componentes do sistema conforme planejado, buscando sempre validar o desenvolvimento realizado.
- **Testes e validação:** A quinta e última fase envolve a execução de testes no sistema, avaliando todo o processo, as atividades realizadas e os resultados obtidos nas fases anteriores. O objetivo é validar o produto desenvolvido e prepará-lo para a entrega final.

Assim, todo o fluxo de desenvolvimento do projeto será guiado por metas claras, com cada fase organizada para alcançar os melhores resultados. Apesar da separação entre as fases do desenvolvimento, cada uma é responsável por revisar e validar o trabalho da anterior, assegurando a continuidade e a funcionalidade do projeto. A seguir, será apresentado um quadro detalhando as atividades realizadas em cada fase e seus respectivos prazos.

Quadro 04 - Fluxo de execução e desenvolvimento

Fase	Período (em semanas)	Fase de desenvolvimento	Atividades
Pesquisa e planejamento inicial	1ª e 2ª semanas	Pesquisa e levantamento de requisitos	<ul style="list-style-type: none"> ● Estudo sobre os sistemas com mesmo domínio; ● Realização de pesquisa por meio de um formulário público; ● Definição dos requisitos funcionais e não funcionais do sistema; ● Detalhamento do escopo de atividades e cronograma do projeto; ● Pesquisa e seleção de tecnologias e ferramentas necessárias; ● Elaboração de um plano de estudo e desenvolvimento.
Planejamento do backend	3ª e 4ª semanas	<i>Design</i> e prototipação	<ul style="list-style-type: none"> ● Definição das atividades e funcionalidades da API do sistema; ● Detalhamento do escopo e das entidades para o banco de dados; ● Desenvolvimento de diagramas conceitual e lógico do banco de dados; ● Estudo sobre ferramentas utilizadas para o desenvolvimento, como: ORM, API do <i>Google Books</i>; ● Elaboração de um plano de desenvolvimento do <i>backend</i> do projeto e atualização dos requisitos.
Planejamento do frontend	5ª a 7ª semanas	<i>Design</i> e prototipação	<ul style="list-style-type: none"> ● Estudo de <i>User Experience (UX)</i> e <i>User Interface (UI)</i>; ● Estudos para a implementação de ações e atividades da API no <i>frontend</i>;

			<ul style="list-style-type: none"> • Criação do <i>design</i> do sistema utilizando ferramentas como o <i>Figma</i>; • Elaboração e definição do plano de desenvolvimento do projeto e atualizações dos requisitos.
Reavaliação dos produtos iniciais	8ª a 10ª semana	<i>Reavaliação</i>	<ul style="list-style-type: none"> • Avaliação e refinamento dos protótipos; • Planejamento de novas funcionalidades e adoção de novas tecnologias; • Análise e planejamento da integração entre <i>frontend</i> e <i>backend</i>; • Documentação das avaliações e modificações realizadas.
Desenvolvimento da API principal (<i>Backend</i>)	10ª a 14ª semanas	Implementação e Testes de Validação	<ul style="list-style-type: none"> • Implementação e desenvolvimento dos requisitos no sistema; • Criação do banco de dados, rotas de acessos aos dados e <i>middleware</i> de autenticação e autorização de acesso ao sistema; • Implementação de tratamentos de erros; • Integração entre a API desenvolvida e a API do <i>Google Books</i>; • Integração entre a API desenvolvida e a API do <i>Gemini AI</i>¹⁵. • Desenvolvimento de testes unitários para cada rota do sistema e testes de integração para o fluxo de execução natural do <i>Create, Read, Update and Delete (CRUD)</i> de cada entidade; • Revisão e, caso necessário, atualização dos requisitos do sistema.

¹⁵ Gemini AI - Disponível na URL: <https://gemini.google.com/app?hl=pt-BR>. Acesso em: 28/01/2025.

Desenvolvimento da API de Orientada a Eventos (<i>Backend</i>)	15 ^a a 18 ^a semanas	Implementação e Testes de Validação	<ul style="list-style-type: none"> ● Implementação e desenvolvimento dos requisitos no sistema; ● Implementação de canais de bidirecionalidade; ● Desenvolvimento de testes unitários; ● Revisão e, caso necessário, atualização dos requisitos do sistema.
Desenvolvimento do <i>frontend</i>	18 ^a a 23 ^a semanas	Implementação e Testes de Validação	<ul style="list-style-type: none"> ● Implementação e desenvolvimento dos requisitos no sistema; ● Criação das páginas, janelas ou <i>pop-ups</i> implementando o <i>layout</i> desenvolvido; ● Implementação de autenticação, uso de <i>cookies</i> e de tratamentos de erros; ● Desenvolvimento de testes de sistema para cada funcionalidade do <i>frontend</i>; ● Revisão e, caso necessário, atualização dos requisitos do sistema.
Integração <i>frontend</i> com <i>backend</i>	23 ^a a 25 ^a semanas	Implementação e Testes de Validação	<ul style="list-style-type: none"> ● Implementação e integração do <i>frontend</i> do sistema com as duas APIs desenvolvidas; ● Desenvolvimento de testes de sistema para validar a integração das partes envolvidas; ● Revisão e, caso necessário, atualização dos requisitos do sistema.
Desenvolvimento do <i>Chat</i>	26 ^o a 28 ^o semanas	Implementação e Testes de Validação	<ul style="list-style-type: none"> ● Implementação do <i>chat</i>; ● Desenvolvimento de testes de sistema para o <i>chat</i>; ● Revisão e atualização dos requisitos do sistema conforme necessário.

Conclusão e entrega	28° a 30°	Teste de Validação e Conclusão	<ul style="list-style-type: none"> ● Avaliação final dos testes e do projeto; ● Atualização da documentação do sistema; ● Encerramento formal e entrega da primeira versão do <i>Book Collection</i>.
----------------------------	-----------	--------------------------------	--

Fonte: Elaborado pelo o autor.

Assim, a metodologia proposta assegura uma abordagem estruturada e eficiente para o desenvolvimento do sistema *Book Collection*. Com fases bem definidas e uma revisão constante do progresso, o projeto seguiu bem posicionado para atingir seus objetivos e garantir a funcionalidade desejada.

2.3 Conclusão Metodológica

Neste capítulo, foram detalhadas a organização do documento, a pesquisa e os projetos relacionados, a gestão dos requisitos, o desenvolvimento ágil e o fluxo de execução. Inicialmente, foi estabelecida a estrutura do documento e a importância de um planejamento meticuloso para garantir a clareza e a coesão do trabalho. Em seguida, a análise de projetos semelhantes e a realização de uma pesquisa pública forneceram uma base sólida para identificar e levantar os requisitos do sistema.

A gestão dos requisitos, facilitada pelo uso de ferramentas como o *Jira*, permitiu uma organização eficiente e o acompanhamento das necessidades do projeto. A aplicação do *Scrum Solo* em conjunto com a prática do TDD, assegura o foco, disciplina e adaptabilidade durante o desenvolvimento.

O fluxo de execução foi dividido em fases claras: Pesquisa e Levantamento dos Requisitos, *Design* e Prototipação, Reavaliação, Implementação, e Testes e Validação. Cada fase garantiu que o sistema *Book Collection* fosse desenvolvido de maneira eficiente e alinhada com os objetivos propostos.

No próximo capítulo, será detalhado a *Descrição do Sistema*, explorando suas funcionalidades, tecnologias utilizadas e arquitetura proposta, complementando o trabalho metodológico realizado até aqui.

3 DESCRIÇÃO DO SISTEMA

Este capítulo detalha as informações sobre o sistema proposto para o desenvolvimento deste projeto, denominado *Book Collection*, possuindo como principal finalidade aproximar pessoas do mundo literário e oferecer um ambiente controlado, onde leitores podem gerenciar e organizar suas leituras em sua estante virtual.

3.1 Elicitação dos Requisitos

Esta seção retrata a busca e estudo a respeito dos requisitos elaborados para atender a proposta de ação do projeto desenvolvido, sendo assim, “a elicitação de requisitos, é o processo de buscar, descobrir, adquirir, elaborar requisitos para sistemas baseados em computador” (Zowghi & Coulin, 2005). Apresentando assim, quais as funcionalidades poderão ou deverão ser produzidas e implementadas ao sistema. Sendo assim, o ponto inicial de partida para o desenvolvimento de um *software*, contribuindo para as entregas incrementais e possíveis mudanças no decorrer do projeto.

Os requisitos constituem um conjunto de restrições e descrições a respeito das funcionalidades de um sistema, sendo assim, frequentemente os requisitos são classificados em requisitos funcionais e não funcionais. Os requisitos funcionais, como definido por Sommerville (2011), “são declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações”. Por outro lado, os requisitos não funcionais que não estão diretamente relacionados aos serviços que o sistema disponibiliza, representando muitas vezes restrições de desempenho, proteção ou disponibilidade do sistema como um todo.

Nesse sentido, os requisitos analisados e desenvolvidos para com este sistema, seguem a seguinte descrição de prioridade, a fim de elencar e classificar o grau de importância do requisito para o funcionamento do mesmo:

- **Essencial:** Indica que o requisito é imprescindível para o funcionamento do sistema;
- **Importante:** Indica que o requisito não é essencial para o funcionamento do sistema, contudo seu funcionamento, sem implementação do requisito, se torna insatisfatório;
- **Desejável:** Indica que o requisito não compromete as funcionalidades

básicas do sistema, podendo funcionar de forma satisfatória sem a implementação deste.

Sendo assim, o quadro a seguir apresenta os requisitos funcionais elaborados para a implementação do sistema *Book Collection*.

Quadro 05 - Requisitos Funcionais

Requisito	Objetivo	Descrição	Prioridade
RF01	Cadastrar usuários	O sistema deve permitir que os usuários possam criar contas no sistema, sendo assim, se cadastrando no sistema.	Essencial
RF02	Efetuar <i>login</i>	O sistema deve permitir que os usuários cadastrados possam fazer o <i>login</i> para acessar o sistema.	Essencial
RF03	Efetuar <i>logout</i>	O sistema deve permitir que os usuários cadastrados possam fazer o <i>logout</i> , (sair do sistema), assim que lhes for conceitual.	Essencial
RF04	Manter livros	O sistema deve apresentar para os usuários cadastrados dados e informações sobre os livros pesquisados e apresentados no sistema.	Essencial
RF04	Manter estante virtual	O sistema deve permitir que os usuários cadastrados possam adicionar e remover livros à sua estante virtual.	Essencial
RF05	Categorizar leituras	O sistema deve permitir que os usuários cadastrados possam organizar suas leituras em coleções padrão ou criar coleções personalizadas.	Essencial
RF06	Visualizar leitura	O sistema deve permitir que os usuários cadastrados possam visualizar os livros, bem como as informações deste, presentes em sua estante.	Essencial
RF07	Gerenciamento	O sistema deve permitir que os	Essencial

	momentâneo a leitura	usuários cadastrados no sistema possam atualizar o estado de sua leitura, bem como avaliar e comentar o progresso da leitura.	
RF08	Pesquisar e visualizar informações sobre os livros	O sistema deve permitir que os usuários cadastrados possam pesquisar e visualizar as informações referentes a esse livro.	Essencial
RF09	Recomendar livros	O sistema deve recomendar livros aos usuários com base em seus padrões de leitura, considerando fatores como gêneros mais lidos, autores favoritos, livros avaliados e pesquisas realizadas na plataforma.	Essencial
RF10	Pesquisar e visualizar informações sobre os outros usuários	O sistema deve permitir que os usuários cadastrados possam pesquisar e visualizar informações referentes a outros usuários.	Essencial
RF11	Avaliação de leituras	O sistema deve permitir que os usuários cadastrados possam realizar avaliações pessoais sobre os livros já lidos.	Importante
RF12	Manter perfis seguindo	O sistema deve permitir que os usuários cadastrados possam seguir outros perfis de usuários.	Importante
RF13	Manter perfis seguidores	O sistema deve permitir que os usuários cadastrados possam ser seguidos por outros perfis de usuários.	Importante
RF14	Publicar notas de texto	O sistema deve permitir que os usuários cadastrados possam publicar notas em um mural de publicações.	Importante
RF15	Moderar conteúdos	O sistema deve realizar a moderação de conteúdos publicados pelos usuários, garantindo a conformidade com as políticas da plataforma.	Importante

RF16	Notificações de eventos	O sistema deve exibir notificações para os usuários cadastrados, informando-os sobre os resultados de suas ações e atividades relacionadas aos perfis que os seguem.	Importante
RF17	Manter <i>Chat</i>	O sistema deve permitir que os usuários cadastrados possam interagir com outros perfis enviando e recebendo mensagens.	Essencial

Fonte: Elaborado pelo o autor.

O Quadro 05 apresentou as implementações dos requisitos funcionais necessários para o sistema *Book Collection*. Agora, o quadro adiante detalha os requisitos não funcionais, que descrevem as características e os critérios de qualidade que o sistema deve atender, especificando como ele deve operar para realizar suas funcionalidades de forma eficiente e consistente.

Quadro 06 - Requisitos não funcionais

Requisito	Objetivo	Descrição	Prioridade
RNF01	Usabilidade	O sistema deve ser intuitivo e de fácil utilização, garantindo que os usuários possam acessar os recursos de forma rápida e eficiente.	Essencial
RNF02	Ambiente <i>Web</i>	O sistema deve funcionar em um ambiente <i>web</i> .	Essencial
RNF03	<i>Web design</i> responsivo	O sistema deve ser acessível em dispositivos de diferentes tamanhos de tela.	Importante
RNF04	Comunicação assíncrona	O sistema deve incorporar funcionalidades assíncronas, garantindo interações fluidas e contínuas entre os usuários.	Essencial
RNF05	Segurança	O sistema deve garantir que somente usuários autenticados possam acessar dados confidenciais, além de	Essencial

		implementar mecanismos de criptografia para proteger essas informações.	
--	--	---	--

Fonte: Elaborado pelo o autor.

Assim, os quadros de requisitos funcionais e não funcionais apresentados fornecem uma visão geral das necessidades do sistema *Book Collection*. Esses requisitos serão detalhados mais profundamente nas próximas seções de *Modelagem, Funcionalidades e Interfaces do Sistema*, onde serão explorados de forma prática e aplicada. A seguir, será abordada a seção de *Tecnologias Utilizadas*, que descreve as ferramentas e *frameworks* que suportam o desenvolvimento do sistema.

3.2 Tecnologias Utilizadas

Nesta seção, serão exploradas as tecnologias fundamentais que desempenharam um papel crucial no desenvolvimento do sistema. A escolha criteriosa das tecnologias é primordial para garantir a eficácia, segurança e escalabilidade do sistema. Durante esta seção, serão discutidas as ferramentas que foram empregadas para implementar as diversas partes do sistema. Em particular, serão destacados o uso do *React*¹⁶ e *TypeScript*¹⁷ para o desenvolvimento da interface do usuário, o *PostgreSQL* como o banco de dados principal, e o *Prisma*¹⁸ como uma ferramenta de *Object-Relational Mapping* (ORM).

3.2.1 React JS

React é uma biblioteca *JavaScript* de código aberto desenvolvida pelo *Facebook* em 2011, amplamente utilizada para a criação de interfaces de usuário interativas e dinâmicas. De acordo com a documentação oficial, “*React* é uma biblioteca *JavaScript* para renderizar interfaces de usuário” (React, 2023). A principal proposta do *React* é facilitar o desenvolvimento de aplicações *web* com interfaces complexas, dividindo-as em componentes, “um componente *React* é uma função *JavaScript* que você pode adicionar marcação” (React, 2023).

Uma das características centrais do *React* é o *Document Object Model Virtual* (*Virtual DOM*), que permite atualizações eficientes da interface em resposta a mudanças de estado, resultando em uma experiência de usuário mais rápida e responsiva. O *React* também

¹⁶ React - Disponível na URL: <https://react.dev>. Acesso em 23/04/2024.

¹⁷ TypeScript - Disponível na URL: <https://www.typescriptlang.org>. Acesso em 23/04/2024.

¹⁸ Prisma ORM - Disponível na URL: <https://www.prisma.io>. Acesso em 23/04/2024.

adota um fluxo de dados unidirecional, facilitando o gerenciamento do estado e tornando o desenvolvimento de aplicativos complexos mais previsível e simples de manter.

Com uma comunidade ativa e um ecossistema robusto de ferramentas e bibliotecas, o *React* continua a evoluir e se adaptar às demandas do desenvolvimento *web* moderno. Sua popularidade é evidenciada pelo grande número de empresas e desenvolvedores que o utilizam em projetos de todos os tamanhos e complexidades ao redor do mundo.

3.2.2 Typescript

Desde o seu lançamento em 2012 pela *Microsoft*, o *TypeScript* tem ganhado destaque como uma linguagem de programação que combina a flexibilidade do *JavaScript* com recursos adicionais de tipagem estática. Como descrito na documentação oficial, “o *TypeScript* é uma linguagem fortemente tipada baseada em *JavaScript*” (Typescript, 2024), o que significa que oferece a capacidade de especificar tipos de dados para variáveis, parâmetros de função e retornos de função.

Essa abordagem fortemente tipada confere maior segurança e robustez ao código, ajudando a detectar erros de tipo durante a fase de desenvolvimento, em vez de em tempo de execução. Além disso, o *TypeScript* proporciona um ambiente de desenvolvimento mais rico, com recursos como autocompletar, refatoração segura e navegação de código aprimorada, que aumentam a produtividade dos desenvolvedores. Embasando a definição, como é apresentado na documentação oficial:

“O objetivo do TypeScript é ser um verificador de tipo estático para programas JavaScript - em outras palavras, uma ferramenta que é executada antes da execução do código (estático) e garante que os tipos do programa estejam corretos (verificado de tipo).”

Uma das principais vantagens do *TypeScript* é a sua capacidade de escalar em projetos grandes e complexos, onde a manutenção do código e a colaboração entre equipes são essenciais. A tipagem estática ajuda a documentar o código de forma mais clara e precisa, facilitando a compreensão e a manutenção do mesmo ao longo do tempo. Além disso, o *TypeScript* é amplamente suportado pela comunidade de desenvolvedores e possui uma grande variedade de bibliotecas e *frameworks* compatíveis, o que o torna uma escolha popular para o desenvolvimento de aplicações *web* modernas e escaláveis.

3.2.3 Prisma ORM

O *Prisma* é uma ferramenta de mapeamento objeto-relacional (ORM) de código aberto, amplamente utilizada por desenvolvedores para criar APIs modernas com tecnologias

como *Node.js*¹⁹ e *TypeScript*. Como um ORM, o *Prisma* simplifica significativamente a interação com bancos de dados relacionais, permitindo que os desenvolvedores trabalhem com modelos de dados em um nível mais abstrato, usando linguagens de programação familiares. Como elucidado na documentação oficial:

“O *Prisma ORM* torna mais fácil para os desenvolvedores raciocinarem sobre suas consultas ao banco de dados, fornecendo uma API limpa e de tipo seguro para enviar consultas ao banco de dados que retorna objetos *JavaScript* antigos e simples.” .

O *Prisma* é composto por várias partes essenciais, cada uma desempenhando um papel importante no processo de desenvolvimento de aplicativos. O *Prisma Client* é uma das principais funcionalidades, fornecendo um construtor de consultas gerado automaticamente e com tipo seguro para *Node.js* e *TypeScript*. Isso significa que os desenvolvedores podem escrever consultas de banco de dados de forma intuitiva, aproveitando a segurança de tipo fornecida pelo *TypeScript*.

Além do *Prisma Client*, o *Prisma* também oferece o *Prisma Migrate*, um sistema de migração de dados que simplifica o processo de evolução do esquema do banco de dados ao longo do tempo. Com o *Prisma Migrate*, os desenvolvedores podem gerenciar facilmente alterações no esquema do banco de dados, garantindo que as atualizações sejam aplicadas de forma segura e eficiente.

Por fim, o *Prisma Studio* é uma ferramenta gráfica poderosa que permite visualizar e editar dados diretamente no banco de dados. Com uma interface de usuário intuitiva, o *Prisma Studio* simplifica a tarefa de explorar e manipular dados, tornando o processo de desenvolvimento mais eficiente e produtivo.

No geral, o *Prisma* é uma ferramenta versátil e poderosa que simplifica o desenvolvimento de aplicativos modernos, oferecendo aos desenvolvedores uma maneira mais rápida, segura e eficiente de interagir com bancos de dados relacionais.

3.2.4 Express JS

O *Express*²⁰ é um *framework* minimalista para *Node.js*, projetado para construir aplicações *web* e APIs de forma rápida e eficiente. Criado em 2010, o *Express* tem sido amplamente adotado pela comunidade de desenvolvedores por sua simplicidade e flexibilidade. Ele fornece uma camada robusta de recursos para aplicações *web*, sem impor

¹⁹ Node.Js. Disponível em: <https://nodejs.org/pt>. Acesso em: 23/04/2024.

²⁰ Express - Disponível na URL: <https://expressjs.com>. Acesso em: 10/08/2024.

uma estrutura rígida, permitindo que os desenvolvedores organizem seu código da maneira que acharem mais conveniente. Assim, segundo a documentação oficial:

“*Express* é uma estrutura de roteamento leve e flexível com recursos básicos e mínimos que devem ser aumentados por meio do uso do módulo *Express middleware*.”.

Com o *Express*, é possível gerenciar rotas *Hypertext Transfer Protocol* (HTTP), manipular solicitações e respostas, e integrar *middlewares* para adicionar funcionalidades, como autenticação, *login*, e manipulação de erros. Como apresentado na documentação oficial, “com uma infinidade de métodos utilitários HTTP e *middleware* à sua disposição, criar uma API robusta é rápido e fácil”. Sua compatibilidade com o ecossistema *Node.js* torna uma escolha natural para desenvolvedores que buscam uma solução leve e eficaz para criar servidores *web* e APIs RESTful.

No contexto do desenvolvimento do sistema *Book Collection*, o *Express* se mostra particularmente útil quando combinado com tecnologias como *TypeScript* e *Prisma ORM*. O uso do *TypeScript* com *Express* oferece tipagem estática, o que contribui para um código mais seguro e fácil de manter, prevenindo erros comuns durante o desenvolvimento. Já o *Prisma ORM* facilita a interação com o banco de dados *PostgreSQL*, permitindo a manipulação de dados de forma eficiente e tipada.

Em conjunto, essas tecnologias proporcionam um ambiente de desenvolvimento ágil e seguro para a criação da API do *Book Collection*, garantindo que o sistema seja escalável, robusto e mantenha um desempenho de alto nível.

3.2.5 *Socket.io*

O *Socket.io*²¹ é uma biblioteca amplamente utilizada para comunicação em tempo real entre clientes e servidores, baseada no protocolo *WebSocket*. Essa tecnologia permite a troca de dados de forma bidirecional e assíncrona, garantindo uma comunicação eficiente e de baixa latência. Diferente das requisições HTTP convencionais, que seguem um fluxo de requisição e resposta, o *Socket.io* possibilita a manutenção de uma conexão persistente, permitindo o envio e recebimento de mensagens instantaneamente, sem a necessidade de múltiplas requisições ao servidor.

No contexto do *Book Collection*, o *Socket.io* desempenha um papel fundamental nas funcionalidades que demandam interação assíncrona e comunicação bidirecional. Entre os principais cenários de aplicação, destacam-se aqueles que envolvem notificações instantâneas, trocas de mensagens entre usuários e atualizações dinâmicas no sistema. Essas

²¹ Socket.IO. Disponível em: <https://socket.io>. Acesso em: 05/09/2024.

aplicações garantem uma experiência mais fluida e interativa, proporcionando um ambiente dinâmico e conectado para os usuários.

Nas seções seguintes, serão apresentadas as funcionalidades e arquiteturas onde essa ferramenta será implementada no sistema, detalhando como essa tecnologia contribuirá para a eficiência e responsividade do *Book Collection*.

3.2.6 Google Books API

A *API do Google Livros* é uma poderosa ferramenta disponibilizada pelo *Google*, projetada para fornecer acesso a uma vasta quantidade de informações sobre livros publicados em todo o mundo. Essa API oferece uma ampla gama de recursos para desenvolvedores, permitindo que eles integrem dados relacionados a livros em suas próprias aplicações.

O *Google Books API* desempenha um papel fundamental no desenvolvimento de sistemas como o *Book Collection*, oferecendo acesso a dados como detalhes dos livros, resumos, capas, classificações e muito mais. Como embasado na documentação oficial:

“O Google Livros é nosso esforço para tornar o conteúdo de livros mais detectável na *Web*. Com a API do Google Livros, seu aplicativo pode realizar pesquisas de texto completo e recuperar informações sobre livros, visibilidade e disponibilidade de e-books. ”.

A API do *Google Books* oferece acesso a dados e informações sobre livros, com recursos tanto públicos, quanto privados. Os recursos públicos incluem informações detalhadas sobre livros, como dados de publicação e sinopses; enquanto os recursos privados, como a criação de estantes e a leitura de livros dentro do sistema, requerem autenticação e autorização dos usuários. No contexto do *Book Collection*, serão utilizados apenas os recursos públicos da API, conhecidos como "volumes". Estes são definidos como: “os dados hospedados pelo Google Livros sobre um livro ou uma revista, sendo o principal recurso da *API Books*”.

A integração com a API *Google Books* pode ser feita de duas formas: através de uma “chave da API”, que identifica o sistema ao acessar o *Google Books*, ou um “*token OAuth 2.0*”, que pode ser usado para acessar recursos privados. Além disso, essa API usa verbos HTTP para realizar a solicitação dos recursos, conforme apresentado na documentação:

“Nas APIs RESTful do Google, o cliente especifica uma ação usando um verbo HTTP, como POST, GET, PUT ou DELETE. [...] Como todos os recursos da API têm URIs exclusivos acessíveis por HTTP, a REST

permite o armazenamento em cache dos dados e é otimizada para funcionar na infraestrutura distribuída da Web.”.

Com uma vasta biblioteca de dados e recursos abrangentes, a *Google Books API* é uma ferramenta essencial para o *Book Collection*, permitindo a integração de informações detalhadas sobre livros e ampliando significativamente o alcance e a utilidade do sistema para os usuários.

3.2.7 Gemini AI

O *Gemini AI* é um modelo de inteligência artificial desenvolvido pelo *Google*, pertencente à categoria de *Large Language Models* (LLMs), ou Modelos de Linguagem de Grande Escala. Esses modelos representam um avanço significativo na IA, sendo redes neurais treinadas com vastas quantidades de dados textuais, o que lhes permite realizar tarefas complexas, como compreensão e geração de linguagem natural, análise semântica, tradução de idiomas e inferências contextuais.

Segundo a IBM (2025), os LLMs são modelos de base de IA projetados para processar e gerar texto de maneira semelhante à humana. Como destacado no estudo:

“LLM são uma categoria de modelos de base de IA treinados com imensas quantidades de dados, tornando-os capazes de entender e gerar linguagem natural e outros tipos de conteúdo para executar uma grande variedade de tarefas.” (IBM, 2025)

Diferente de sistemas tradicionais baseados em regras, os LLMs utilizam aprendizado profundo para identificar padrões complexos, o que lhes confere maior adaptabilidade e precisão na geração de respostas. Ainda segundo a IBM (2025), “esses modelos foram projetados para entender e gerar texto como um humano, além de outras formas de conteúdo, com base na vasta quantidade de dados utilizados para treiná-los.”

O *Gemini AI*, como um dos principais representantes dessa tecnologia, integra esses avanços para oferecer respostas mais naturais e contextualmente adequadas, consolidando-se como uma ferramenta poderosa na evolução da inteligência artificial generativa.

Ao longo do desenvolvimento do sistema, o modelo de linguagem *Gemini AI* foi utilizado como uma ferramenta fundamental. Como o próprio modelo se descreve, “*Eu me considero uma ferramenta de processamento de linguagem natural, treinada para auxiliar na geração de texto e na compreensão da linguagem humana*”, o *Gemini AI* demonstrou ser capaz de gerar informações coerentes e relevantes, auxiliando nas funcionalidades que lhe foram aplicadas.

No *Book Collection*, o *Gemini AI* foi incorporado para desempenhar um papel estratégico em diversas funcionalidades do sistema. Sua aplicação se estende desde a moderação de conteúdo, filtrando automaticamente conteúdos identificados como inapropriados, até a complementação de informações, garantindo descrições mais completas para os livros cadastrados.

Além disso, a Inteligência Artificial (IA) é utilizada na recomendação personalizada de leituras, ajustando sugestões com base nos hábitos e preferências dos usuários ao longo do tempo. Com essa integração, a inteligência artificial contribui significativamente para a experiência na plataforma, tornando-a mais dinâmica, segura e adaptável. A seguir, a forma como o *Book Collection* consome os serviços do *Gemini AI* será detalhada na seção de *Funcionalidades*, onde serão explorados os métodos de integração e as interações entre o sistema e a API da IA.

Do ponto de vista técnico, o desenvolvimento de sistemas com presença de IA, como no caso do *Book Collection*, envolve a integração de modelos de *machine learning* com arquiteturas modernas. Entretanto, o *Gemini AI* pode ser acessado via API, permitindo que aplicações web e móveis incorporem suas funcionalidades sem a necessidade de um treinamento de modelo próprio. Além disso, sua escalabilidade permite que a IA processe grandes volumes de dados em tempo real, tornando-se uma solução viável para aplicações que exigem alto desempenho e adaptabilidade.

Com essas características, a presença da inteligência artificial em sistemas modernos se torna cada vez mais essencial. A utilização de modelos como o *Gemini AI* não apenas aprimora a usabilidade e segurança das plataformas, mas também proporciona novas possibilidades de automação, análise de dados e personalização da experiência do usuário. No caso do *Book Collection*, a IA desempenha um papel estratégico, transformando a forma como os usuários interagem com os livros e com a comunidade literária, tornando o sistema mais eficiente, interativo e inteligente.

3.2.8 PostgreSQL

O *PostgreSQL*²², também conhecido como *Postgres*, é um Sistema Gerenciador de Banco de Dados (SGBD) desenvolvido a partir do projeto POSTGRES, originário da Universidade da Califórnia em Berkeley, no ano de 1986. Inicialmente denominado *Postgres 95* em 1994, recebeu o nome *PostgreSQL* em 1995 para refletir sua crescente maturidade e funcionalidades avançadas. Hoje, o *PostgreSQL* é amplamente reconhecido como um dos mais poderosos e sofisticados sistemas de banco de dados objeto-relacional disponíveis.

²² PostgreSQL - Disponível na URL: <https://www.postgresql.org>. Acesso em 23/04/2024.

Segundo a documentação oficial, o “*PostgreSQL* é uma poderosa solução de banco de dados objeto-relacional”, oferecendo uma ampla gama de recursos avançados, incluindo suporte para tipos de dados complexos, transações com Atomicidade, Consistência, Isolamento e Durabilidade (ACID), integridade referencial, gatilhos (*triggers*), procedimentos armazenados, entre outros. Esses recursos fazem do *PostgreSQL* uma escolha ideal para uma variedade de aplicações, desde pequenos projetos de código aberto até grandes sistemas corporativos.

Uma das características distintivas do *PostgreSQL* é sua natureza de código aberto, o que significa que seu código-fonte está disponível gratuitamente para qualquer pessoa interessada em examiná-lo, modificá-lo ou contribuir para o seu desenvolvimento. O *PostgreSQL* é mantido e coordenado pelo *PostgreSQL Global Development Group*, uma comunidade global de desenvolvedores voluntários que trabalham para melhorar e aprimorar continuamente o sistema, sendo então considerado, “o mais avançado banco de dados de código aberto disponível em qualquer lugar.”.

Devido a sua confiabilidade, desempenho e conformidade com padrões, o *PostgreSQL* ganhou uma reputação de ser um dos bancos de dados mais avançados e confiáveis disponíveis atualmente. Sua adoção é vista em uma ampla gama de aplicações e setores, desde *startups* e empresas de tecnologia até grandes corporações e instituições governamentais. Em resumo, o *PostgreSQL* é uma escolha sólida para qualquer projeto que exija um banco de dados robusto, flexível e escalável.

3.2.9 Redis

O *Redis*²³ é um banco de dados de código aberto, usado principalmente como um armazenamento de estrutura de dados na memória, que pode ser utilizado como banco de dados, cache e *message broker*. Desenvolvido por Salvatore Sanfilippo e lançado em 2009, o *Redis* é conhecido por sua alta performance e suporte a diversos tipos de estruturas de dados. Segundo a documentação oficial:

“Redis é um servidor de estrutura de dados de código aberto. Ele pertence à classe de bancos de dados NoSQL conhecidos como armazenamentos de chave/valor. Chaves são identificadores exclusivos, cujo valor pode ser um dos tipos de dados que o Redis suporta.”.

Em sistemas modernos, o *Redis* é amplamente utilizado para melhorar a performance e escalabilidade das aplicações, principalmente devido à sua capacidade de armazenar dados na memória, o que permite acesso extremamente rápido às informações. Dessa forma, como apresentado na documentação oficial:

²³ Redis - Disponível na URL: <https://redis.io/docs/latest/>. Acesso em: 28/04/2024.

“O Redis é famoso por ser um banco de dados extremamente rápido. Essa velocidade vem do fato de que ele armazena e serve todos os dados da memória em vez do disco. O Redis é durável, então seus dados serão persistidos, mas todas as leituras serão de uma cópia dos dados mantidos na memória.”

No contexto do sistema *Book Collection*, o *Redis* será utilizado para armazenar as informações dos usuários, como dados de seu perfil e das suas atividades no uso do sistema, garantindo acesso rápido, eficiente a esses dados, além de atribuir maior escalabilidade para a aplicação. Isso é particularmente útil para operações frequentes, como autenticação de usuários, armazenamento de sessões e gerenciamento de cache, onde a velocidade de acesso aos dados é crítica para a experiência do usuário.

Além disso, para assegurar a consistência dos dados, o *Redis* emprega mecanismos de persistência, como *Redis Database* e *Append-Only File*, permitindo a recuperação dos dados em caso de falhas. A utilização de replicação e *clusters* também contribui para garantir a integridade das informações, prevenindo perda de dados e melhorando a disponibilidade do sistema. Por fim, a configuração de políticas de expiração de chaves permite que o cache seja atualizado conforme necessário, evitando dados obsoletos e garantindo uma sincronização eficiente com o banco de dados principal.

Ao adotar o *Redis*, o sistema se beneficia de um tempo de resposta reduzido e de uma maior capacidade de escalabilidade, permitindo que o *Book Collection* gerencie um grande volume de usuários e operações com eficiência.

3.3 Modelagem

Esta seção abordará um estudo e análise dos requisitos do sistema por meio da elaboração e planejamento estrutural do sistema, por meio de diagramas e estudos de caso. Segundo Guedes (2011), “um sistema de informação precisa ter uma documentação extremamente detalhada, precisa e atualizada para que possa ser mantido com facilidade, rapidez e correção.”

Para tanto, o estudo nessa seção corresponde a criação documental abrangendo uma definição estrutural sobre o que sistema *Book Collection*, resultando assim em um modelo de *software*, “um modelo de software captura uma visão de um sistema físico, é uma abstração do sistema com um certo propósito, como descrever aspectos estruturais ou comportamentais do *software*” (Guedes, 2011). Posteriormente, serão exploradas em detalhes as modelagens do sistema, dos dados e da arquitetura.

3.3.1 Modelagem do sistema

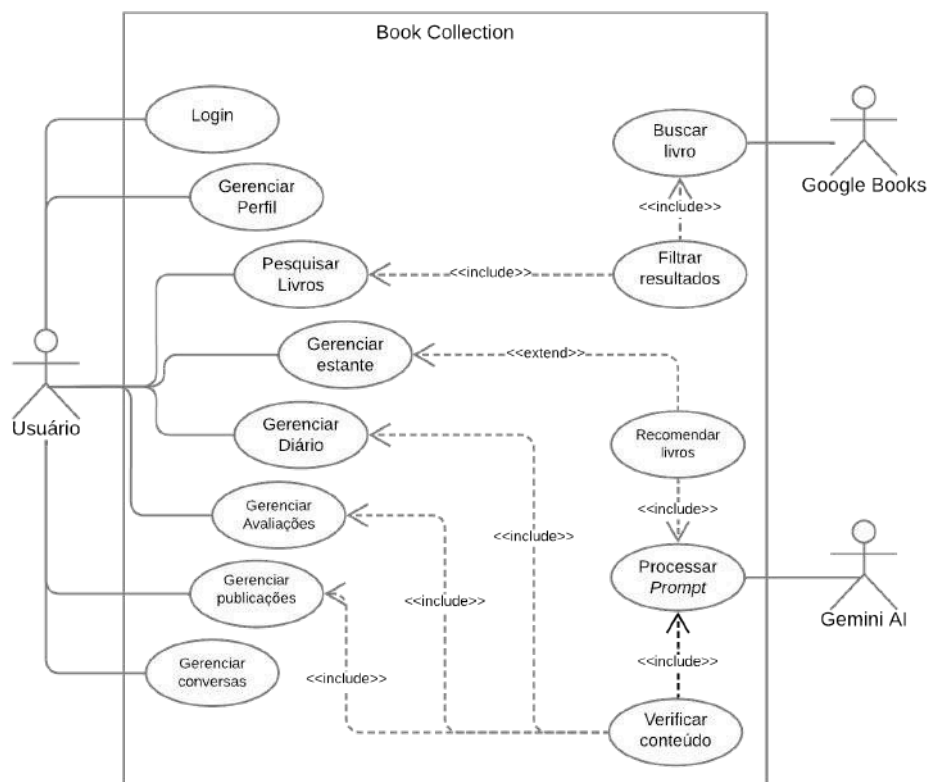
Esta seção abordará a análise da modelagem do sistema, empregando diagramas para representar sua estrutura, comportamento e interações. Os diagramas são essenciais para compreender e visualizar as diversas funcionalidades e aspectos do sistema. Segundo Fowler (2005), “os diagramas UML desempenham um papel fundamental na modelagem de sistemas de software, oferecendo uma representação visual das diferentes perspectivas e aspectos do sistema”. A utilização de diagramas de casos de uso, casos de uso e diagramas de atividades proporcionará uma visão abrangente do funcionamento do sistema, desde as interações dos usuários até os processos internos.

3.3.1.1 Diagrama de Caso de Uso

O Diagrama de Caso de Uso é um diagrama de alto nível do conjunto de casos de uso, possibilitando a demonstração das possíveis interações descritas na seção dos requisitos. Segundo Sommerville (2011), “os casos de uso identificam as interações individuais entre o sistema e seus usuários ou outros sistemas”. A seguir, na Figura 03, apresenta-se um Diagrama de Caso de Uso.

A Figura 03 ilustra as principais interações de um usuário com o sistema *Book Collection*. Entre as funcionalidades destacadas estão o *login*, o gerenciamento de perfil (incluindo publicações, seguidores e dados pessoais), a organização da estante virtual, a criação de diários de leitura com comentários e anotações e a avaliação de livros. Também é exibida a relação com a API do *Google Books*, responsável por fornecer informações dos livros selecionados e pesquisados pelos usuários. Além de fomentar a participação do *Google Gemini AI*, responsável, nesse cenário, pela verificação e validação de conteúdo, inibindo a publicação de conteúdo malicioso, além de auxiliar no retorno de recomendação de livros para o usuário.

Figura 03 - Diagrama de Caso de Uso



Fonte: Elaborado pelo o autor.

Com base na Figura 03 é possível visualizar de forma clara a participação de cada ator no sistema e como suas ações influenciam diretamente o fluxo das operações. Na sequência, serão detalhadas as tabelas correspondentes aos casos de uso, oferecendo uma visão estruturada e compreensiva das interações no sistema.

3.3.1.2 Caso de uso

Os Casos de Uso são narrativas e descrições em texto utilizadas para apresentar os requisitos de um sistema, segundo Fowler (2005), “os Casos de Uso servem para descrever as interações típicas entre os usuários de um sistema e o próprio sistema, fornecendo uma narrativa de como o sistema é utilizado”. Os principais casos de uso do sistema *Book Collection* são Gerenciar Estante Virtual, Gerenciar Leitura, Gerenciar Publicações e Pesquisar Livros.

A gerência da estante virtual permite que os usuários organizem seus livros em diferentes coleções, categorizando-os conforme seu *status* de leitura ou preferências pessoais. Essa funcionalidade é detalhada no Quadro 07, que apresenta as interações envolvidas nesse processo.

Quadro 07 - Caso de uso: Gerência de estante virtual

UC_1	Gerenciar Estante Virtual
Autores	Usuário.
Pré-condições	O usuário possui cadastro no sistema e está logado no sistema.
Pós-condições	Permitir o usuário visualizar e editar sua estante virtual.
Cenário típico	<ol style="list-style-type: none"> 1. O usuário acessa a tela inicial do sistema; 2. O usuário acessa a página de “Estante” presente no menu lateral; 3. O sistema redireciona para a estante de coleções de livros do usuário; 4. O sistema exibe a lista de livros e as suas coleções; 5. O usuário escolhe uma coleção de sua estante virtual, clicando em “Ver Coleção”, caso seja uma coleção padrão (Lido, Desejo ler, Lendo, Avaliados), ou clicando no componente da coleção desejada;
Cenários alternativos	<p>4.a Criar uma nova coleção:</p> <ol style="list-style-type: none"> 1. O usuário clica no componente de criação de uma coleção; 2. O sistema apresentará um formulário aparecerá no fim da página; 3. Usuário preenche as informações e confirma a criação; 4. Após a confirmação, o sistema apresenta uma notificação, informando que a coleção foi criada. <p>5.a Adicionar um livro a coleção:</p> <ol style="list-style-type: none"> 1. Ao acessar a página da coleção, duas funcionalidades serão apresentadas ao usuário: “Adicionar Livro” e “Compartilhar”; 2. O usuário clica em “Adicionar Livro”; 3. O sistema redireciona para a página de livros; 4. O usuário escolhe o livro e clicar no símbolo de “Marca-páginas”, escolhe a coleção; 5. Por fim, o sistema confirma a adição do livro à coleção por meio de uma notificação. <p>5.b Remover um livro da coleção:</p> <ol style="list-style-type: none"> 1. O usuário escolhe o livro e clicar no componente definido como “Ver”; 2. O sistema redireciona o usuário para a página de detalhes do livro; 3. O ícone de “Marca-páginas” apresentará como preenchido, ao clicar, o usuário poderá ver as coleções onde o livro está alocado; 4. O usuário escolhe de qual coleção o livro será removido e clica

	<p>em “Remover”;</p> <p>5. Após a confirmação, o sistema apresentará uma notificação de confirmação da remoção.</p> <p>5.c Avaliar livro:</p> <ol style="list-style-type: none"> 1. O usuário escolhe o livro e clicar no componente definido como “Ver”; 2. O sistema redireciona o usuário para a página de detalhes do livro; 3. O usuário clica no ícone de avaliação, definindo como um balão de comentário; 4. Após concluir a avaliação, o sistema notifica o usuário e o livro é automaticamente salvo na coleção padrão denominada “Avaliados”.
--	---

Fonte: Elaborado pelo o autor.

A gerência da leitura permite que o usuário registre o progresso de leitura de um livro, acompanhando suas anotações, a quantidade de páginas lidas e suas impressões ao longo da leitura. O Quadro 08 apresenta a estrutura dessa funcionalidade no sistema.

Quadro 08 - Caso de uso: Gerência de leitura

UC_2	Gerenciar Leitura
Autores	Usuário e <i>Gemini AI</i> .
Interessados	Usuário.
Pré-condições	O usuário possui cadastro no sistema e está logado no sistema.
Pós-condições	Permitir o usuário visualizar e editar seu estado de uma leitura.
Cenário típico	<ol style="list-style-type: none"> 1. O usuário acessa a tela inicial do sistema; 2. O usuário seleciona o livro desejado na seção “Leituras atuais”; 3. O usuário clica no componente definido como “Gerir leitura”; 4. O sistema apresenta uma janela contendo os últimos estados da leitura; 5. O usuário clica em “Novo status”; 6. O sistema apresenta um formulário, permitindo assim que o usuário adicione um novo estado de leitura; 7. O usuário após concluir pode clicar em “Concluir”; 8. O conteúdo do registro de leitura será validado pelo sistema; 9. O sistema notifica o usuário, confirmando a criação e listando

	os registros de leitura do sistema.
Cenários alternativos	<p>1. a Primeira alternativa ao passo 1:</p> <ol style="list-style-type: none"> 1. O usuário acessa a página de “Perfil” presente no menu lateral; 2. O sistema redireciona para a página de perfil do usuário; 3. Retorna ao cenário principal no passo 4. <p>1. b Segunda alternativa ao passo 1:</p> <ol style="list-style-type: none"> 1. O usuário presente na página de “Histórico de leituras”; 2. O usuário escolhe o livro desejado e clica no componente definido como “Diário”; 3. Retorna ao cenário principal no passo 4. <p>4. a Editar registro de leitura:</p> <ol style="list-style-type: none"> 1. O sistema apresenta a lista de registros de leitura; 2. O usuário pode editar o último registro informado, clicando no ícone de edição, definido como um “Lápis”; 3. O sistema apresenta um formulário contendo as informações do registro de leitura, permitindo que o usuário edite as informações que deseje; 4. Retorna ao cenário principal no passo 7. <p>4. b Remover registro de leitura:</p> <ol style="list-style-type: none"> 1. O sistema apresenta a lista de registros de leitura; 2. O usuário pode remover o último registro informado, clicando no ícone de remoção, definido como uma “Lixeira”; 3. Retorna ao cenário principal no passo 4. <p>7. a Validação de conteúdo malicioso:</p> <ol style="list-style-type: none"> 1. O <i>backend</i> do sistema, capta a conteúdo do registro e por meio de um <i>prompt</i> de comando, aciona a <i>Gemini AI</i>, solicitando uma análise do conteúdo; 2. O sistema, com base em um retorno positivo da Inteligência Artificial, irá retornar uma mensagem de sucesso e criará o registro de leitura; 3. Caso o registro infira algum conteúdo malicioso ou discriminatório, o sistema irá inibir o registro de leitura, retornando um alerta com o erro, solicitando a alteração do conteúdo, além de armazenar a infração no histórico de denúncias do usuário.

Fonte: Elaborado pelo o autor.

A funcionalidade de gerência de publicações está diretamente relacionada à camada social do *Book Collection*, permitindo que os usuários compartilhem citações, reflexões e

interajam com conteúdos postados por outros membros. O Quadro 09 ilustra como essas interações ocorrem dentro da plataforma.

Quadro 09 - Caso de uso: Gerência de publicação

UC_3	Gerenciar publicação
Autores	Usuário e <i>Gemini AI</i> .
Interessados	Usuário.
Pré-condições	O usuário possui cadastro no sistema e está logado no sistema.
Pós-condições	Permitir o usuário visualizar, criar, remover, curtir e comentar uma publicação de texto.
Cenário típico	<ol style="list-style-type: none"> 1. O usuário acessa a tela inicial do sistema; 2. O usuário acessa a página de “Citações” presente no menu lateral; 3. O sistema redireciona para a página de publicação de notas; 4. O sistema exibe a lista de notas publicadas recentemente; 5. O usuário acessa o campo de texto “No que está pensando?”; 6. O usuário informa a nota de texto; 7. O conteúdo da publicação é validada pelo sistema; 8. O usuário publica a sua nota de texto, sendo assim visível para os seus amigos.
Cenários alternativos	<p>4. a Remover uma citação publicada:</p> <ol style="list-style-type: none"> 1. O usuário acessa a citação desejada, desde que tenha sido publicado por ele; 2. O usuário clica no símbolo de “Lixeira” que será visível somente nas publicações criadas por ele; 3. O sistema informa uma notificação de confirmação; 4. Após a confirmação, o sistema remove a citação do histórico do usuário. <p>4. b Curtir uma citação publicada:</p> <ol style="list-style-type: none"> 1. O usuário acessa a citação desejada; 2. O usuário clica no símbolo de “Coração”; 3. O sistema apresenta uma animação, colorindo o símbolo e marcando a publicação como “curtida”; 4. O sistema notifica o usuário responsável pela publicação, desde de que seja diferente do responsável pela curtida, notificando-o. <p>4. b Comentar uma citação publicada:</p> <ol style="list-style-type: none"> 1. O usuário acessa a citação desejada;

	<ol style="list-style-type: none"> 2. O usuário clica no símbolo de “Balão de fala”; 3. O sistema apresenta uma animação, apresentando o campo de texto e outros comentários; 4. Após escrever o conteúdo da mensagem, o usuário poderá enviar seu comentário. 5. O conteúdo da mensagem é avaliado pelo sistema; 6. Retorna uma mensagem de confirmação de criação do comentário; 7. O sistema notifica o usuário responsável pela publicação, desde que seja diferente do responsável pelo comentário, notificando-o. <p>7. a Validação de conteúdo malicioso:</p> <ol style="list-style-type: none"> 1. O <i>backend</i> do sistema, capta a conteúdo da publicação e por meio de um <i>prompt</i> de comando, aciona a <i>Gemini AI</i>, solicitando uma análise do conteúdo; 2. O sistema, com base no retorno da Inteligência Artificial, irá retornar uma mensagem de sucesso e criará a publicação; 3. Caso o conteúdo da publicação infira algum conteúdo malicioso ou discriminatório, o sistema irá inibir sua publicação, retornar um alerta com o erro e solicitação de alteração do conteúdo.
--	---

Fonte: Elaborado pelo o autor.

Por fim, a pesquisa de livros possibilita aos usuários a busca por títulos e autores no sistema, integrando-se à *Google Books API* para fornecer informações detalhadas sobre cada obra. O Quadro 10 apresenta o fluxo dessa funcionalidade, demonstrando como os usuários podem explorar o catálogo disponível.

Quadro 10 - Caso de uso: Busca de livros

UC_4	Pesquisar livros
Autores	Usuário, <i>Google Books API</i> .
Interessados	Usuário.
Pré-condições	O usuário possui cadastro no sistema e está logado no sistema.
Pós-condições	Permitir o usuário pesquisar e visualizar informações de livros.
Cenário típico	<ol style="list-style-type: none"> 1. O usuário acessa a tela inicial do sistema; 2. O usuário acessa a página de “Livros” presente no menu lateral;

	<ol style="list-style-type: none"> 3. O sistema redireciona para a página de livros; 4. O sistema exibe a lista de livros indicados para o usuário; 5. O usuário seleciona a barra de pesquisa; 6. O usuário informa o nome do livro desejado; 7. O sistema captura o nome informado pelo o usuário; 8. O sistema realiza requisição à <i>API do Google Books</i> que retorna uma lista de livros; 9. O sistema exibe lista de livros para o usuário; 10. O usuário clica no componente “Ver”, ao lado das breves informações dos livros retornados; 11. O sistema exibe a tela com dados do livro e a possibilidade de adicionar a coleção.
Cenários alternativos	<p>8.a Google Books API retorna lista vazia:</p> <ol style="list-style-type: none"> 1. O sistema realiza requisição à <i>API do Google Books</i> que retorna uma lista vazia (livros não encontrados); 2. O sistema exibe mensagem avisando que não foram encontrados livros com o termo buscado; 3. Retorna ao cenário principal no passo 5.

Fonte: Elaborado pelo o autor.

Após a análise dos Casos de Uso apresentados, torna-se evidente a lógica e os passos necessários para que os usuários do sistema possam realizar as funcionalidades desejadas. Esses casos de uso fornecem um roteiro claro e detalhado das interações entre os usuários e o sistema, delineando os caminhos a serem seguidos para alcançar os objetivos pretendidos. Compreender esses Casos de Uso é fundamental para garantir que o sistema atenda às necessidades e expectativas dos usuários, proporcionando uma experiência intuitiva e eficiente.

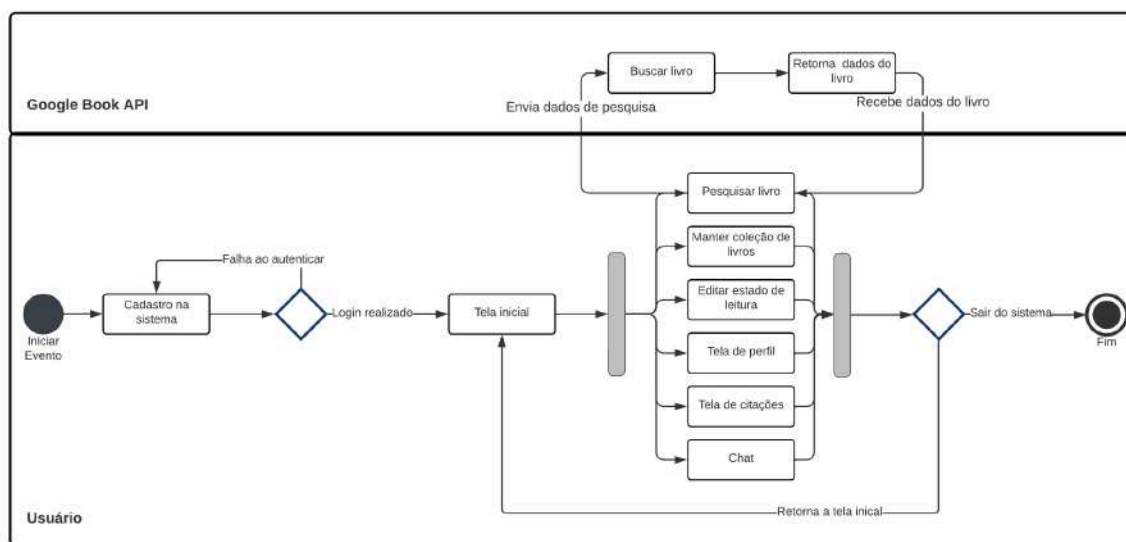
3.3.1.3 Diagrama de atividade

Os Diagramas de Atividades são uma ferramenta visual que complementa e amplia a compreensão do fluxo das funcionalidades do sistema ao representar de forma gráfica o fluxo de atividades dentro do sistema, representando o comportamento dinâmico e os processos internos. De acordo com Fowler (2005), “os Diagramas de Atividades são uma técnica para desenvolver lógica de procedimento, processos de negócios e fluxo de trabalho”. Na Figura 04 apresenta-se um Diagrama de Atividade do usuário com relação a API do *Google Books*.

Através da Figura 04 evidencia-se o fluxo de interação do usuário com o sistema, delineando as diversas etapas que um usuário segue durante a utilização normal do *Book Collection*. Além disso, a figura destaca a integração crucial entre o *Book Collection* e a API

do *Google Books*, responsável por fornecer os dados essenciais sobre os livros, tanto para apresentação quanto para pesquisa pelos usuários. Essa interação entre o sistema e a API garante uma experiência rica e completa aos usuários, ao mesmo tempo em que simplifica o acesso e a disponibilidade de informações sobre os livros presentes na plataforma.

Figura 04 - Diagrama de atividade



Fonte: Elaborado pelo o autor.

A Figura 04 ilustra claramente o fluxo de interação do usuário com o sistema, detalhando as etapas seguidas durante a utilização do *Book Collection*. Este diagrama proporciona uma visão abrangente da jornada do usuário, desde o início até o término de sua interação. Na próxima seção, abordaremos a modelagem do banco de dados, que será apresentada em duas etapas distintas: modelagem conceitual e lógica.

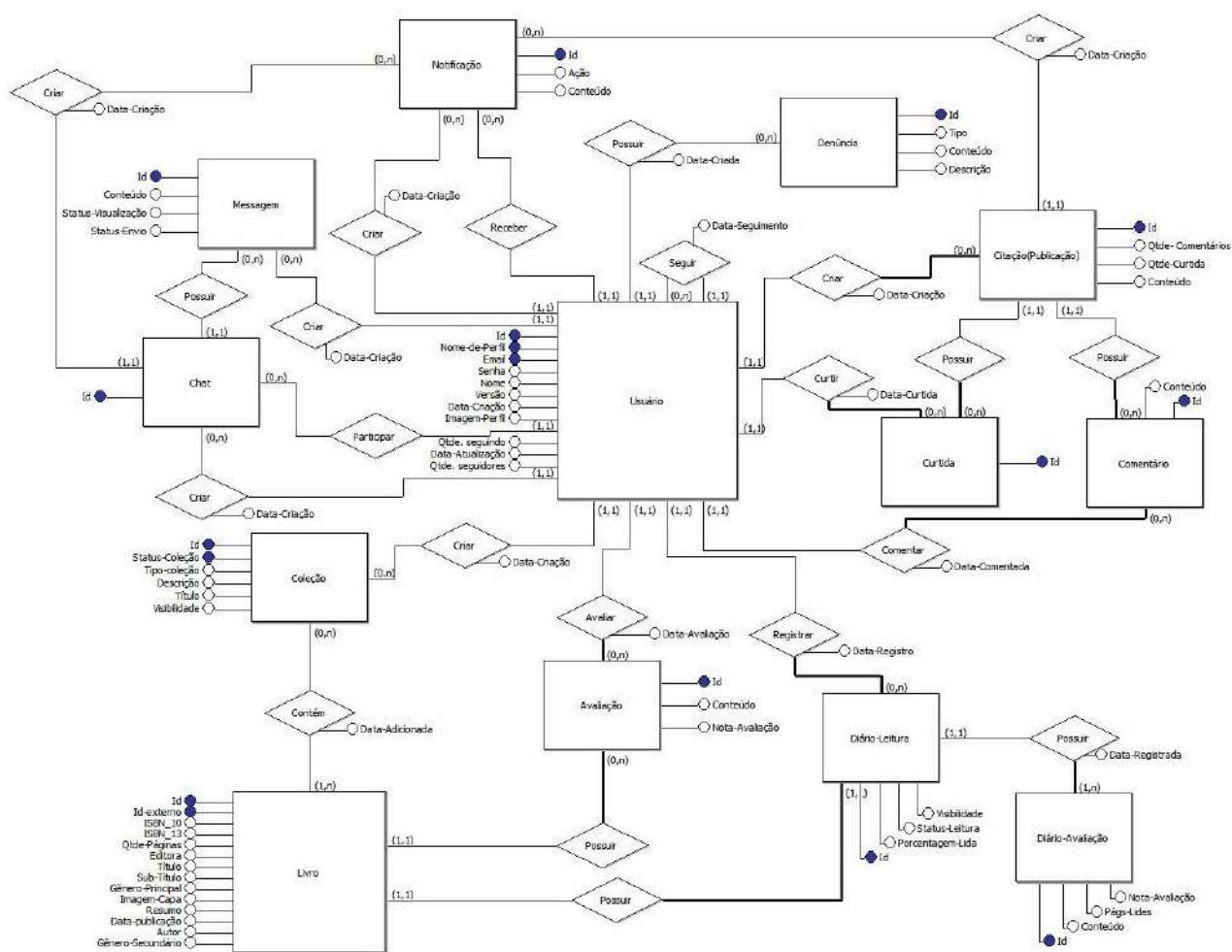
3.3.2 Modelagem de dados

Nesta seção, será realizada uma análise da modelagem do banco de dados do sistema, utilizando diagramas para representar sua estrutura, comportamento e interações. A modelagem de dados consiste na criação de um diagrama simplificado para organizar o armazenamento e o fluxo de informações essenciais para o desenvolvimento de um banco de dados. A notação adotada para a elaboração desses diagramas é a Linguagem de Modelagem Unificada (UML), definida por Guedes (2011), como, “uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos”. Os modelos discutidos neste capítulo incluem o Modelo Conceitual de Dados e o Modelo Lógico de Dados.

3.3.2.1 Modelagem Conceitual de dados

O Modelo Conceitual é uma representação abstrata dos conceitos e relacionamentos importantes em um domínio específico. Ele visa capturar os aspectos essenciais e fundamentais de um sistema ou processo, fornecendo uma visão de alto nível das entidades, atributos e relacionamentos envolvidos. De acordo com França & Celestino Jr. (2015), “No modelo conceitual devem estar presentes as entidades e seus relacionamentos além dos atributos das entidades e em alguns casos dos relacionamentos.” O modelo conceitual é uma etapa crucial no processo de modelagem de dados, pois ajuda a identificar e definir os principais elementos e suas interações antes de se detalhar a implementação técnica. A seguir, na Figura 05 apresenta-se a modelagem conceitual das entidades que envolvem a criação do banco de dados.

Figura 05 - Modelagem Conceitual dos Dados



Fonte: Elaborado pelo o autor.

Após a análise da modelagem conceitual dos dados, o *Book Collection* foi projetado com diversas entidades que refletem as principais interações do sistema. As entidades fundamentais incluem *Usuário*, responsável por armazenar informações essenciais como nome, e-mail e autenticação; e, *Livro*, que contém atributos como título, ISBN, autor e resumo. Para a organização das leituras, o sistema conta com a entidade *Coleção*, permitindo que os usuários classifiquem seus livros em categorias como "Já lido", "Lendo", "Deseja ler" e "Avaliados", além de possibilitar a criação de coleções personalizadas.

Além dessas, o sistema incorpora funcionalidades de interação social por meio das entidades *Chat* e *Mensagem*, que viabilizam a troca de mensagens entre usuários, bem como *Publicação*, que abrange postagens de citações e reflexões. As interações nessas postagens são registradas pelas entidades *Comentário* e *Curtida*, enquanto a entidade *Denúncia* permite relatar possíveis infrações, como a tentativa de publicação de conteúdo inadequado.

No que se refere ao acompanhamento da leitura, o sistema conta com a entidade *Diário de Leitura*, onde os usuários podem registrar seu progresso e anotações sobre suas leituras, e *Avaliação de Livro*, que armazena as resenhas e notas atribuídas pelos usuários. Essas avaliações são adicionadas automaticamente à coleção "Avaliados". Por fim, o sistema implementa a entidade *Notificação*, que informa os usuários sobre ações realizadas dentro da plataforma, garantindo uma experiência dinâmica e interativa.

Essas relações criam uma rede de interações, permitindo que os usuários sigam uns aos outros, enviem mensagens, curtam e comentem publicações, além de organizarem suas leituras de forma personalizada. As avaliações e o diário de leitura oferecem uma camada adicional de engajamento, permitindo que os usuários registrem e compartilhem suas impressões sobre os livros que estão lendo ou que já leram.

Essa estrutura de entidades e relacionamentos não só proporciona uma visão clara do modelo conceitual de dados do sistema, como também garante a escalabilidade e a flexibilidade necessárias para futuras expansões. Dessa forma, o *Book Collection* está bem preparado para suportar uma ampla gama de interações e funcionalidades, oferecendo uma experiência rica e personalizada para os usuários.

3.3.2.2 Modelagem Lógica de dados

O Modelo Lógico de dados é uma etapa essencial no processo de desenvolvimento de sistemas de informação, representando uma interpretação refinada dos requisitos do sistema em termos de estruturas de dados e relacionamentos. Segundo Guedes França & Celestino Jr. (2015), “o modelo lógico apresenta uma visão abstrata apropriada à equipe de

O Modelo Lógico define de forma precisa as tabelas, atributos e chaves necessárias para a criação do banco de dados, refletindo a estrutura lógica que suportará as operações do sistema *Book Collection*. Ele inclui a tabela *Usuário*, que armazena dados essenciais dos usuários, como IDs, nomes, e-mails e senhas, além de tabelas dedicadas a entidades como *Livro*, que detalha as informações dos livros, incluindo título, ISBN e resumo.

As relações entre as entidades são formalizadas por meio de chaves estrangeiras, assegurando a integridade referencial entre tabelas como *Mensagem*, *Publicação*, *Coleção* e *Avaliação de Livro*. Cada tabela é projetada para otimizar o armazenamento e a recuperação dos dados, garantindo que as interações dos usuários com o sistema sejam registradas e manipuladas de forma eficiente e consistente. A estrutura resultante é um banco de dados relacional robusto, pronto para atender às demandas do sistema com escalabilidade e desempenho.

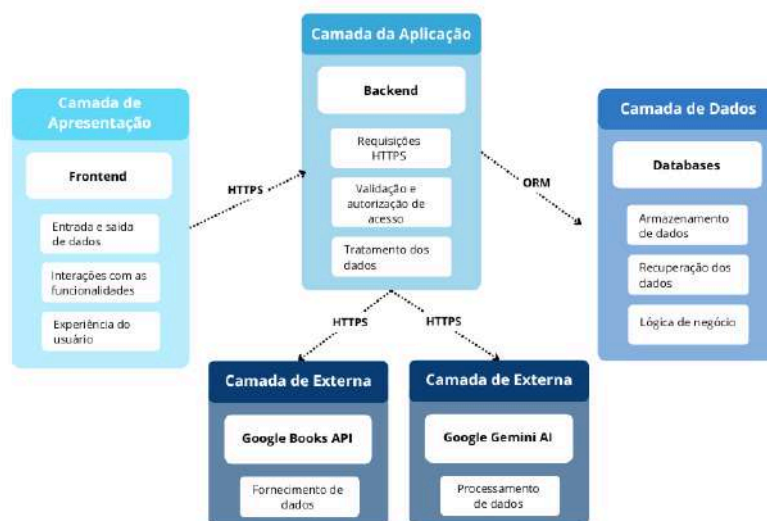
Em conclusão, o Modelo Lógico proporciona uma base sólida e detalhada para a implementação do banco de dados, transformando a abstração do modelo conceitual em uma estrutura concreta e funcional. Ao definir claramente as tabelas, atributos e chaves, ele assegura que o banco de dados não apenas reflita as necessidades do sistema, mas também opere com eficiência e precisão. Esse mapeamento lógico é crucial para garantir que todas as interações dos usuários e operações internas sejam realizadas de maneira coerente e segura, estabelecendo uma fundação robusta para o desenvolvimento e a expansão futura do sistema.

3.3.3 Modelagem da Arquitetura

Esta seção abordará uma análise e estudo acerca dos padrões organizacionais e estruturais da composição e implementação do código do sistema desenvolvido. De acordo com Sommerville (2011): “o projeto de arquitetura está preocupado com a compreensão de como um sistema deve ser organizado e com a estrutura geral desse sistema”. Sendo assim, o projeto de arquitetura é uma fase inicial crucial no desenvolvimento de *software*, pois define os principais componentes estruturais e suas interações.

Assim, a partir do estudo dos diagramas, da modelagem de dados e das tecnologias empregadas, como a API do *Google Books*, podem-se destacar os principais componentes e camadas da arquitetura da aplicação. O sistema é uma aplicação *web* que envolve interações com usuários, manipulação e acesso a dados, assegurando seu armazenamento. A figura a seguir apresenta um Diagrama de Camadas, estruturando a base para o entendimento da organização do sistema.

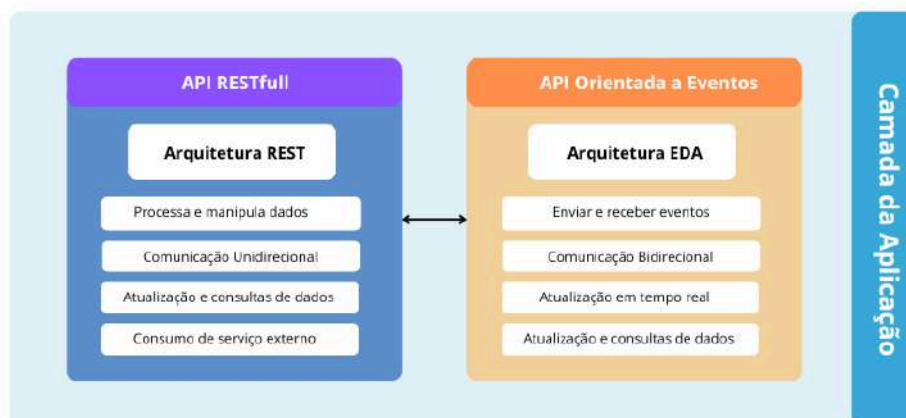
Figura 07 - Diagrama de Camadas da Arquitetura



Fonte: Elaborado pelo o autor.

Com base nesta análise inicial, após examinar os requisitos e funcionalidades do sistema, além de estudar arquiteturas que atendam a essas necessidades, identifica-se a importância de utilizar uma API para gerenciar a comunicação entre a Camada de Apresentação e as demais camadas do sistema. Entretanto, essa comunicação pode ser tanto unidirecional quanto bidirecional. Para tanto, são necessárias duas APIs, uma usando o modelo de integração, permitindo acesso aos dados por meio de comandos, e uma outra sendo orientada por eventos, permitindo que o sistema execute ações baseadas em eventos gerados nas camadas mais internas. Nesse contexto, a Figura 08 a seguir oferece uma visão detalhada da estrutura da Camada de Aplicação.

Figura 08 - Diagrama de Camadas da Camada de Aplicação



Fonte: Elaborado pelo o autor.

Por conseguinte, conforme a análise dos dois diagramas anteriores, entende-se que essa arquitetura, conseguem salientar os desafios e manter a funcionalidade do sistema. Ademais serão apresentados os padrões arquiteturais comumente usados em sistemas *web*, como o MVC, REST e o padrão de orientação a eventos, que serão aplicados no desenvolvimento do *Book Collection*.

3.3.3.1 MVC: *Model-View-Controller*

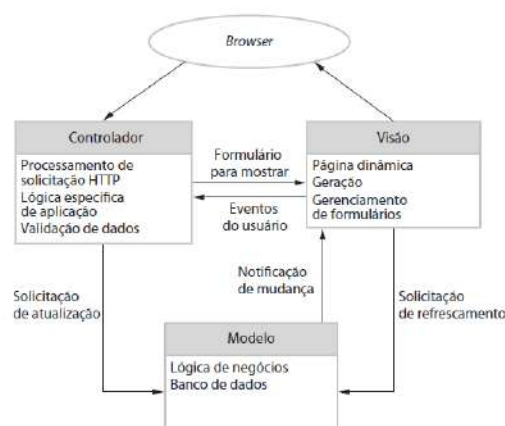
O Padrão de arquitetura de *software* Modelo-Visão-Controlador (MVC), ou em inglês, *Model-View-Controller*, propõem uma maneira de divisão de responsabilidade e funções entre três camadas: a camada para a interação com o usuário (*View*), a camada para a manipulação de dados (*Model*) e a camada de controle (*Controller*). Segundo Sommerville (2011); “esse padrão é a base do gerenciamento de interação em muitos sistemas baseados em *Web*”. Por tais características o padrão arquitetural foi escolhido como base para o desenvolvimento desse sistema.

Com o MVC é possível dividir os códigos relativos à implementação da interface do usuário, as regras de negócio e ao armazenamento de dados, sendo assim, segundo as definições de Sommerville:

“[...] O componente Modelo gerencia o sistema de dados e as operações associadas a esses dados. O componente Visão define e gerencia como os dados são apresentados ao usuário. O componente Controlador gerencia a interação do usuário (por exemplo, teclas, cliques do mouse etc.) e passa essas interações para a Visão e o Modelo” (Sommerville, 2011).

Sendo assim, a seguir uma representação estrutural do padrão MVC:

Figura 09 - Arquitetura de aplicações Web usando o padrão MVC



Fonte: (Sommerville, 2011).

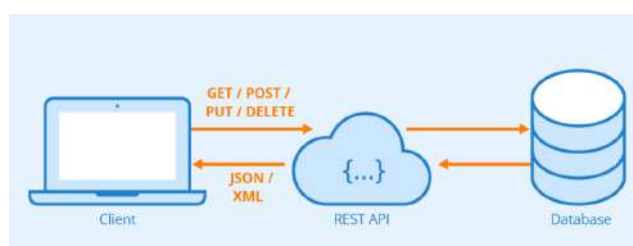
Por meio da análise, percebe-se que o padrão arquitetural MVC apresenta-se inúmeras vantagens para o desenvolvimento de *software*, onde uma dessas vantagens refere-se a manutenibilidade do sistema e facilidade de realizar testes e encontrar possíveis erros, como referido por Sommerville (2011); “às noções de separação e independência são fundamentais para o projeto de arquitetura, porque permitem que alterações sejam localizadas”, e neste sentido, observa-se que a política e estrutura de dividir o código em três camadas promovem essas noções de separação e independência referidas pelo autor.

Sendo assim, no contexto do *Book Collection*, a arquitetura MVC, apresenta uma estrutura de organização dos códigos, separando os códigos relacionados ao *frontend* e ao *backend*. Dessa forma, ao escolher o padrão arquitetural MVC para a estrutura base do sistema *Book Collection*, garante-se uma maior flexibilidade, facilidade de manutenção e implementação, além de facilitar a análise e integração com outras plataformas e serviços externos.

3.3.3.2 REST - *Representational State Transfer*

A *Representational State Transfer*, ou simplesmente REST (em português, transferência de estado representacional), é uma arquitetura de software que define um conjunto de princípios e diretrizes para a criação de serviços *web* que sejam escaláveis, flexíveis e interoperáveis, sendo assim, padronizando o funcionamento de uma API, as APIs baseadas nessa arquitetura são nomeadas por API REST ou API RESTful. Baseada em princípios como a utilização de Identificadores de Recursos Uniformes (URIs) para identificar recursos de forma única e a utilização adequada dos métodos HTTP para operações *Create*, *Read*, *Update* e *Delete* (CRUD), a arquitetura REST promove a construção de APIs flexíveis e escaláveis. Sendo assim, a seguir uma representação estrutural do funcionamento da arquitetura REST:

Figura 10 - Funcionamento da Arquitetura REST



Fonte: (Hostinger²⁴, 2023).

²⁴ Hostinger. **O que é uma API RESTful e Porque Isso Importa?** Disponível em: <https://www.hostinger.com.br/tutoriais/api-restful>. Acesso em: 26/04/2024.

Segundo o estudo realizado pela *Hostinger*:

“Assim, quando um usuário deseja usar uma funcionalidade da aplicação, seu dispositivo envia uma solicitação via HTTP ao servidor. O servidor localiza o recurso e comunica a representação do estado dele na resposta ao usuário através do mesmo protocolo. E são essas representações que podem ser feitas em diversos formatos.” (Hostinger, 2023).

No contexto do *Book Collection*, a escolha de adotar uma arquitetura REST para a API se justifica pelas suas vantagens em termos de simplicidade, escalabilidade e facilidade de integração. Ao seguir os princípios do REST, como identificação de recursos, uso de métodos HTTP e representação dos dados em formato *JavaScript Object Notation (JSON)*, pode-se criar uma API consistente e intuitiva, que seja fácil de entender e usar tanto para desenvolvedores quanto para clientes.

Além disso, a adoção de uma API REST é especialmente benéfica quando se planeja integrar com outras APIs, como é o caso da API do *Google Books*. Como ambas as APIs seguem o protocolo HTTP, torna-se mais simples e eficiente realizar requisições e trocar dados entre elas. Isso possibilita uma integração mais transparente e robusta, permitindo que o *Book Collection* aproveite os recursos e informações disponíveis na API do *Google Books* de forma eficaz e sem complicações.

Dessa forma, ao escolher a arquitetura REST para a API do *Book Collection*, garante uma maior flexibilidade, interoperabilidade e facilidade de manutenção, além de facilitar a integração com outras plataformas e serviços externos, como a API do *Google Books*.

3.3.3.3 EDA: *Event-Driven Architecture*

A Arquitetura Orientada a Eventos, ou do inglês *Event-Driven Architecture (EDA)*, é um estilo arquitetural onde a comunicação entre os componentes do sistema é feita por meio de eventos, que representam mudanças ou ações de interesse. Para tanto, segundo Picanço (2023), um evento “representa um fato significativo, uma mudança de estado; ele é imutável: uma vez que aconteceu, não pode ser desfeito.” Em vez de os componentes se comunicarem diretamente, eles emitem eventos que são processados por outros componentes, criando um fluxo assíncrono de informações.

“(EDA) É um estilo arquitetural no qual aplicações ou microsserviços reagem a eventos em tempo real após alguma mudança de estado, em vez de responder a comandos.” (Picanço, 2023).

Nesse contexto, o artigo da IBM²⁵ explica que,

“Quando um aplicativo ou serviço toma uma ação ou inicia uma mudança cujas informações são necessárias para outro aplicativo ou serviço, ele publica um evento, ou seja, um registro dessa ação ou mudança, que outro aplicativo ou serviço pode consumir e processar para tomar ações como resposta.” (IBM, 2023).

No sistema *Book Collection*, a EDA será essencial para viabilizar a comunicação assíncrona entre a camada de apresentação e a camada de aplicação. Isso permitirá que ações como enviar mensagens, comentar e curtir uma publicação sejam refletidas imediatamente na interface do usuário, sem a necessidade de recarregar a página. A utilização de *Socket.io*, uma biblioteca que possibilita a troca de eventos em tempo real, será fundamental para essa interação dinâmica.

O grande diferencial da Arquitetura Orientada a Eventos em relação à arquitetura baseada em *solicitação/resposta* (ou modelo de integração) é o desacoplamento entre os componentes que geram e consomem eventos. Enquanto, na arquitetura de integração, os comandos exigem uma ação específica e uma resposta direta, na EDA os eventos simplesmente notificam que algo aconteceu, sem esperar uma resposta imediata. Isso torna o sistema mais flexível, escalável e adequado para interações assíncrona.

Dessa forma, ao adotar a EDA, o sistema *Book Collection* poderá ser desenvolvido para seguir os princípios de comunicação instantânea e bidirecionalidade, essenciais para funcionalidades como notificações instantâneas, *chats* e trocas de mensagens, oferecendo uma experiência mais interativa e eficiente para os usuários.

3.3.3.4 Correlação entre as arquiteturas

Desse modo, o padrão arquitetural do sistema será desenvolvido com uma correlação entre o MVC, REST e o EDA, mesclando os benefícios de ambas as arquiteturas. No contexto do *Book Collection*, o MVC será utilizado para organizar o sistema em camadas: o *Model*, que gerencia os dados e a lógica de negócios por meio do *Prisma ORM* e *PostgreSQL*; a *View*, desenvolvida com *React* e *TypeScript*, que fornece a interface do usuário; e o *Controller*, responsável por processar as solicitações HTTP via *Express*, garantindo a comunicação entre o *frontend* e o *backend* de forma eficiente.

Complementando essa estrutura, o padrão REST, essencial para a construção de APIs, segue princípios como a *statelessness* e o uso de métodos HTTP para manipular

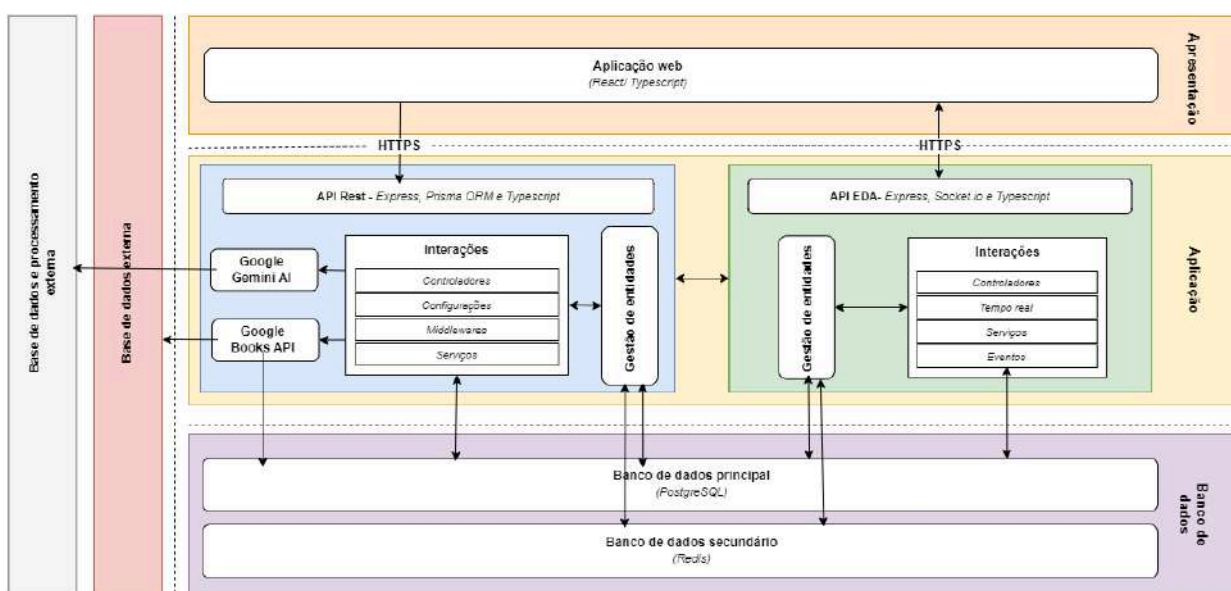
²⁵ IBM. *Arquitetura orientada a eventos*. Disponível em: <https://www.ibm.com/br-pt/topics/event-driven-architecture>. Acesso em: 28 jan. 2025.

recursos. No *Book Collection*, a API REST gerenciada pelo *Express* facilita a comunicação entre o *frontend* e o *backend*, permitindo operações como criação, leitura, atualização e exclusão de informações de livros e dados de usuários. Assim, o REST assegura escalabilidade e eficiência na comunicação entre as partes do sistema, garantindo desempenho e integridade dos dados.

Além disso, a EDA será crucial para garantir funcionalidades assíncronas no sistema. Tecnologias como *Socket.io* serão utilizadas para permitir a comunicação assíncrona e a troca de mensagens entre usuários, sem a necessidade de recarregar a interface. A EDA viabiliza, por exemplo, a implementação de notificações instantâneas sobre novos comentários, curtidas ou atividades nos perfis de usuários. Além disso, a comunicação em tempo real entre usuários, facilitando *chats* e trocas de mensagens, será otimizada pela arquitetura de eventos, permitindo uma experiência mais fluida e dinâmica, sem comprometer a performance do sistema.

Ao correlacionar essas três arquiteturas em um diagrama unificado, podemos visualizar como o padrão MVC se integra com o REST e EDA para formar uma estrutura coesa e eficiente, criando um ambiente altamente interativo e responsivo, que atende às demandas dos usuários de forma eficiente e escalável. Assim, o diagrama a seguir apresenta esta correlação entre estas arquiteturas, bem como o uso das tecnologias antes apresentadas:

Figura 11 - Correlação entre as arquiteturas do sistema



Fonte: Elaborado pelo o autor.

Este diagrama, ao ser implementado, fornecerá uma visão clara de como os diferentes componentes do sistema se relacionam e interagem, servindo como um guia essencial para a implementação e manutenção contínua do código. A distribuição das tecnologias nas camadas de Banco de Dados (*Model*), Apresentação (*View*) e Aplicação (*Controller*) garante que cada parte do sistema tenha funções bem definidas, facilitando a organização e o desenvolvimento.

A combinação dos padrões MVC, REST e EDA no desenvolvimento do *Book Collection* não só otimiza o fluxo de dados e a estrutura organizacional, como também estabelece uma base sólida para futuras expansões e melhorias da plataforma.

3.3.4 Conclusão da modelagem

A conclusão da seção de Modelagem apresentada neste capítulo destaca a importância de cada etapa na construção do sistema *Book Collection*. A modelagem de sistema definiu os principais casos de uso e fluxos de interação entre os componentes, enquanto a modelagem de dados estabeleceu a estrutura do banco de dados, assegurando a integridade e eficiência no armazenamento das informações.

Já a modelagem de arquiteturas demonstrou a aplicação de padrões sólidos, como o MVC, REST e EDA, alinhados com as tecnologias escolhidas para o projeto. Com essa base devidamente estruturada, a próxima seção explorará as funcionalidades principais do sistema, detalhando como elas serão implementadas para garantir uma experiência eficaz e intuitiva ao usuário.

3.4 Funcionalidades

Esta seção apresentará as principais funcionalidades planejadas para o desenvolvimento do sistema *Book Collection*. O objetivo é detalhar e descrever as interações dos usuários com o sistema, além de como o sistema responde a essas interações. Serão abordadas as funcionalidades essenciais que compõem a experiência do usuário, fornecendo uma visão clara de como cada aspecto do sistema foi projetado para atender às necessidades dos usuários.

3.4.1 Autenticação e gerenciamento de sessão

Nesta funcionalidade, ocorre a interação inicial entre o usuário e o sistema *Book Collection*, onde o usuário pode realizar o *login* ou cadastro. Durante o processo de autenticação, o usuário fornece suas credenciais, que são verificadas pelo sistema. Uma vez

autenticado, o sistema gera um *token* de autorização, utilizando o *JSON Web Token (JWT)*²⁶, com uma chave privada. Esse *token* é então retornado ao usuário, permitindo seu acesso ao sistema através do mecanismo *Bearer Token*.

Além disso, como as interfaces do sistema são desenvolvidas utilizando a tecnologia *React*, um gerenciamento de estado global é implementado por meio do *React Context*. Essa abordagem permite armazenar e compartilhar o *token* de acesso em toda a aplicação, garantindo que o usuário autenticado possa acessar as demais funcionalidades de forma segura e eficiente. Adicionalmente, para melhorar o desempenho e garantir a disponibilidade rápida das informações, os dados dos usuários autenticados e autorizados serão armazenados em um banco de dados de cache, utilizando o *Redis*.

3.4.2 Gestão de perfil e edição concorrente

Uma das funcionalidades centrais do *Book Collection* é a gestão de perfil do usuário, que abrange o controle de informações pessoais, preferências e interações sociais no sistema. Entre essas funcionalidades, destaca-se a edição do perfil, que possibilita ao usuário modificar suas informações, como nome, foto, e detalhes de contato.

No entanto, por se tratar de um ambiente *web*, existe a possibilidade de acessos concorrentes, em que uma mesma conta pode ser acessada em dois dispositivos ou sessões simultâneas. Essa situação pode gerar conflitos quando mais de um perfil tenta editar as informações ao mesmo tempo. Para garantir a consistência e evitar a sobrescrita de dados importantes, o sistema *Book Collection* adotará mecanismos de controle de concorrência.

Esses mecanismos envolverá a implementação de um bloqueio otimista (*optimistic locking*), onde ao carregar os dados do perfil para edição, o sistema armazena um identificador de versão. Ao salvar as alterações, essa versão é verificada para assegurar que nenhuma outra modificação tenha sido feita entre o momento de carregamento dos dados e a tentativa de salvamento. Caso haja conflito, o sistema notificará o usuário para que revise suas alterações, garantindo que nenhuma informação seja perdida.

3.4.3 Gerenciamento de recomendações literárias

A recomendação personalizada de livros é uma das funcionalidades centrais do *Book Collection*, proporcionando aos usuários sugestões alinhadas aos seus interesses literários. Para garantir uma experiência relevante e envolvente, é fundamental compreender os critérios utilizados para gerar essas recomendações e como elas são aplicadas no sistema.

²⁶ JWT. Disponível em: <https://jwt.io>. Acesso em: 05/09/2024.

O sistema baseia-se em alguns parâmetros para sugerir livros: a análise dos principais gêneros textuais presentes na estante virtual do usuário, bem como pelos gêneros e autores predominantes em cada uma das suas coleções, sejam elas públicas ou privadas. Assim, à medida que o usuário busca e adiciona livros às suas coleções, suas preferências são atualizadas, permitindo que novas recomendações sejam ajustadas dinamicamente.

No entanto, essa abordagem funciona melhor para usuários já ativos na plataforma. Para novos usuários, cuja estante virtual está vazia, o sistema adota uma estratégia diferente, combinando duas ferramentas fundamentais: *Gemini AI* e *Google Books API*. O *Gemini AI* é responsável por identificar padrões literários atuais e sugerir um conjunto inicial de gêneros e autores. Com base nesses dados, a *API do Google Books* realiza uma busca para oferecer sugestões iniciais que ajudem a traçar o perfil literário do usuário. Com esses dados iniciais, o sistema permite que o usuário selecione as obras que melhor correspondem aos seus gostos, proporcionando um ponto de partida mais preciso para futuras recomendações.

Apesar de eficiente nesta primeira versão do sistema, esse modelo apresenta algumas limitações, especialmente na busca realizada pela *API do Google Books*. Os testes, realizados durante o desenvolvimento, indicaram que pesquisas por gêneros, mesmo quando variados, retornam resultados repetitivos, com pouca variação de livros sugeridos. Diante disso, embora a funcionalidade atenda aos requisitos iniciais, futuras melhorias serão necessárias para aprimorar a precisão das recomendações.

3.4.4 Gerenciamento de estante virtual

No sistema *Book Collection*, a "*Estante Virtual*" representa a organização das coleções de livros de cada usuário. No banco de dados, como representado nas Figuras 5 e 6, isso é refletido em uma tabela dedicada, onde estão armazenadas as definições das coleções de livros. Assim, "*Estante Virtual*" segue como uma denominação fora do escopo de desenvolvimento do sistema, mas sendo usado para referir a organização de coleções de um usuário.

As coleções se dividem em dois tipos: coleções padrão e coleções personalizadas. As coleções padrão são automaticamente criadas quando o usuário se cadastra, e incluem as categorias "Lidos", "Lendo", "Desejo Ler" e "Avaliados". Dentre estas, apenas "Lidos", "Lendo", e "Desejo Ler" podem ser diretamente manipuladas pelo usuário, permitindo adicionar e remover livros conforme desejado. Já a coleção "Avaliados" é atualizada automaticamente pelo sistema toda vez que o usuário avalia um livro, e não pode ser editada manualmente.

Além disso, as coleções personalizadas permitem que o usuário crie e gerencie novas coleções de acordo com seus interesses específicos. Essas coleções podem ser nomeadas e organizadas livremente, proporcionando uma experiência mais personalizada na gestão de sua biblioteca virtual.

3.4.5 Gerenciamento de camada social

No contexto do *Book Collection*, que se define como uma rede social voltada para a interação entre leitores, diversas funcionalidades sociais foram implementadas para promover essa interação. Entre elas, destacam-se a publicação de notas na comunidade, o envio de mensagens, e a possibilidade de seguir e ser seguido por outros usuários.

Dado que essas funcionalidades envolvem interações instantâneas entre múltiplos usuários, é essencial o uso de uma tecnologia capaz de suportar comunicação instantânea. Para isso, será adotada a biblioteca *Socket.io*, uma solução robusta para a comunicação bidirecional em tempo real em aplicações *web*, em conjunto com uma API orientada a eventos. A *Socket.io* facilita a criação de canais de comunicação persistentes entre o cliente e o servidor, permitindo que as mensagens e atualizações sejam transmitidas instantaneamente, sem a necessidade de recarregar a página.

3.4.5.1 Publicação de notas na comunidade

A funcionalidade de publicação de notas permite que os usuários compartilhem pensamentos, opiniões e resenhas com a comunidade. Essas notas são visíveis para outros usuários que tenham uma relação de "*Amizade*" no sistema, ou seja, que sigam e sejam seguidos reciprocamente. Além disso, as notas podem receber curtidas e comentários, promovendo ainda mais a interação social. A *Socket.io* garante que as notificações de novas curtidas ou novos comentários, sejam instantaneamente atualizadas para os usuários conectados.

3.4.5.2 Envio de mensagens

Outra funcionalidade essencial é o envio de mensagens privadas entre os usuários. Esta permite que dois usuários conversem diretamente, trocando mensagens de texto em tempo real. Novamente, a *Socket.io* desempenha um papel crucial aqui, assegurando que as mensagens enviadas sejam imediatamente entregues ao destinatário, criando uma experiência de *chat* fluida e responsiva. As mensagens são armazenadas no banco de dados para garantir persistência, permitindo que os usuários acessem o histórico de conversas quando necessário.

3.4.5.3 Seguir e ser seguido

Por fim, a funcionalidade de seguir e ser seguido estabelece a rede de "*Amizades*" no *Book Collection*. Quando um usuário segue outro, ele ganha acesso às publicações e atualizações daquele perfil. A *Socket.io* também é usada para notificar instantaneamente os usuários sobre novos seguidores ou alterações na sua rede social.

Estas funcionalidades foram desenvolvidas para criar um ambiente interativo e dinâmico, onde os usuários podem se conectar e compartilhar suas experiências de leitura. As Figuras 32, 39, e 25 ilustram, respectivamente, como essas funcionalidades são apresentadas na interface do usuário, proporcionando uma visão clara de como o sistema facilita a interação social.

3.4.6 Gerenciamento de estados de leitura

No contexto do *Book Collection*, a funcionalidade de Gerenciamento dos Estados de Leitura desempenha um papel crucial na experiência do usuário, permitindo uma interação rica e organizada com o acervo de livros disponíveis na plataforma.

Através da integração com a API externa, *Google Books API*, o sistema acessa e exibe informações detalhadas dos livros buscados pelos usuários. Uma vez que o usuário encontra o livro desejado, conforme mostrado na Figura 27, ele pode visualizar essas informações e interagir com o sistema de diversas maneiras, como avaliar o livro, adicioná-lo a uma coleção ou manter um diário de leitura.

Conforme descrito na subseção de *Gerenciamento da Estante Virtual*, um livro pode ser associado a uma ou mais coleções do usuário, com a possibilidade de mudar a coleção à qual o livro pertence. O *Book Collection* oferece também uma funcionalidade dedicada à avaliação de livros, permitindo que os usuários expressem suas opiniões e notas sobre os títulos que leram. Essas avaliações são armazenadas no banco de dados do sistema e, de forma automática, o livro avaliado é adicionado à coleção de livros avaliados, sem necessidade de ação adicional por parte do usuário.

Além disso, o sistema oferece a funcionalidade exclusiva de gerenciamento de diários de leitura. Embora a primeira versão do *Book Collection* não permita a leitura direta de livros na plataforma, ela oferece aos usuários a possibilidade de registrar suas impressões e reflexões ao longo da leitura de um livro específico. O diário de leitura funciona como um registro contínuo, permitindo que o usuário adicione comentários, atribua notas e mantenha um histórico detalhado de sua experiência com cada obra.

É importante destacar que as informações registradas podem ser definidas como públicas ou privadas, sendo a visibilidade dessas informações determinada pela escolha do próprio usuário. Além disso, o sistema realiza automaticamente o cálculo da porcentagem de leitura com base no número de páginas do livro e na quantidade de páginas lidas informadas pelo usuário.

Assim, o gerenciamento de diários de leitura, como ilustrado nas Figuras 34, 35 e 36, configura-se como um diário virtual, que enriquece a experiência do leitor e facilita o acompanhamento de suas leituras ao longo do tempo.

3.4.7 Gerenciamento de comunicação com serviços externos

Para gerenciar e controlar grande parte das funcionalidades do *Book Collection*, tornou-se essencial a integração com serviços externos, conforme apresentado na seção de *Tecnologias Utilizadas*. Embora, em um cenário futuro, essa dependência possa limitar a expansão ou o desenvolvimento de novas funcionalidades, nesta primeira versão, o uso dessas ferramentas externas se mostra vantajosa, pois está alinhada com os objetivos e funcionalidades do sistema. A seguir, será detalhada a forma como o sistema se comunica com esses serviços.

3.4.7.1 Comunicação e consumo com o *Google Books API*

Conforme descrito na seção de *Tecnologias Utilizadas*, a *Google Books API* fornece acesso a uma ampla base de dados sobre livros, permitindo buscas com diversos filtros. No contexto do *Book Collection*, essa integração se mostra essencial, pois garante a obtenção e o gerenciamento das principais informações do sistema: os dados dos livros.

A integração com a *Google Books API* pode ser realizada por meio de uma "*chave de API*", que identifica a aplicação ao acessar os serviços da API, ou através de um "*token OAuth 2.0*", utilizado para acessar recursos privados. Como o objetivo deste projeto é apenas consultar informações públicas sobre livros, optou-se pelo uso da chave de API. Para isso, é necessário obter uma chave de acesso na página de credenciais do *Google Cloud*²⁷, ressaltando que é indispensável possuir uma conta de desenvolvedor.

No *Book Collection*, a API é utilizada para buscar e exibir informações sobre livros. Esse consumo se dá tanto pelas pesquisas realizadas diretamente pelo usuário, como demonstrado na Figura 26, onde é possível buscar um livro pelo título, quanto pela exibição de mais detalhes na página de detalhes do livro, conforme ilustrado na Figura 27. Além disso,

²⁷ Google Cloud - Disponível na URL: <https://console.cloud.google.com/apis/credentials>. Acesso em: 29/01/2025.

a API é empregada na recomendação de livros com base nos gostos do usuário, conforme abordado na subseção *Gerenciamento de Recomendações de Livros*, atuando tanto de forma independente quanto em conjunto com a IA para fornecer sugestões personalizadas.

A *Google Books API* disponibiliza uma rota HTTP que permite a busca e recuperação de informações sobre livros com base em diferentes critérios. No *Book Collection*, essa integração ocorre por meio de requisições que possibilitam pesquisas por título, autor, ISBN e gênero. Assim, ao realizar uma busca, o sistema envia uma solicitação para a API, que retorna os livros correspondentes ao critério pesquisado.

Dessa forma, o Algoritmo 02, ilustra parcialmente o processo de consumo da *Google Books API* no sistema. Além disso, ele exemplifica uma requisição para a busca de livros com base no título, demonstrando a integração da API ao fluxo do *Book Collection*.

Algoritmo 02 - Consumo do *Google Books API*

```

1  export class GoogleBookService {
2    private googleApiUrl: string;
3    private googleApiKey: string;
4
5    constructor () {
6      //Buscando variáveis de ambiente
7      const urlGoogleApi = process.env.GOOGLE_API;
8      const keyGoogleApi = process.env.GOOOGLE_API_KEY;
9
10     //Validando que elas são válidas
11     if (!urlGoogleApi || !keyGoogleApi) throw new Error("Url for API GOOGLE
12     BOOK is not found");
13
14     //Settando atributos privados
15     this.googleApiUrl= urlGoogleApi;
16     this.googleApiKey = keyGoogleApi;
17   };
18
19   //Método para procurar livros por titulo
20   async searchBookForTitle (title: string) {
21     try {
22       const response = await axios.get(`${this.googleApiUrl}/volumes`, {
23         params: {
24           q: `intitle:${title}`,
25           key: this.googleApiKey
26         }
27       });
28       return response.data;
29     } catch (error) {
30       throw new Error("Error in search book for title");
31     }
32   };

```

Fonte: Elaborado pelo autor.

Apesar da riqueza de informações fornecidas, a API apresenta algumas limitações que podem impactar o sistema no futuro, sendo a principal delas a restrição de requisições diárias. Inicialmente, há um limite de aproximadamente mil requisições por dia, além de restrições

quanto ao número de solicitações permitidas por minuto. Para lidar com essa limitação, o sistema armazena localmente os dados dos livros consultados pelos usuários, reduzindo a necessidade de buscas repetitivas. No entanto, essa abordagem pode gerar novos desafios, como o crescimento acelerado da base de dados e possíveis impactos no desempenho do sistema, especialmente à medida que o número de usuários aumenta.

3.4.7.2 Comunicação e consumo com o *Google Gemini AI*

Outra ferramenta desenvolvida pelo *Google*²⁸ que desempenha um papel importante no sistema é o *Gemini AI*. Como descrito na seção de *Tecnologias Utilizadas*, essa inteligência artificial foi incorporada ao *Book Collection* com o objetivo principal de aprimorar a experiência do usuário, auxiliando na geração de recomendações literárias e em outras funcionalidades, que demandam análise e processamento de dados textuais.

No sistema *Book Collection*, o *Gemini AI* desempenha um papel fundamental em diversas funcionalidades. Uma de suas aplicações principais é a moderação de conteúdos, garantindo que as interações dos usuários dentro da plataforma sejam seguras e respeitosas. O modelo é responsável por analisar notas de comunidade, avaliações e comentários, identificando automaticamente discursos de ódio, conteúdos discriminatórios ou inapropriados. Isso impede a propagação de mensagens ofensivas e mantém a integridade da experiência social na plataforma.

É importante destacar que o sistema respeita a privacidade dos usuários, garantindo que a moderação de conteúdos se limite a interações públicas, como notas de comunidade, avaliações e comentários. Dessa forma, mensagens privadas entre usuários não são analisadas pelo *Gemini AI*, garantindo que nenhum processo externo interfira nas interações privadas dos usuários, preservando, assim, a confidencialidade e privacidade dessas comunicações.

Além disso, o *Gemini AI* contribui para a complementação e correção de informações sobre livros. Como a API do *Google Books* pode fornecer dados incompletos, como a ausência de resumos ou descrições detalhadas de algumas obras, a IA é capaz de gerar textos coerentes e informativos para preencher essas lacunas. Isso melhora significativamente a experiência do usuário, garantindo um catálogo mais estruturado e consistente.

Outra aplicação importante da IA no *Book Collection* está na recomendação personalizada de livros. Para novos usuários, a inteligência artificial sugere obras com base em padrões de leitura amplamente reconhecidos. Com o tempo, conforme o usuário interage com a plataforma, avaliando e organizando seus livros, o sistema ajusta suas sugestões de

²⁸ Google for developers - Disponível em: <https://www.google.com>. Acesso em 02/02/2025.

maneira cada vez mais precisa, considerando os gostos e preferências individuais. Essa personalização não apenas melhora a experiência dentro do sistema, mas também contribui para um maior engajamento dos leitores, incentivando a descoberta de novas leituras alinhadas aos seus interesses.

Assim, como ocorre com a *Google Books API*, o *Gemini AI* é acessado por meio de uma “*chave de API*”, que pode ser gerada no *Google AI Studio*²⁹. Para obter essa chave, é necessário acessar a plataforma, criar um projeto e configurar uma credencial de acesso. Após esse processo, a chave pode ser utilizada na configuração da API dentro do sistema.

A comunicação com o *Gemini AI* ocorre através de requisições HTTP, onde o sistema envia um comando estruturado na forma de um *prompt* e recebe como resposta um conteúdo gerado pela IA. Como se trata de um modelo de linguagem de grande escala (LLM), a qualidade das respostas depende diretamente da clareza e precisão do *prompt* enviado. Assim, para garantir um bom desempenho da IA no sistema, os comandos precisam ser bem formulados, diretos e específicos.

A seguir, serão apresentados alguns algoritmos que detalham a configuração do *Gemini AI* no *Book Collection*, bem como exemplos de *prompts* utilizados para otimizar a geração de recomendações e demais interações no sistema.

²⁹ Google AI Studio - Disponível na URL: <https://aistudio.google.com>. Acessado em: 29/01/2025.

Algoritmo 03 - Configuração do *Gemini AI*

```
1  export class GoogleGeminiService {
2    private geminiKey: GoogleGenerativeAI;
3    private modelGemini: GenerativeModel;
4
5    constructor() {
6      const keyGoogleGemini = process.env.GEMINI_API_KEY;
7
8      if (!keyGoogleGemini) throw new Error("Key for API GOOGLE GEMINI is not
9 found");
10
11     this.geminiKey = new GoogleGenerativeAI(keyGoogleGemini);
12     this.modelGemini = this.geminiKey.getGenerativeModel({ model:
13 "gemini-1.5-flash" });
14   };
15
16   //Função auxiliar, realização da pesquisa pela IA
17   private async runningQuery(prompt: string): Promise<{success: boolean, message:
18 string}> {
19     try {
20       const response = await this.modelGemini.generateContent(prompt.trim());
21
22       const candidates = response?.response?.candidates;
23       if (!candidates || !candidates[0]?.content?.parts?.[0]?.text) {
24         return { success: false, message: "Response IA not found" };
25       }
26
27       return { success: true, message:
28 candidates[0].content.parts[0].text.trim() };
29
30     } catch (error) {
31       return { success: false, message: "An error occurred while generating the
32 query" };
33     }
34   };
35 }
```

Fonte: Elaborado pelo autor.

Algoritmo 04 - Geração de *Prompts*

```

1 // Método para gerar um resumo para um livro que não possua resumo
2 async createBookSummary(title: string, author: string): Promise<{ success:
3 boolean, message: string }> {
4   try {
5     // Formulando a requisição de conteúdo com um prompt claro
6     const prompt = `
7       Crie um resumo para o livro: $(title), do(a) autor(a): $(author).
8       O resumo deve conter entre dez a 120 palavras. O resumo deve falar
9       sobre a história e enredo do livro, destacando nomes de personagens
10      principais e secundário. Retorne o resumo em uma string. Ignore
11      outros parâmetros e respostas, retorne somente o resumo do livro
12      indicado.
13    `;
14
15    // Enviando a requisição e aguardando resposta
16    const response = await this.runningQuery(prompt);
17
18    // Retornando o texto da query gerada
19    return { success: response.success, message: response.message };
20
21    } catch (error) {
22      return { success: false, message: "An error occurred while generating
23      the query" };
24    };

```

Fonte: Elaborado pelo autor.

Apesar das vantagens proporcionadas pela integração do *Gemini AI*, seu uso no *Book Collection* também apresenta algumas limitações que devem ser consideradas. Como qualquer modelo de inteligência artificial, o *Gemini AI* pode gerar respostas imprecisas ou inconsistentes, especialmente quando os *prompts* enviados não são claros ou suficientemente específicos. Além disso, sua capacidade de interpretar nuances subjetivas em recomendações literárias pode ser limitada, o que pode impactar a personalização das sugestões e a precisão das análises de conteúdo geradas pela IA.

Outra limitação relevante está relacionada à interpretação textual de conteúdos publicados pelos usuários. Embora o modelo tenha um bom desempenho na identificação de padrões e no reconhecimento de mensagens inadequadas, ainda pode apresentar dificuldades em compreender frases subjetivas ou ambíguas, levando a interpretações equivocadas em determinados contextos.

Além das limitações inerentes à IA, o sistema também depende da estabilidade da API externa do *Gemini AI*, que está sujeita a restrições como limites de requisições diárias e possíveis períodos de instabilidade ou manutenção. Como o *Book Collection* não possui controle direto sobre essas variáveis, falhas temporárias no serviço podem afetar a experiência do usuário.

Por fim, questões relacionadas à privacidade e segurança dos dados devem ser consideradas. Embora o sistema não envolva informações sensíveis dos usuários, as interações com a IA exigem o envio de dados para servidores externos, tornando essencial um monitoramento contínuo para garantir conformidade com boas práticas de segurança.

Com essa integração, o *Book Collection* se beneficia do potencial do *Gemini AI* para enriquecer a experiência do usuário, proporcionando sugestões de leitura mais assertivas e alinhadas aos interesses individuais. No entanto, assim como qualquer ferramenta externa, seu uso exige um planejamento cuidadoso, tanto para garantir a precisão das respostas, quanto para evitar possíveis limitações de uso. Futuramente, melhorias poderão ser implementadas para otimizar ainda mais essa funcionalidade, tornando-a ainda mais eficiente e personalizada.

3.4.8 Conclusão das funcionalidades

A seção de Funcionalidades do sistema *Book Collection* delinea as principais interações e recursos disponíveis para os usuários, desde o *login* seguro e personalização de contas, até a gestão de coleções e interações sociais. Assim, essas funcionalidades serão integradas de maneira coesa, proporcionando uma experiência rica e personalizada. A arquitetura do sistema e as tecnologias utilizadas, conforme apresentado, garantem um ambiente robusto e escalável, pronto para atender às necessidades atuais e futuras dos usuários.

A seguir, será apresentada a seção de *Interfaces do Sistema*, na qual serão exibidas as versões finais das interfaces de usuário para o sistema *Book Collection*, levando em consideração os protótipos, requisitos e funcionalidades elencados.

3.5 Interface Gráfica

A seguir, esta seção abordará conceitos referentes às interfaces do sistema em questão, apresentando o resultado final das interfaces, com base no *design* antes desenvolvido para as mesmas, além de apresentar explicações sobre funcionalidades e o seu fluxo de execução. Sendo assim, a seguir será apresentado as páginas que compõem as interfaces do *Book Collection*.

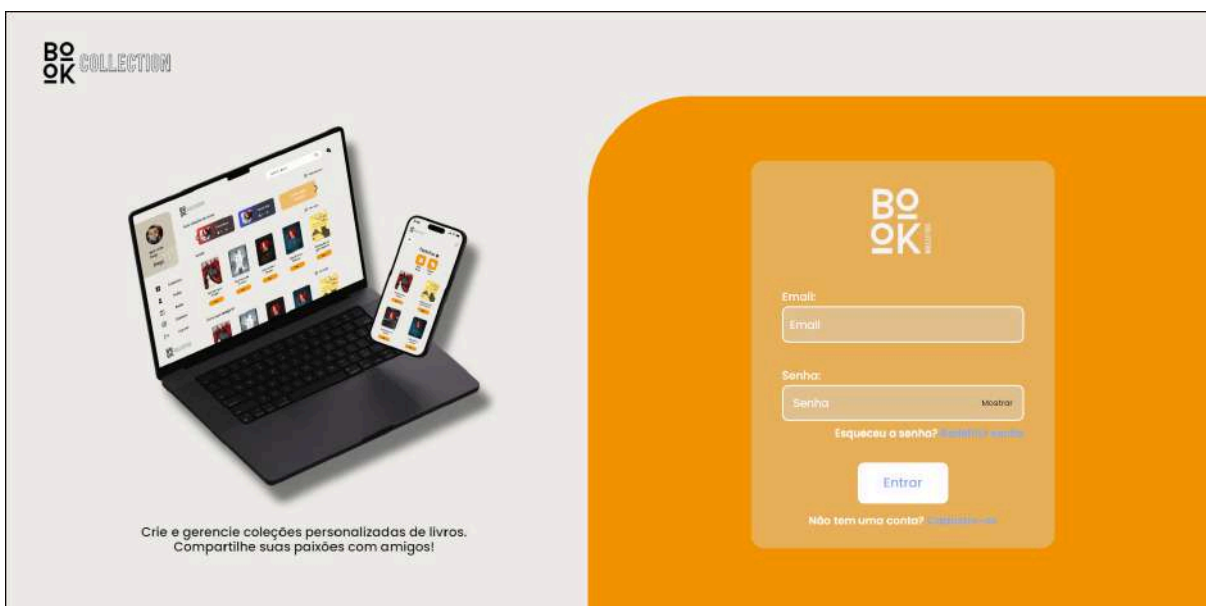
Inicialmente, a paleta de cores do sistema foi cuidadosamente selecionada para proporcionar uma experiência visual agradável e coesa aos usuários, além de serem escolhidas tonalidades que se destacassem e tornassem peça fundamental ao sistema. A cor base, um tom creme suave (#CCC4B1), foi escolhida para criar uma sensação de calma e

elegância, proporcionando um fundo neutro que não distrai o usuário das informações principais, definida pelo seguinte valor hexadecimal: #ECEBE8. Já a cor primária, um vibrante laranja (#F49300), foi escolhida para adicionar vitalidade e destacar elementos importantes da interface, como botões de ação e destaques de texto. As cores complementares em branco e preto são utilizadas para equilibrar o *design* e garantir uma boa legibilidade em todos os contextos. Essa paleta de cores foi pensada para transmitir uma sensação de modernidade e profissionalismo, ao mesmo tempo em que oferece uma experiência visual atraente e acessível aos usuários.

3.5.1 Tela de *Login* e Cadastro

A Tela de *Login*, apresentada na figura abaixo, é a primeira interface com a qual o usuário tem contato ao acessar o sistema. Para utilizar as funcionalidades da plataforma, o usuário deve primeiro realizar o *login*, garantindo sua autenticação e sendo redirecionado para a Tela Inicial, conforme ilustrado na Figura 15.

Figura 12 - Tela de *Login*



Fonte: Elaborado pelo o autor.

Caso um usuário com perfil ativo esqueça ou deseje alterar sua senha, a Tela de *Login* oferece essa opção por meio de um link denominado “*Redefinir senha*”, localizado abaixo do campo de entrada. Ao clicar nesse link, o usuário é redirecionado para a Tela de Redefinição de Senha, conforme apresentado na Figura 13.

Figura 13 - Tela de Redefinição de Senha

A imagem mostra a interface de recuperação de senha do sistema BO OK COLLECTION. No canto superior esquerdo, há o logotipo 'BO OK COLLECTION' e um botão 'Voltar' com uma seta para trás. O formulário principal, intitulado 'Recuperar Senha', está centralizado e possui um fundo laranja. Ele contém os seguintes campos e elementos:

- Um campo de texto rotulado 'Email:' com o placeholder 'Email'.
- Um campo de texto rotulado 'Senha:' com o placeholder 'Senha' e um ícone de olho para alternar a visibilidade, rotulado 'Mostrar'.
- Um campo de texto rotulado 'Confirmar Senha:' com o placeholder 'Confirmar Senha' e um ícone de olho para alternar a visibilidade, rotulado 'Mostrar'.
- Um botão 'Atualizar' localizado abaixo dos campos de senha.

No canto inferior direito da tela, há o logotipo 'BO OK'.

Fonte: Elaborado pelo o autor.

O formulário apresentado na figura anterior permite ao usuário redefinir a senha de seu perfil. Após a alteração, o sistema o redireciona para a Tela de *Login*, onde ele poderá se autenticar com a nova senha.

Por outro lado, conforme mostrado na Figura 12, caso o usuário ainda não tenha um cadastro e deseje criar uma conta, ele pode clicar no link “Cadastre-se”, localizado abaixo do botão “Entrar”. Ao fazê-lo, será direcionado para a Tela de Cadastro, apresentada na Figura 14.

Figura 14 - Tela Cadastro

A imagem mostra a tela de cadastro da plataforma Book Collection. No topo esquerdo, há o logotipo 'BOOK COLLECTION' e um botão 'Voltar' com uma seta para trás. O formulário principal, intitulado 'Criar Conta', está centralizado e possui os seguintes campos: 'Nome' com o placeholder 'Nome', 'Email' com o placeholder 'Email', 'Nome de usuário' com o placeholder 'Nome de usuário', e 'Senha' com o placeholder 'Senha' e um ícone de olho para alternar a visibilidade. Abaixo dos campos, há um botão azul 'Cadastrar' e um link 'Já possui uma conta? Voltar'. No canto inferior direito da tela, o logotipo 'BOOK' é exibido.

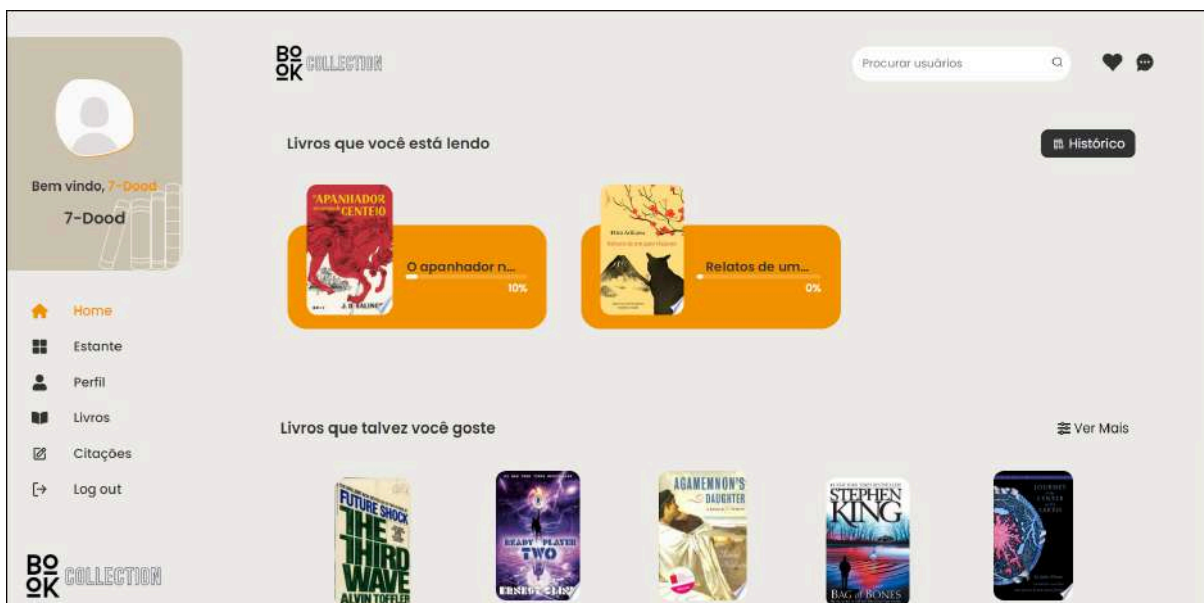
Fonte: Elaborado pelo o autor.

O painel de cadastro, conforme ilustrado na imagem acima, possui um design simples e objetivo, oferecendo um formulário para que o usuário forneça as informações necessárias para a criação de sua conta na plataforma *Book Collection*.

Após o cadastro, o sistema autentica automaticamente o usuário e o redireciona para a Tela Inicial, ilustrada na Figura 15. No entanto, é importante destacar que, por se tratar de um novo usuário, a página inicial exibirá funcionalidades específicas para orientá-lo na exploração da plataforma, conforme será apresentado a seguir na Figura 16 e 17.

3.5.2 Tela Inicial

Figura 15 - Tela Inicial



Fonte: Elaborado pelo o autor.

A Tela Inicial oferece uma interface simples, intuitiva e abrangente, permitindo ao usuário gerenciar suas leituras, acompanhando o seu diário de leitura à medida que avança em um livro. Além disso, a página exibe recomendações personalizadas com base nos gêneros literários de preferência do usuário.

Na parte superior, há um campo de busca para pesquisar outros usuários. Além disso, a tela conta com um ícone em formato de coração, que exibe as notificações do sistema, (Figura 37), além de um outro ícone em formato de balão de conversa, que redireciona o usuário para o *Chapter*, o *chat* da plataforma. Na lateral, um menu vertical oferece acesso rápido às demais páginas do sistema.

Vale destacar que, para novos usuários, o *Book Collection* ainda não possui dados suficientes para personalizar completamente essas funcionalidades, especialmente no que diz respeito às preferências literárias. Para solucionar isso, o sistema implementa uma funcionalidade específica para coletar informações sobre os gostos do usuário, conforme ilustrado nas figuras a seguir.

Figura 16 - Tela Inicial para novos Usuários



Fonte: Elaborado pelo o autor.

A figura acima ilustra a primeira interação de novos usuários com o sistema *Book Collection*. Para aprimorar a experiência inicial, o sistema oferece uma funcionalidade que permite ao usuário selecionar até 10 livros a partir de uma lista previamente definida. Essa seleção é gerada com o auxílio das ferramentas externas *Google Books API* e *Gemini AI*, já mencionadas anteriormente. A Figura 17 a seguir apresenta essa funcionalidade de escolha de livros em mais detalhes.

Figura 17 - Pesquisa de Interesses de Leitura

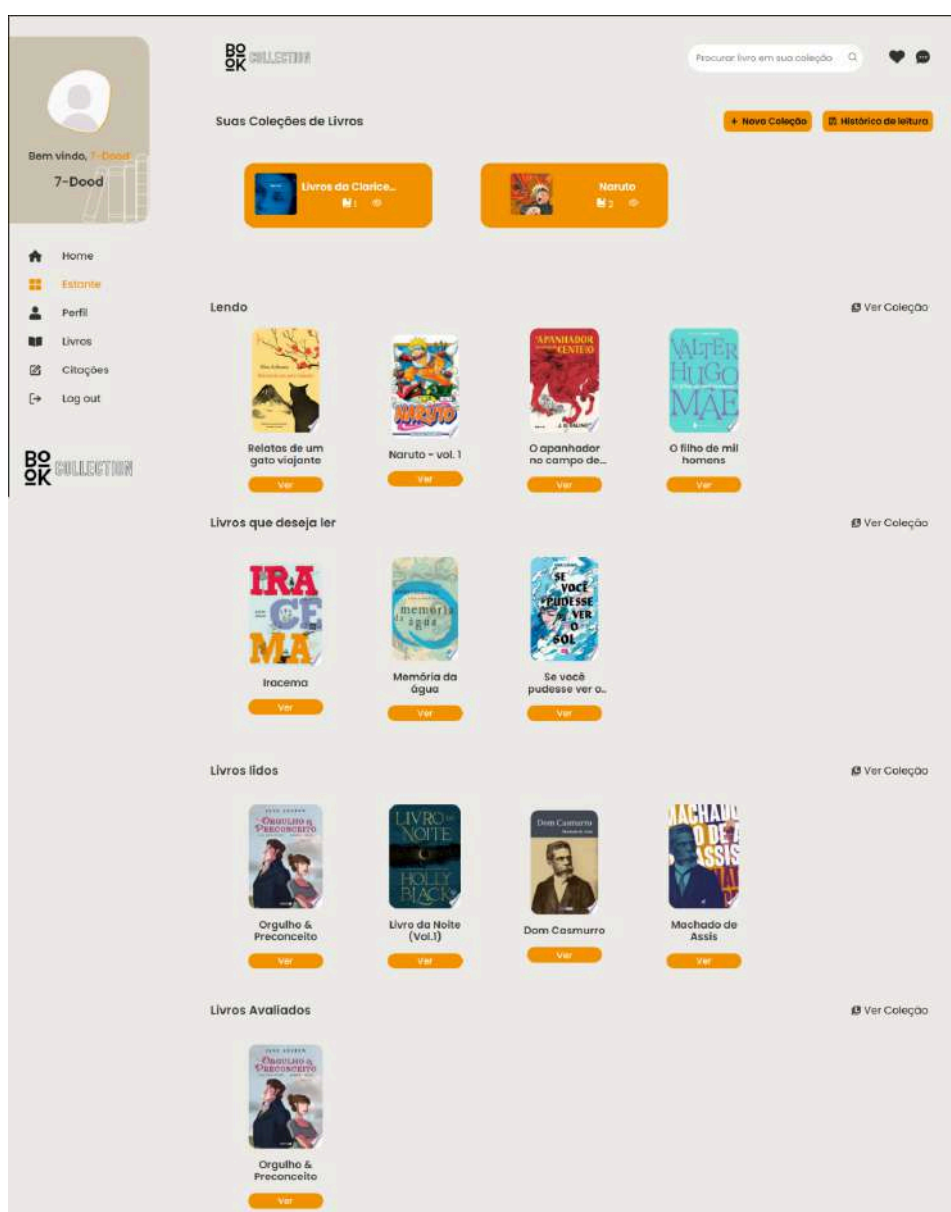


Fonte: Elaborado pelo o autor.

A figura acima apresenta uma visão parcial da funcionalidade de seleção de livros. A lista inicial contém aproximadamente 30 títulos, abrangendo diversos gêneros e autores. Para tanto, o sistema solicita que o usuário escolha alguns livros, até no máximo 10 títulos, e, com base nas escolhas do usuário, o sistema identifica suas principais preferências e ajusta as recomendações para melhor atender aos seus interesses.

3.5.3 Tela de Estante e de Coleção de livros

Figura 18 - Tela de Estante Virtual



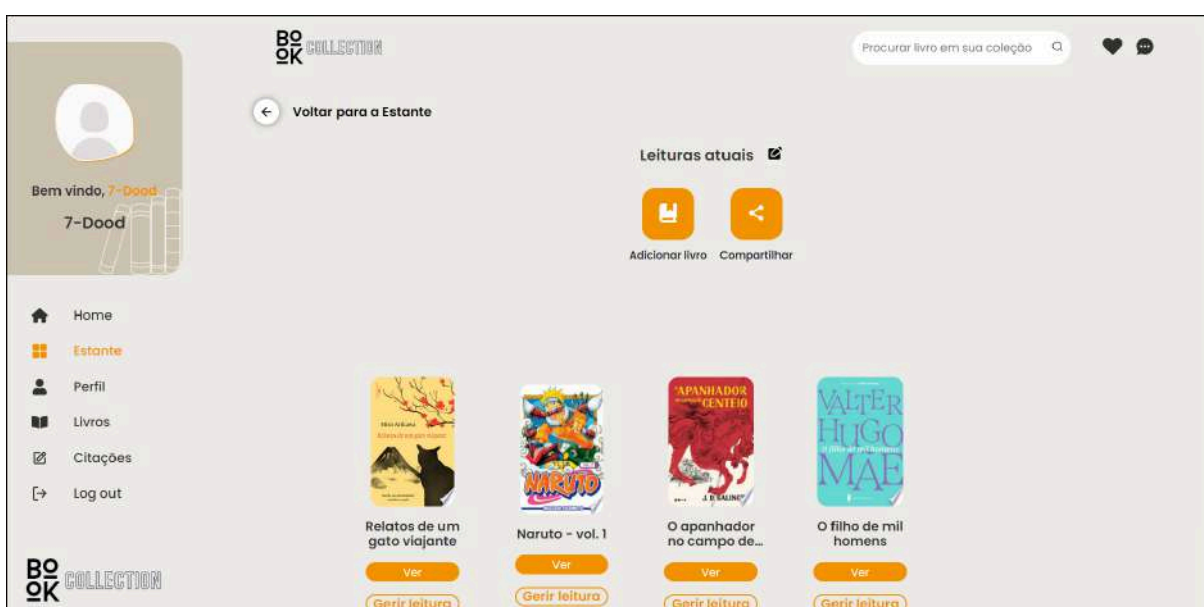
Fonte: Elaborado pelo o autor.

A Tela de Estante Virtual proporciona uma visão geral da organização dos livros na estante do usuário, exibindo-os conforme as coleções em que foram alocados. Inicialmente, são apresentadas as coleções criadas pelo próprio usuário, acompanhadas da opção de adicionar uma nova coleção. Além disso, a tela exibe as coleções padrão do sistema.

Caso o usuário deseje criar uma nova coleção, basta clicar no componente correspondente, que ativará um formulário na parte inferior da página para preenchimento. Outra funcionalidade disponível é o acesso ao histórico de diários de leitura. Para isso, o usuário pode clicar em "Histórico de leitura", sendo redirecionado para a página específica dessa funcionalidade, conforme ilustrado na Figura 33.

A Figura 18 foca exclusivamente na estrutura organizacional da estante. No entanto, caso o usuário queira obter mais informações sobre um livro específico, pode clicar no botão "Ver", que o levará à página de detalhes do livro (Figura 27). Já para explorar com mais profundidade os livros de uma coleção específica, basta selecionar a coleção desejada ou, no caso das coleções padrão, clicar no botão "Ver coleção"; isso o redirecionará para a página correspondente da coleção de livros, conforme demonstrado na Figura 19.

Figura 19 - Tela de Coleção de Livros



Fonte: Elaborado pelo o autor.

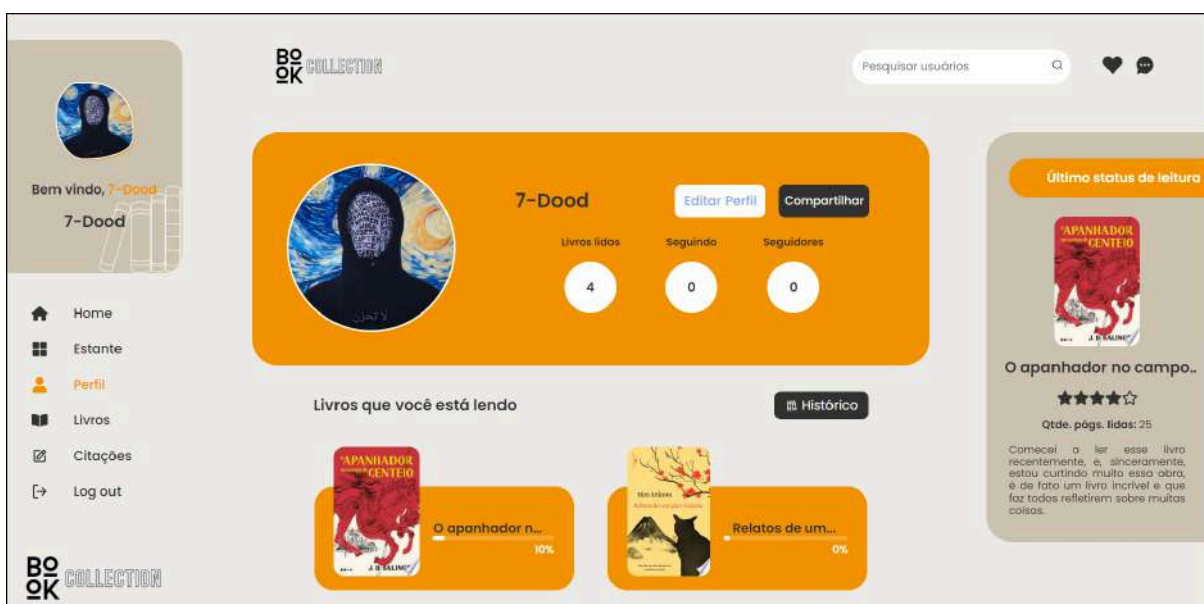
A Figura 19 apresenta um exemplo de como as informações de uma coleção de livros serão exibidas na estante virtual do usuário. Nessa interface, além da exibição dos livros

pertencentes à coleção, o usuário terá acesso a funcionalidades adicionais, como a opção de adicionar um novo livro.

3.5.4 Telas de Perfil

A Tela de Perfil, ilustrada a seguir, apresenta o painel de informações do usuário, exibindo dados como a quantidade de livros lidos, conforme a coleção padrão "Lidos", além de informações sobre seguidores e perfis seguidos. Também inclui o histórico de leitura do usuário, detalhando os registros e gerenciamentos de leitura realizados.

Figura 20 - Tela de Perfil



Fonte: Elaborado pelo o autor.

A interface apresentada na Figura 20 conta ainda com três seções distintas de publicações: uma dedicada às avaliações de livros, outra para citações, — que são textos publicados pelo usuário no *Book Collection* —, e uma terceira destinada à exibição de livros lidos e coleções criadas com visibilidade pública. A distinção entre essas categorias é explorada com mais detalhes nas figuras seguintes.

A Figura 21 apresenta a seção dedicada às avaliações realizadas pelo usuário. As avaliações são exibidas em ordem decrescente, sempre destacando a mais recente no topo. Além disso, a interface conta com um filtro de paginação, permitindo a apresentação entre diferentes avaliações.

Figura 21 - Publicação de usuário: Avaliações



Fonte: Elaborado pelo o autor.

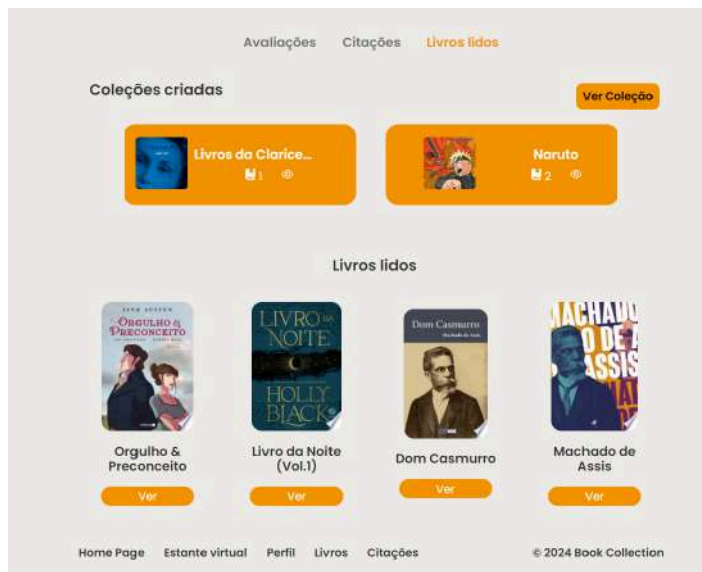
Já a Figura 22 representa a seção destinada às publicações do usuário. Assim como na seção de avaliações, as publicações são exibidas em ordem decrescente, acompanhadas de um filtro de paginação. Além disso, cada publicação exibe informações complementares, como curtidas e comentários, além de ações que permitem ao usuário manipular suas informações.

Figura 22 - Publicação de usuário: Citações



Fonte: Elaborado pelo o autor.

Por fim, a Figura 23 ilustra a seção de coleções do usuário, onde são exibidos os livros lidos e as coleções personalizadas criadas para organizar suas leituras. As coleções que foram marcadas como públicas ficam disponíveis para visualização por outros usuários.

Figura 23 - Publicação de usuário: Livros lidos

Fonte: Elaborado pelo o autor.

Além disso, as figuras acima demonstram as diferenças entre os três modos de exibição de conteúdos na Tela de Perfil, de acordo com a opção escolhida pelo usuário. Outra funcionalidade relevante é a possibilidade de editar as informações do perfil. A figura a seguir ilustra como o sistema responde a uma solicitação de edição.

Figura 24 - Tela de edição de perfil

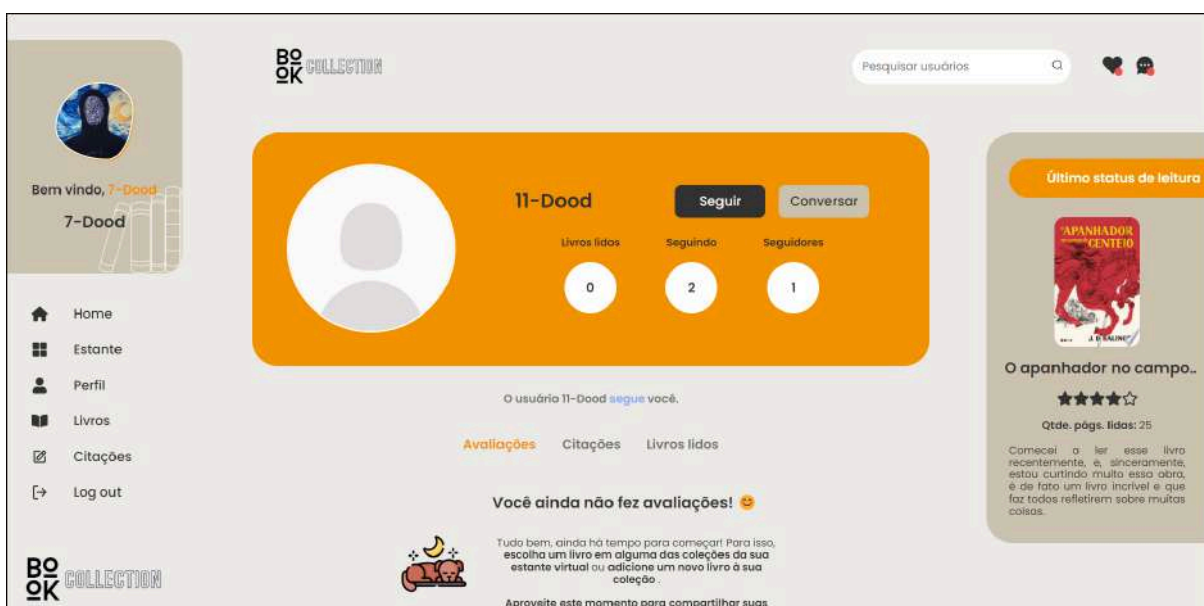
The screenshot shows the 'Edição de perfil' (Profile Editing) screen. At the top left, there is a 'Voltar' (Back) button. The header includes the 'BOOK COLLECTION' logo and the title 'Edição de perfil'. Below the title is a circular placeholder for a profile picture with an 'Escolher imagem' (Choose image) button underneath. The form contains three input fields: 'Seu Nome:' with the value 'Douglas da Silva Araújo', 'Nome de Perfil:' with the value '7-Dood', and 'Seu E-mail:' with the value 'testOne@gmail.com'. An 'Editar Perfil' (Edit Profile) button is located at the bottom of the form.

Fonte: Elaborado pelo o autor.

Como mostrado na figura acima, um formulário simples e intuitivo é exibido em uma nova página de edição, permitindo que o usuário visualize e atualize as informações do seu perfil de maneira fácil e eficiente. Assim, o perfil do usuário é atualizado conforme as alterações realizadas.

Os mesmos conceitos da Tela de Perfil se aplicam à visualização do perfil de outros usuários, com algumas diferenças. Além das informações do perfil, são apresentados botões para seguir ou iniciar uma conversa com o usuário em questão. A Figura 25 ilustra essas diferenças sutis na apresentação da página de perfil.

Figura 25 - Tela de Perfil de um Amigo



Fonte: Elaborado pelo o autor.

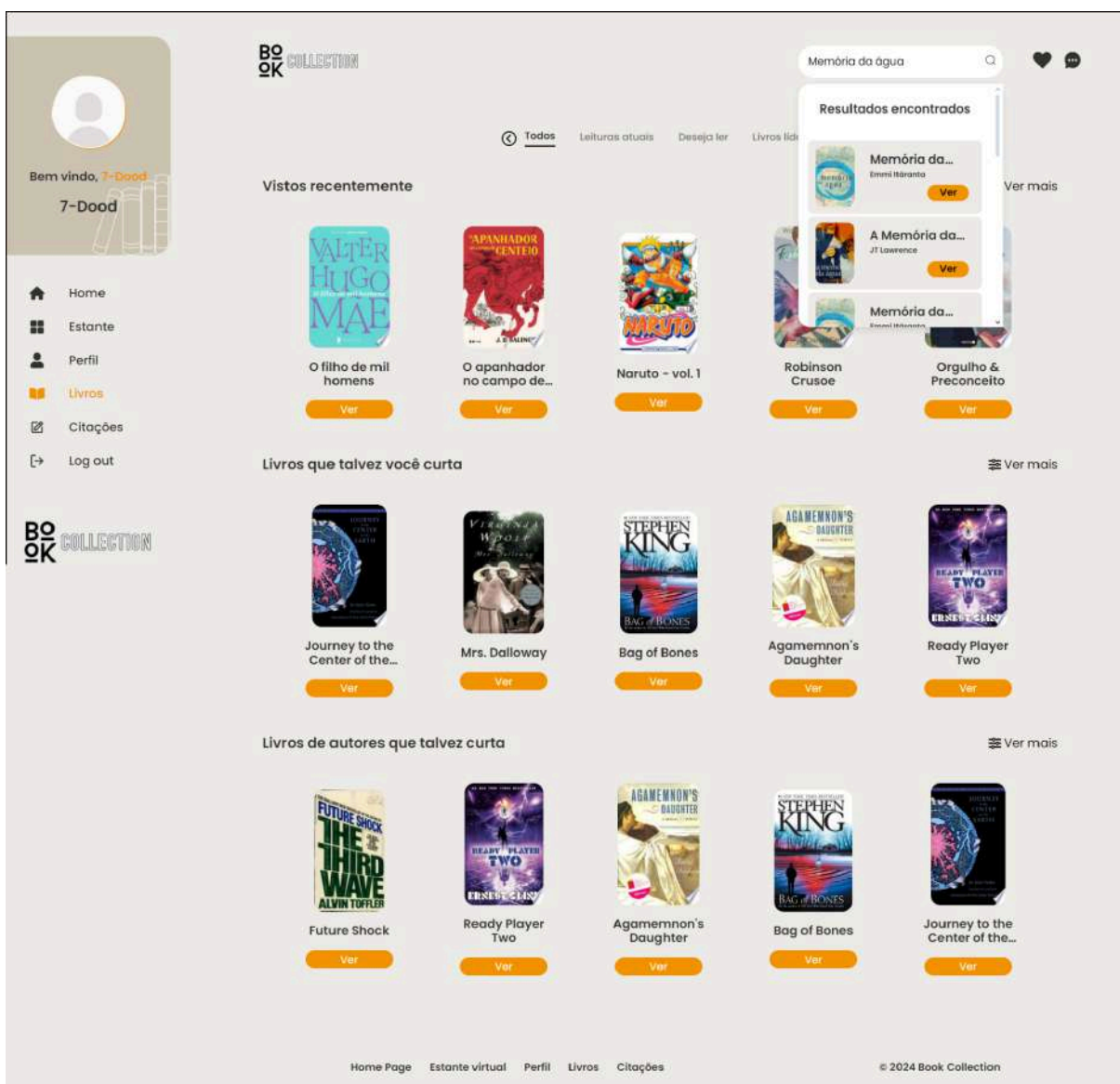
Na figura 25 é apresentada a Tela de Perfil de um amigo, nessa tela um usuário não possui tantas funcionalidades, quanto na sua tela, uma vez que nesse contexto, somente é possível analisar as informações gerenciadas e apresentadas pelo usuário dono do perfil. Além disso, tem-se a possibilidade de observar as coleções de um usuário, desde que essas sejam marcadas como “públicas”, permitindo que os perfis analisem os gostos de outros usuários.

3.5.5 Telas de Livros

A Tela de Pesquisa de Livros oferece uma visão geral para a busca e descoberta de novas leituras. Utilizando informações da API do *Google Books*, esta página exibe um conjunto de livros alinhados aos interesses do usuário, facilitando a exploração de novas

obras. O usuário pode realizar pesquisas diretamente no campo de busca, enviando uma solicitação ao *backend* do sistema *Book Collection*, que consulta instantaneamente o banco de dados da API do *Google Books*.


Figura 26 - Tela de Pesquisa de Livros



Fonte: Elaborado pelo o autor.

Após realizar a busca do livro, ou encontrar um livro do seu interesse, o usuário pode clicar no botão "Ver", sendo redirecionado para uma página com detalhes completos sobre o livro, conforme ilustrado na figura a seguir.

Figura 27 - Tela de Informações de um Livro



Bem vindo, 7-Dood

7-Dood

- Home
- Estante
- Perfil
- Livros**
- Citações
- Log out


BOOK COLLECTION

Procurar livro

← Voltar

Informações sobre o livro

Salva em - Deseja ler



Se você pudesse ver o sol

Ann Liang

Gênero principal: Young Adult Fiction, Romance, Romantic Comedy

Gênero secundário: Young Adult Fiction, Fantasy, Contemporary

Editora: Alt

Sinopse

Uma garota com o poder de ficar invisível descobre os segredos mais escandalosos da elite chinesa

Da autora best-seller Ann Liang, de *Dessa vez é real*

Alice Sun sempre se sentiu invisível no Internato Internacional de Airington, onde é a única estudante bolsista entre os adolescentes mais ricos e influentes de Pequim. Ela se dedica para ser a melhor aluna da turma e garantir um futuro digno para sua família.

Até que algo muito estranho acontece: Alice começa a ficar literalmente invisível. Invisível de verdade! E quando seus pais revelam que não vão mais conseguir pagar o colégio — mesmo com a bolsa de estudos — ela percebe que pode usar esse novo poder a seu favor.


Com acesso ilimitado a todos os segredos dos estudantes, Alice oferece seus serviços de invisibilidade para descobrir qualquer coisa que seus colegas desejam saber — em troca de um pagamento, é claro!

Mas quando seu grande rival acadêmico se torna seu parceiro de negócios e as tarefas escalam de pequenos escândalos para crimes de verdade, as coisas começam a sair de controle. Agora, Alice precisa decidir se vale a pena arriscar sua dignidade para ajudar a família e permanecer no internato.

Se você pudesse ver o sol é um romance contemporâneo com toque fantástico que, em breve, ganhará uma adaptação para série.

Sua avaliação

Que tal avaliar esse livro?




Olá, parece que você **ainda não avaliou** esse livro. Que tal avaliá-lo agora? Para isso, basta **adicionar esse livro a alguma das suas coleções**, e em seguida poderá **avaliar esse livro**.

Lembre-se, seu comentário pode fazer com que esse livro se torne uma ótima experiência para outro leitor.

Avaliações desse livro:

Seja o primeiro(a) a comentar!



Olá, parece que esse livro **ainda não possui avaliações**. Seja o primeiro a avaliar esse livro. Para isso, basta **adicionar esse livro a alguma das suas coleções**, e em seguida poderá **avaliar esse livro**.

Lembre-se, seu comentário pode fazer com que esse livro se torne uma ótima experiência para outro leitor.

Home Page

Estante virtual

Perfil

Livros

Citações

© 2024 Book Collection

Fonte: Elaborado pelo o autor.

A Tela de Informações de um Livro oferece uma visão detalhada sobre o livro, incluindo dados como a quantidade de páginas, o autor e a sinopse. Além disso, a página apresenta um resumo das avaliações feitas pelos usuários sobre a obra. Uma funcionalidade importante é a opção de adicionar o livro à coleção do usuário, permitindo sua categorização conforme as preferências do leitor.

Por fim, a tela exibe uma lista de livros com atributos semelhantes ao livro em exibição, proporcionando mais opções de leitura. Para adicionar um livro a uma coleção, o usuário pode clicar no ícone de "Marcador de Páginas", localizado na lateral acima do título do livro, o que abrirá uma janela virtual para selecionar a coleção onde se deseja adicionar o livro, conforme ilustrado na figura abaixo.

Figura 28 - Seleção de Coleção



Fonte: Elaborado pelo o autor.

Vale destacar que, ao acessar um livro já adicionado à coleção, a mesma página de informações será exibida, mas com um ambiente de configuração diferenciado. As próximas figuras ilustram essa diferença no ambiente de configuração da página.

A Figura 29 ilustra o cenário em que um determinado livro ainda não foi adicionado a nenhuma coleção do usuário. Nesse caso, além de visualizar as informações do livro, o usuário tem como única ação disponível adicioná-lo a uma coleção. Isso ocorre porque, dentro da lógica do Book Collection, avaliações e comentários só podem ser feitos para livros que estejam na coleção do usuário. Essa abordagem reflete a ideia de que, no contexto físico, para avaliar um livro, pressupõe-se que ele tenha sido lido ou, pelo menos, organizado dentro de uma estante pessoal. Assim, a adição do livro a uma coleção representa um passo essencial para sua categorização e interação dentro do sistema.

Figura 29 - Funcionalidades de livros não presentes na coleção



Fonte: Elaborado pelo o autor.

Por outro lado, a Figura 30 ilustra o cenário em que um livro já está presente em uma ou mais coleções do usuário. Nesse caso, além de visualizar as coleções nas quais o livro já foi adicionado, o sistema oferece as opções de adicioná-lo a novas coleções ou removê-lo de algumas delas. Além disso, um ícone de ação é exibido, permitindo que o usuário acesse a funcionalidade de avaliação do livro. Dessa forma, o ambiente de gerenciamento de coleções torna-se dinâmico e intuitivo, garantindo que o usuário tenha total controle sobre sua biblioteca digital e suas interações com cada livro.

Figura 30 - Funcionalidades de livros presentes na coleção



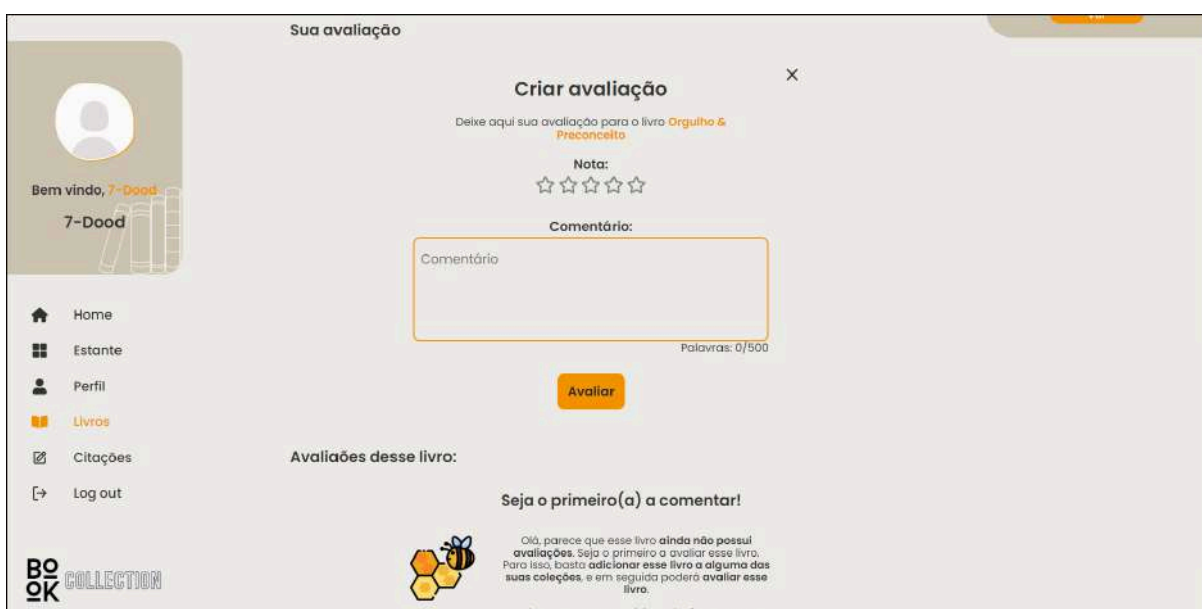
Fonte: Elaborado pelo o autor.

Como mostrado nas figuras acima, a transição entre os ambientes é clara e intuitiva devido à presença dos ícones. Quando um livro não está na coleção de um usuário, mas o usuário deseja adicioná-lo, basta clicar no ícone de "salvar", representado por um ícone de

marca-páginas. Isso abrirá o componente onde o usuário poderá selecionar em qual coleção adicionar o livro.

Caso o livro já esteja na coleção, o ambiente de seleção exibe as coleções nas quais o livro se encontra, permitindo ao usuário removê-lo facilmente de qualquer uma dessas coleções. Além disso, outra funcionalidade importante é a possibilidade de o usuário avaliar o livro, utilizando o ícone de "Comentário". Para isso, o livro precisa estar em alguma das coleção do usuário. A figura a seguir ilustra o ambiente de avaliação do livro.

Figura 31 - Avaliar Livro



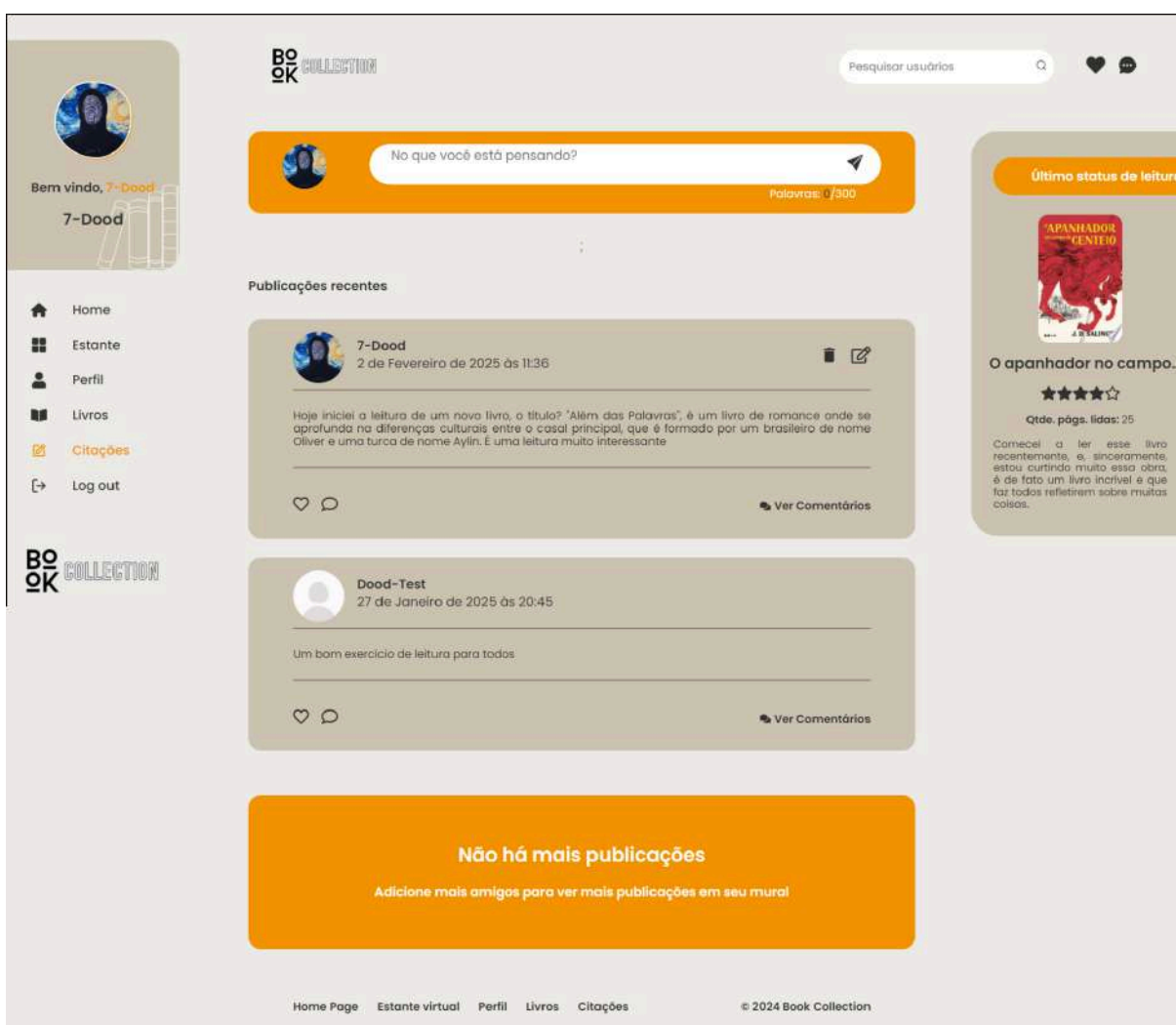
Fonte: Elaborado pelo o autor.

Ao selecionar a opção de avaliar um livro, o sistema exibe um formulário na parte inferior da página, com informações breves sobre o livro escolhido pelo usuário. O formulário inclui um campo para que o usuário insira um resumo do seu comentário sobre o livro, além de um campo de "estrelas", onde ele pode atribuir uma nota ao livro. Após a avaliação, o livro é automaticamente adicionado à coleção padrão de livros chamada "Avaliados". Esta coleção não pode ser modificada pelo usuário e é atualizada apenas quando uma avaliação é criada ou removida. Caso o usuário deseje avaliar um livro enquanto o lê, ele pode utilizar a funcionalidade de gerenciamento de leitura, conforme mostrado nas Figuras 34 e 35.

3.5.6 Tela de Citações

A Figura 32 ilustra como o mural de publicações criadas pelos usuários é exibido. Este mural é visível para outros perfis com os quais há uma "Amizade", ou seja, aqueles com os quais o usuário está conectado por meio da funcionalidade de seguir e ser seguido, ou pelas as publicações realizadas até aquele momento. As publicações no mural oferecem um espaço para interação, permitindo que os usuários curtam ou comentem nas publicações de outros. As citações postadas no dia são organizadas por data e hora. Além de visualizar as publicações, o usuário também tem à disposição um campo para inserir suas próprias citações ou textos que deseje compartilhar.

Figura 32 - Tela Citações

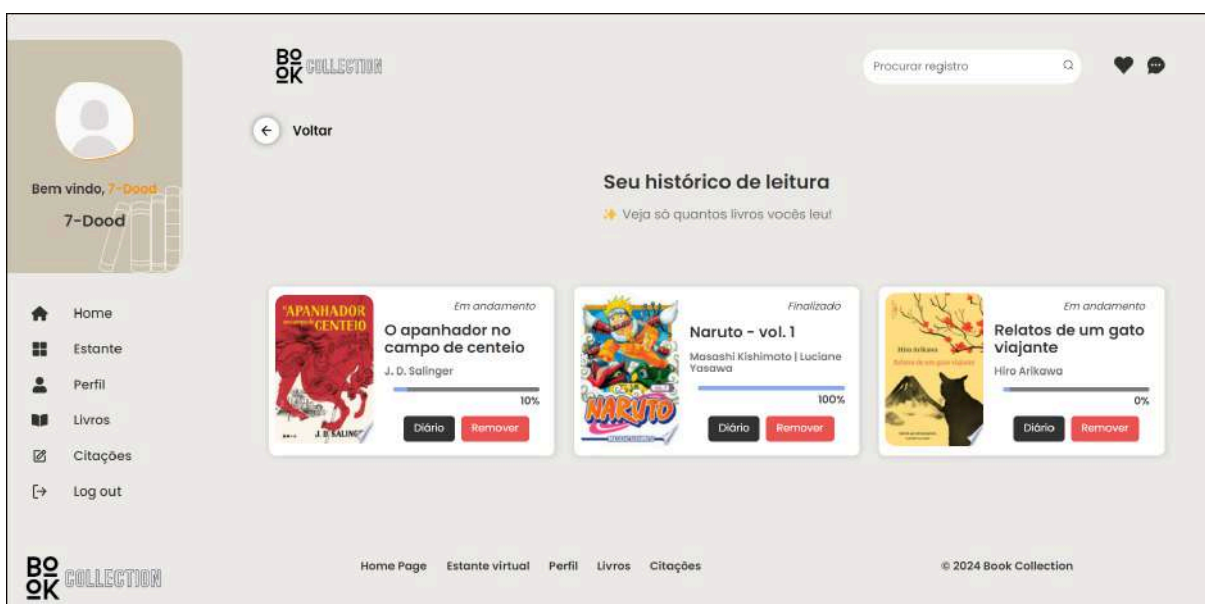


Fonte: Elaborado pelo o autor.

3.5.7 Telas de Gerenciamento de leitura

O gerenciamento de leitura, no contexto do *Book Collection*, é entendido como o ambiente onde o usuário pode registrar seu progresso e realizar avaliações em tempo real de suas leituras, funcionando como um diário de leitura digital. Esses registros de leitura são um dos primeiros componentes apresentados ao usuário na Tela Inicial, conforme ilustrado na Figura 15. Além dessa visualização inicial, todos os registros são centralizados em uma página específica, que reúne o histórico completo de diários de leitura e registros do usuário, conforme apresentado na figura a seguir.

Figura 33 - Tela de Histórico de Gerenciamento de Leitura



Fonte: Elaborado pelo o autor.

A figura acima exibe o histórico de registros de leitura, que oferece diversas funcionalidades, incluindo a opção de visualizar os registros ao clicar no componente denominado "Diário". Ao selecionar essa opção, uma janela virtual é aberta, exibindo uma lista com avaliações e o progresso de leitura correspondente ao registro. Nessa janela, são apresentadas informações como o título do livro, a data da avaliação, a porcentagem da leitura concluída, a quantidade de páginas lidas até o momento e o relato detalhado da avaliação.

Existem dois formatos de visualização: um para livros em andamento, ou seja, aqueles que o usuário ainda está lendo, permitindo a adição de novos registros ou a alteração do último registro realizado, conforme ilustrado na Figura 34. Por outro lado, se um registro público for acessado por um usuário que não seja o responsável pelo gerenciamento de

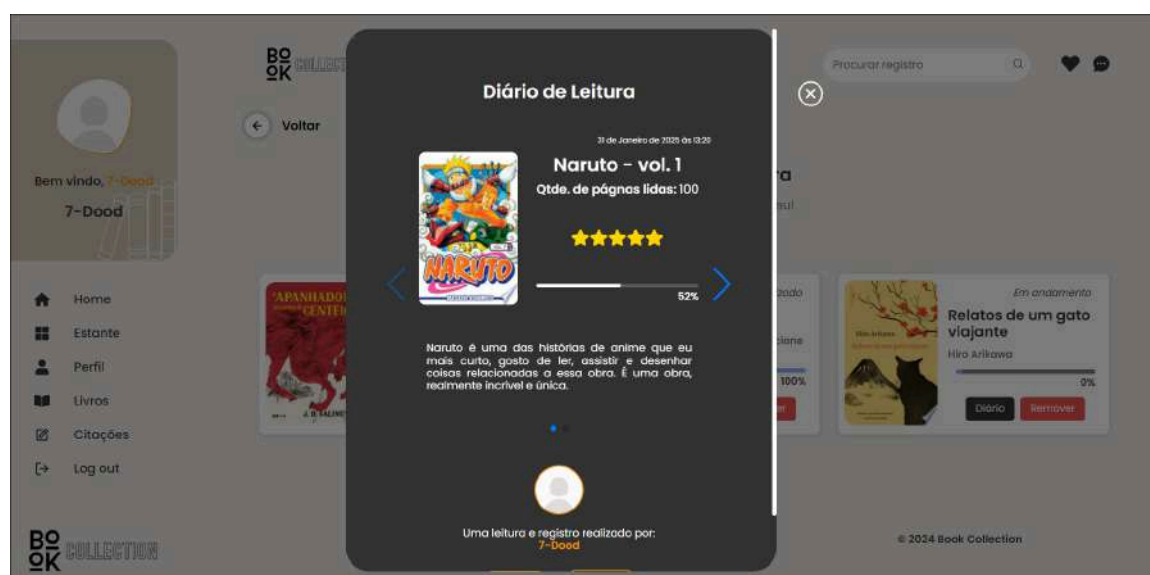
leitura, o sistema exibe uma janela virtual com menos funcionalidades, permitindo apenas a visualização dos registros, como mostrado na Figura 35.

Figura 34 - Janela Virtual: Diário de Leitura em andamento



Fonte: Elaborado pelo o autor.

Figura 35 - Janela Virtual: Visualização pública de um Diário de Leitura

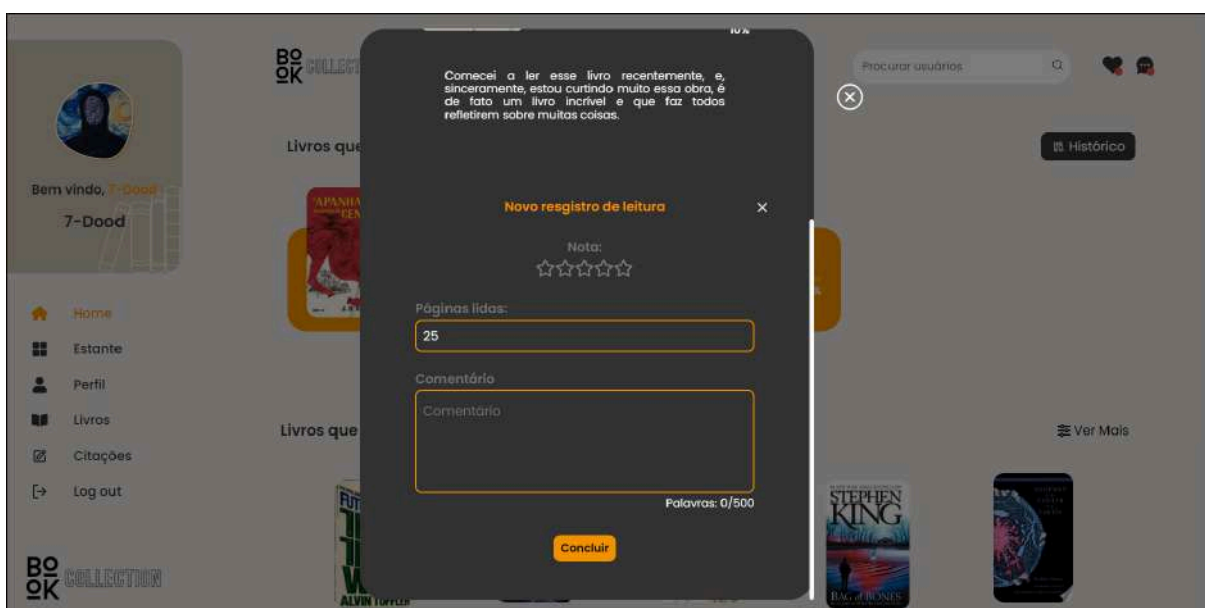


Fonte: Elaborado pelo o autor.

As janelas de gerenciamento podem ser acessadas tanto pela Tela Inicial, conforme ilustrado na Figura 15, quanto pela Tela de Perfil (Figura 20), ou pela Tela de Histórico

(Figura 33). Funções como editar ou remover um registro estão disponíveis apenas para o último registro aplicado pelo usuário em determinada obra. Além disso, ao clicar em "Novo status", o sistema exibe um formulário para a criação de um novo registro. A figura a seguir ilustra esse cenário.

Figura 36 - Janela Virtual: Atualização de Diário de Leitura

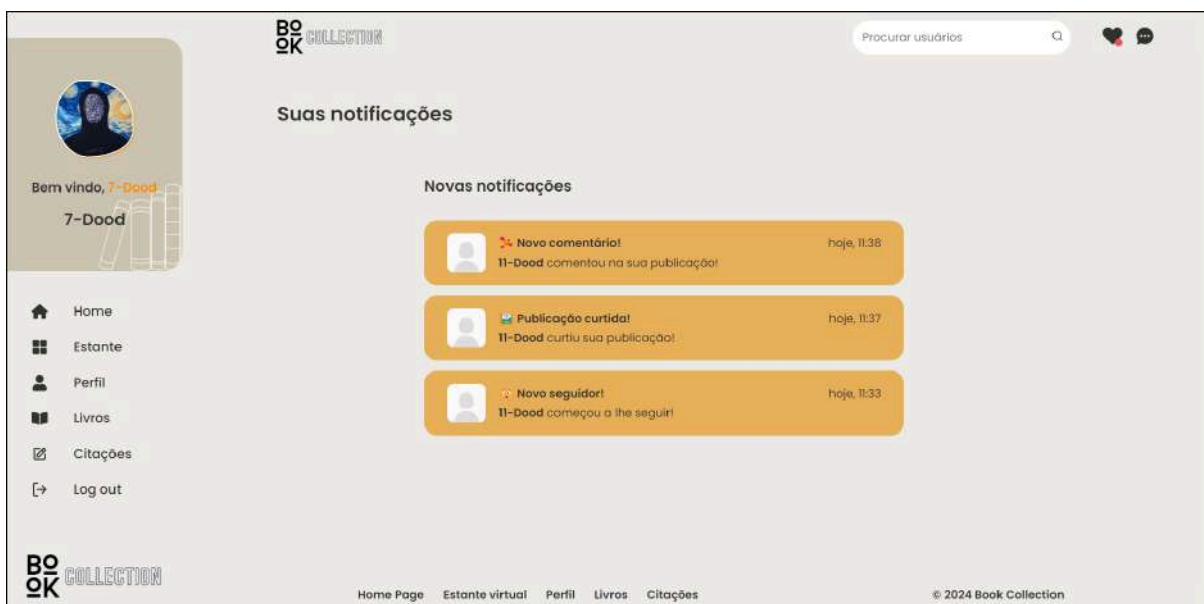


Fonte: Elaborado pelo o autor.

3.5.8 Tela de Notificações

A Tela de Notificações exibe as atualizações do sistema, apresentando as notificações geradas a partir de eventos emitidos e recebidos pela API Orientada a Eventos. Essas notificações incluem mensagens recebidas, curtidas nas publicações dos usuários e interações de seguimento entre usuários. A figura a seguir ilustra como essa tela é apresentada no sistema.

Figura 37 - Tela de Notificações

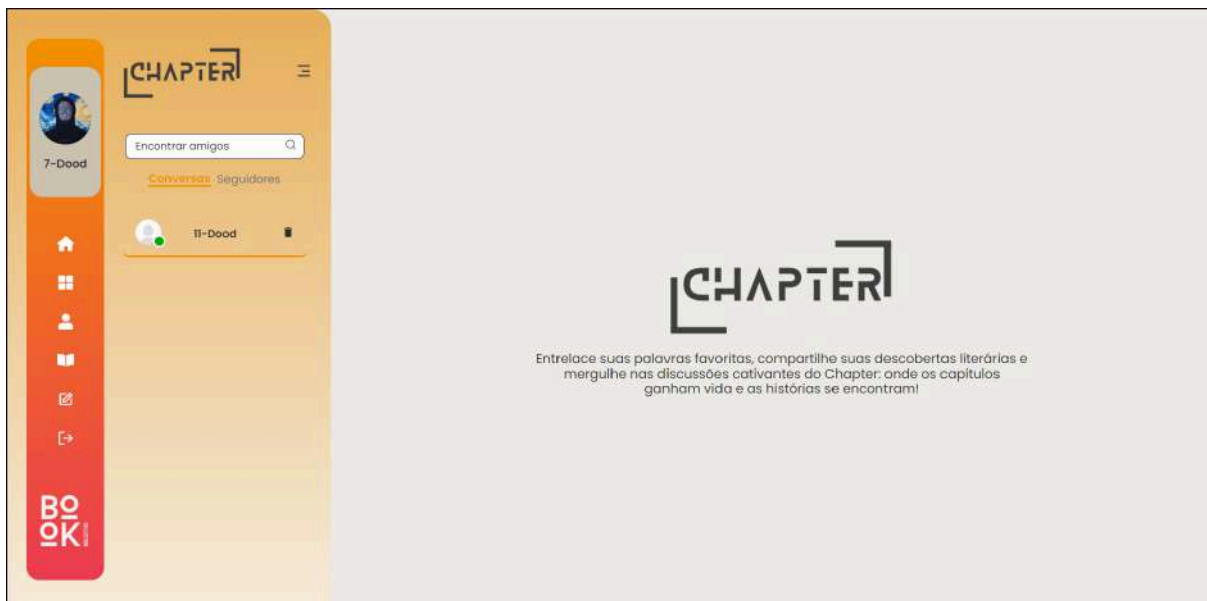


Fonte: Elaborado pelo o autor.

3.5.9 Chapter

A figura a seguir apresenta o *Chapter*, o sistema de gerenciamento de conversas e interações entre os usuários. Ao acessar o ambiente de *chat*, essa será a primeira tela visível ao usuário. Na lateral, há um novo menu simplificado, que ocupa menos espaço, se comparado com o menu disponível nas demais páginas. Além disso, há uma lista de usuários que é exibida com base nos seguidores de um determinado usuário ou em conexões anteriores. Ademais, por meio das funcionalidades assíncronas é possível identificar os usuários que estão *online*, sendo assim apresentados na lista de usuários.

Figura 38 - Tela Inicial do *Chapter*



Fonte: Elaborado pelo o autor.

O cenário de interação entre dois usuários no chat do *Chapter* é mostrado na figura seguinte. As mensagens destacadas em tons de laranja, alinhadas à direita, representam as enviadas pelo proprietário do perfil acessado. As mensagens em tom mais claro, posicionadas à esquerda, correspondem às enviadas pelo destinatário da conversa.

Figura 39 - Tela Conversa com um Amigo



Fonte: Elaborado pelo o autor.

3.5.10 Conclusão sobre as Interfaces do Sistema

A prototipação das interfaces do sistema *Book Collection*, antes realizada, desempenhou um papel crucial no desenvolvimento das interfaces e refinamento das funcionalidades do sistema de sua implementação final. Através dos protótipos, foi possível identificar e ajustar detalhes importantes, garantindo que a experiência do usuário seja intuitiva e agradável.

A escolha cuidadosa da paleta de cores e o *design* focado em usabilidade foram fundamentais, para criar uma interface que não apenas atende às necessidades dos usuários, mas também proporciona uma experiência visualmente atraente.

A prototipação também permitiu mapear o fluxo de execução das funcionalidades, assegurando que todas as interações sejam fluidas e coerentes. Assim, a etapa de prototipação não só validou as decisões de *design*, como também preparou o terreno para o desenvolvimento, assegurando uma base sólida para a implementação do sistema.

4 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo principal o desenvolvimento e a estruturação completa do sistema *Book Collection*, utilizando uma combinação de tecnologias modernas e eficientes, como *React*, *TypeScript*, *PostgreSQL*, *Prisma ORM*, *Redis*, *Google Books API* e *Gemini AI*. A partir de uma revisão de trabalhos relacionados e uma pesquisa pública, responsáveis pela coleta e desenvolvimento dos requisitos, além de uma abordagem metodológica bem definida, foi possível criar uma primeira versão robusta e escalável do sistema, atendendo com eficácia às necessidades dos usuários.

Os resultados alcançados incluem a implementação bem-sucedida de todas as funcionalidades previstas, como o gerenciamento de estantes virtuais, controle de leitura, publicações de notas e pesquisas de livros, alinhadas às melhores práticas observadas em sistemas de gerenciamento de leitura. A escolha das tecnologias não só possibilitou um desenvolvimento ágil e eficiente, mas também garantiu um desempenho satisfatório para a operação do sistema no longo prazo. A utilização do *Jira* para a gestão dos requisitos contribuiu para um controle preciso e o acompanhamento eficaz de todas as etapas do projeto.

Entre as principais contribuições deste trabalho, destaca-se a entrega de um sistema completo e funcional, promovendo uma série de funcionalidades e servindo como base sólida para futuras melhorias e expansões. O *Book Collection* oferece uma solução prática e bem estruturada para o gerenciamento de coleções de livros, cumprindo com sucesso o objetivo de atender às demandas de quem busca uma maneira mais organizada e interativa de gerenciar suas leituras.

Apesar dos avanços alcançados, este trabalho apresenta algumas limitações que podem afetar o desempenho e a escalabilidade do sistema no longo prazo. Uma das principais limitações está na dependência de APIs externas, como a API do *Google Books*, que impõe restrições no número de requisições por unidade de tempo. Este limite pode impactar diretamente a experiência do usuário, especialmente quando o sistema atingir um número significativo de usuários ou quando forem necessárias consultas frequentes à base de dados de livros. Para mitigar esse impacto, seria essencial estabelecer uma parceria com o *Google*, a fim de garantir uma maior flexibilidade nas requisições, ou, alternativamente, desenvolver uma API própria, capaz de fornecer dados exclusivamente para o sistema, reduzindo a dependência de terceiros e assegurando maior controle sobre a qualidade e a disponibilidade das informações.

Outro fator que pode gerar desafios no futuro é o uso da *Gemini AI*. Embora tenha contribuído significativamente para a inteligência do sistema, a dependência de uma plataforma externa para processamento e análise de dados pode limitar a personalização e a escalabilidade das funcionalidades. Para garantir maior autonomia e flexibilidade, seria interessante explorar soluções próprias de inteligência artificial ou parcerias com outras plataformas que atendam às necessidades do sistema de maneira mais eficiente e personalizada.

Além disso, a arquitetura do sistema, embora robusta, ainda está em uma estrutura modular, o que em certa parte, já divide as responsabilidades do sistema, mas está longe do ideal. No futuro, para garantir a escalabilidade e a resiliência do sistema à medida que ele cresce, seria recomendável migrar para uma arquitetura distribuída. A implementação de uma arquitetura baseada em microsserviços permitiria uma maior flexibilidade na gestão de diferentes partes do sistema, facilitando a manutenção e a implementação de novas funcionalidades, além de permitir um crescimento mais sustentável e eficiente à medida que a base de usuários se expande.

Portanto, para trabalhos futuros, recomenda-se explorar novas tecnologias e abordagens para aprimorar a usabilidade, a eficiência, a segurança e a autonomia do sistema, além de considerar a implementação de novas funcionalidades com base no *feedback* dos usuários. A evolução do sistema, a partir dessas limitações, pode resultar em um produto mais robusto, escalável e capaz de atender a um público maior e mais diversificado.

Pessoalmente, o desenvolvimento deste TCC foi uma experiência extremamente enriquecedora, proporcionando um aprendizado profundo sobre as tecnologias adotadas e o processo de criação de sistemas complexos. Através deste projeto, não apenas aprimorei minhas habilidades técnicas, mas também desenvolvi competências em pesquisa, resolução de problemas e gestão de projetos. Essas experiências me prepararam de maneira sólida para enfrentar os desafios que surgem no campo do desenvolvimento de *software*, com uma visão mais ampla sobre a construção de sistemas escaláveis e eficientes.

Em síntese, o sistema *Book Collection*, como descrito nas fases anteriores deste trabalho, representa uma contribuição significativa para o gerenciamento de coleções de livros, atendendo às necessidades dos usuários e abrindo caminho para futuras expansões e melhorias. A conclusão desta fase do projeto marca a finalização da primeira versão do sistema, consolidando as funcionalidades e resultados planejados, e permitindo uma avaliação real de sua eficácia prática. A próxima fase, agora focada em testar e aprimorar o sistema, possibilitará ajustes baseados no *feedback* dos usuários, garantindo que o *Book Collection* evolua de acordo com as demandas reais do público e se torne uma ferramenta ainda mais robusta e acessível.

5 REFERÊNCIAS

AWS. “O que é uma API RESTful?”. Disponível em: <https://aws.amazon.com/pt/what-is/restful-api/>. Acesso em: 26/04/2024.

BOTTENTUIT JUNIOR, João Batista; FURTADO, Cássia; PERCEGUEIRO, Cláudia Maria de Abreu (Orgs.). **Leitura e escrita no mundo digital: desafios e oportunidades para alunos e professores** [https://www.edufma.ufma.br/wp-content/uploads/woocommerce_uploads/2021/03/Livro-Final.pdf]. São Luís: EDUFMA, 2021.

EXPRESS. **Documentação do Express**. Disponível na URL: <https://expressjs.com>. Acesso em: 10/08/2024.

FABRI, J. A.; L'ERARIO, A.; PAGOTTO, T. **SCRUM SOLO**. Disponível em: <https://engenhariosoftware.files.wordpress.com/2016/04/scrum-solo.pdf>. Acesso em: 28/07/2024.

FOWLER, Martin. **UML Essencial: Um breve guia para a linguagem-padrão de modelagem de objetos**. Traduzido por João Tortellos. 3. ed. Porto Alegre: Bookman, 2005.

FRANÇA, Cícero Tadeu Pereira Lima, & CELESTINO JÚNIOR, Joaquim. **Banco de Dados, 2. ed.** Fortaleza: EdUECE, 2015.

FREIRE, Paulo. **A Importância do Ato de Ler**. 1º ed. São Paulo: Cortez, 1989.

GEMINI AI. Disponível na URL: <https://gemini.google.com/app?hl=pt-BR>. Acesso em: 28/01/2025.

GOOGLE AI STUDIO. Disponível na URL: <https://aistudio.google.com>. Acessado em: 29/01/2025.

GOOGLE CLOUD. Disponível na URL: <https://console.cloud.google.com/apis/credentials>. Acesso em: 29/01/2025.

GOOGLE LLC. “**Google Books APIs**”. Disponível em: <https://developers.google.com/books>. Acesso em: 28/04/2024.

GUEDES, G. T. A. **UML2: Uma abordagem prática, 2a ed.** São Paulo: Novatec, 2011.

HOSTINGER. “**O que é uma API RESTful e Porque Isso Importa?**”. Disponível em: <https://www.hostinger.com.br/tutoriais/api-restful>. Acesso em: 26/04/2024.

IBM. *Large Language Models.* IBM Think. Disponível em: <https://www.ibm.com/br-pt/think/topics/large-language-models>. Acesso em: 14/02/ 2025.

IBM. *Arquitetura orientada a eventos.* Disponível em: <https://www.ibm.com/br-pt/topics/event-driven-architecture>. Acesso em: 28/01/2025.

INSTITUTO PRÓ-LIVRO, I. **Retratos da Leitura no Brasil.** Organizada por FAILLA, Zoara. Rio de Janeiro: [s.n.], 2021. Disponível em: https://www.prolivro.org.br/wp-content/uploads/2021/06/Retratos_da_leitura_5_o_livro_IPL.pdf.

JIRA. **Guia do produto.** Disponível na URL: <https://www.atlassian.com/br/software/jira/guides/getting-started/introduction>. Acesso em: 28/07/2024.

MARTIN, Robert C. **Código limpo: habilidades práticas do Agile Software.** Tradução de Rodrigo Almeida. 8. ed. Rio de Janeiro: Alta Books, 2023.

MELO, Jessika Nayara do Amaral **A IMPORTÂNCIA DA LEITURA PRATICADA: UMA ATITUDE REFLEXIVA PARA FORMAÇÃO DO LEITOR.** Revista Científica Semana Acadêmica. Fortaleza, ano MMXIV, Nº. 000066, 26/12/2014.

MONTEIRO, Jean Carlos da Silva. **Geração conectada e os novos métodos de incentivos à leitura na Internet.** In: **Leitura e escrita no mundo digital: desafios e oportunidades para alunos e professores.** São Luís: EDUFMA, 2021. Cap. 8, p. 123-136. Disponível em: https://www.edufma.ufma.br/wp-content/uploads/woocommerce_uploads/2021/03/Livro-Final.pdf.

PRESSMAN, Roger S. **Engenharia de Software: Uma abordagem Profissional.** Porto Alegre: Bookman, 2011.

PRISMA ORM. **Documentação do Prisma.** Disponível na URL: <https://www.prisma.io>. Acesso em 23/04/2024.

PICANÇO, Roberto. **Arquitetura orientada a eventos: soluções escaláveis e em tempo real com EDA.** 1. ed. São Paulo: AOVS Sistemas de Informática, 2023.

POSTGRESQL. **Documentação do PostgreSQL.** Disponível na URL: <https://www.postgresql.org>. Acesso em 23/04/2024.

REACT. **Documentação do React.** Disponível na URL: <https://react.dev>. Acesso em 23/04/2024.

REDIS. **Documentação do Redis**. Disponível na URL: <https://redis.io/docs/latest/>. Acesso em: 28/04/2024.


SCHWABER, K.; SUTHERLAND, J. **Guia Scrum**. 2020. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-3.0.pdf>. Acesso em: 04 abril. 2024.

SOMMERVILLE, I. **Engenharia de Software**. Traduzido por Ivan Bosnic e Kalinka G. de O. Goncalves. 9. ed. São Paulo: Kechi Hirama, 2011.

TYPESCRIPT. **Documentação do TypeScript**. Disponível na URL: <https://www.typescriptlang.org>. Acesso em 23/04/2024.

YUSOF, D. A. (2021). **Reading habits among students in the digital era: Changes of trends and behaviours**. Tun Abdul Razak Library, Universiti Teknologi MARA, Sarawak Branch. <https://doi.org/10.24191/aclim.v1i1.5>.

ZOWGHI, D. & COULIN, C. **Requirements elicitation: A survey of techniques, approaches, and tools**. In: **Engineering and managing software requirements**. [s.l.] Springer, 2005.

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

Documento Digitalizado Ostensivo (Público)

Documento de TCC

Assunto:	Documento de TCC
Assinado por:	Douglas Araujo
Tipo do Documento:	Projeto
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Douglas da Silva Araujo, DISCENTE (202212010010) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 25/03/2025 14:00:05.

Este documento foi armazenado no SUAP em 25/03/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1433611

Código de Autenticação: d0bd16edab

