

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS**

**PROPOSTA DE ARQUITETURA PARA O LENDOCLUB, UMA
PLATAFORMA PARA GERENCIAMENTO DE CLUBES DE
LEITURA**

DANRLEI DE LIRA SILVA

Cajazeiras - PB, 2026

DANRLEI DE LIRA SILVA

**PROPOSTA DE ARQUITETURA PARA O LENDOCLUB, UMA PLATAFORMA
PARA GERENCIAMENTO DE CLUBES DE LEITURA**

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 parcial à obtenção do título de **Tecnólogo em Análise e Desenvolvimento de Sistemas**.

Orientador:

Prof. Me. Diogo Dantas Moreira.

Cajazeiras - PB, 2026



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

DANRLEI DE LIRA SILVA

**PROPOSTA DE ARQUITETURA PARA O LENDOCUB, UMA
PLATAFORMA PARA GERENCIAMENTO DE CLUBES DE LEITURA**

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

Orientador

Prof. Me. Diogo Dantas Moreira

Aprovada em: **13 de fevereiro de 2026.**

Prof. Me. Diogo Dantas Moreira - Orientador

Prof. Dr. Francisco Daladier Marques Junior - Avaliador

IFPB - Campus Cajazeiras

Prof. Me. Afonso Serafim Jacinto - Avaliador

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Fabio Abrantes Diniz**, COORDENADOR(A) DE CURSOS - FUC1 - UNINFO-CZ, em 13/02/2026 12:28:39.
- **Francisco Daladier Marques Junior**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 13/02/2026 13:56:29.
- **Diogo Dantas Moreira**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 13/02/2026 14:20:56.
- **Afonso Serafim Jacinto**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 13/02/2026 15:10:46.

Este documento foi emitido pelo SUAP em 13/02/2026. 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 835855
Verificador: 07075a7897
Código de Autenticação:



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

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

- S586p Silva, Danrlei de Lira.
Proposta de arquitetura para o LendoClub, uma plataforma para gerenciamento de clubes de leitura / Danrlei de Lira Silva.– 2026.
- 56f. : il.
- Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2026.
- Orientador(a): Prof. Me. Diogo Dantas Moreira.
1. Desenvolvimento de sistemas. 2. Arquitetura de software. 3. Clube de leitura - Gestão. 4. Sistema Web. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

IFPB/CZ

CDU: 004.4(043.2)

Dedico este trabalho ao meu avô Zé do Peixe, que sempre me motivou a nunca deixar de estudar, tenho certeza que o senhor estaria orgulhoso.

AGRADECIMENTOS

Em primeiro lugar, gostaria de agradecer à minha família, Layane, Geovana, Clarinha, Totó, Mainha, Painho, Luan, Ruth e Vovó. Obrigado por estarem sempre ao meu lado, mas, principalmente, por serem meu suporte nas horas mais difíceis. Sem vocês, este processo teria sido solitário e triste. Amo vocês com todo o meu coração. Ao meu companheiro, Deivd, agradeço profundamente por sempre me apoiar e por me ouvir, até nas reclamações mais exageradas, compartilhando comigo cada momento, você é incrível, e eu te amo imensamente.

À minha colmeia, meus melhores amigos que encontrei durante o curso, deixo aqui meu agradecimento especial. Vocês tornaram essa jornada muito mais leve e divertida. Sem o apoio de cada um de vocês, não sei se teria chegado até aqui. Amo cada um de vocês de coração. Agradeço também aos meus colegas de graduação, pelos momentos de aprendizado, pelas trocas de experiências e por cada conversa enriquecedora ao longo desses anos. Vocês tornaram essa trajetória acadêmica ainda mais significativa. Aos meus professores, expresso minha profunda gratidão. Cada aula foi uma oportunidade única de aprendizado, e a dedicação de todos foi inspiradora. Muito obrigado pelo conhecimento transmitido e pela paciência.

Em especial, deixo meu mais sincero agradecimento ao meu orientador, Diogo. Desde os primeiros semestres, você se mostrou não apenas um professor exemplar, mas também um grande amigo. Sua orientação foi fundamental para a conclusão deste trabalho e me inspirou a encarar cada etapa com entusiasmo e perseverança. Sua paciência e apoio foram essenciais para o desenvolvimento deste projeto, e sou imensamente grato por acreditar em mim e por me guiar nessa jornada acadêmica. Sua dedicação ao ensino é admirável e continuará a inspirar todos aqueles que têm o privilégio de aprender com você.

Por fim, quero agradecer aos meus primos, Joyce e Zé Neto, por estarem comigo nessa jornada desde o início, antes mesmo de sabermos que todos nós conseguiríamos passar juntos. Obrigado por sempre estarem ao meu lado, participando com empenho em cada projeto e trabalho em equipe. Sem o esforço e paciência de vocês, eu não teria chegado até aqui.

*“Tudo o que temos de decidir é o que fazer com
o tempo que nos é dado.”*

J.R.R. Tolkien, A Sociedade do Anel

RESUMO

Em um cenário cada vez mais marcado pela presença das tecnologias digitais, a leitura permanece como um elemento essencial para o desenvolvimento intelectual, cultural e social dos indivíduos. Embora as transformações tecnológicas modifiquem a forma como as pessoas se relacionam com os livros, elas também criam novas possibilidades para o fortalecimento de comunidades leitoras em ambientes digitais. Nesse contexto, este trabalho apresenta a proposta arquitetural do sistema LendoClub, voltado para o gerenciamento de clubes de leitura.

O sistema tem como objetivo oferecer uma plataforma digital capaz de apoiar a criação, organização e interação entre clubes de leitura, promovendo a troca de experiências entre leitores. Para isso, é proposta uma arquitetura de software fundamentada no estilo de microsserviços, aliada a princípios de arquitetura orientada a eventos, visando atender a requisitos como escalabilidade, segurança e manutenibilidade.

A solução contempla uma aplicação móvel multiplataforma integrada a serviços independentes responsáveis por diferentes domínios do sistema, como clubes, livros e notificações, além do uso de tecnologias modernas para autenticação, comunicação e persistência de dados. Dessa forma, a arquitetura proposta busca servir como base para uma futura implementação do sistema, contribuindo para a discussão sobre o uso de arquiteturas modernas no desenvolvimento de plataformas digitais de incentivo à leitura.

Palavras-chave: Leitura. Clubes de leitura. Arquitetura de software. Microsserviços. Sistemas distribuídos.

ABSTRACT

In a context increasingly marked by the presence of digital technologies, reading remains an essential element for the intellectual, cultural, and social development of individuals. Although technological transformations have changed the way people relate to books, they also create new possibilities for strengthening reading communities in digital environments. In this context, this work presents the architectural proposal of the LendoClub system, aimed at the management of reading clubs.

The system seeks to provide a digital platform capable of supporting the creation, organization, and interaction among reading clubs, promoting the exchange of experiences between readers. To this end, a software architecture based on the microservices style is proposed, combined with principles of event-driven architecture, in order to meet requirements such as scalability, security, and maintainability.

The solution includes a multiplatform mobile application integrated with independent services responsible for different system domains, such as clubs, books, and notifications, in addition to the use of modern technologies for authentication, communication, and data persistence. Thus, the proposed architecture aims to serve as a foundation for a future implementation of the system, contributing to the discussion on the use of modern architectures in the development of digital platforms that encourage reading.

Keywords: Reading. Reading clubs. Software architecture. Microservices. Distributed systems.

LISTA DE FIGURAS

Figura 1.1 – Retratos da leitura no Brasil - O que gosta de fazer no tempo livre	17
Figura 1.2 – Retratos da leitura no Brasil - O que gosta de fazer no tempo livre - Leitor x Não Leitor	17
Figura 1.3 – Resultados ao pesquisar Sucesso no tiktok na Amazon	18
Figura 1.4 – Expositor BookTok	19
Figura 1.5 – Expositor BookTok	19
Figura 1.6 – Retratos da leitura no Brasil - Principal motivação para interesse em leitura de livros	20
Figura 1.7 – Retratos da leitura no Brasil - Último livro lido mais citados	21
Figura 1.8 – Tendências de buscas para o termo Clube do Livro	22
Figura 1.9 – Redes sociais utilizadas como plataforma para clube do livro	23
Figura 4.1 – Arquitetura do projeto	40

LISTA DE QUADROS

Quadro 4.1 – US07 - Adição de Livros do mês	38
Quadro A.1 – US 01 - Registro de Usuário	53
Quadro A.2 – US 02 - Login de Usuário	53
Quadro A.3 – US 03 - Criação de Clube	53
Quadro A.4 – US 04 - Criação de Salas de Bate-Papo	54
Quadro A.5 – US 05 - Aprovação de Registros	54
Quadro A.6 – US 06 - Edição de Perfil	54
Quadro A.7 – US 07 - Adição de Livros do mês	54
Quadro A.8 – US 08 - Pesquisa de Livros	55
Quadro A.9 – US 09 - Criação de Eventos	55
Quadro A.10 – US 10 - Lembretes de Eventos	55
Quadro A.11 – US 11 - Avaliações e Comentários	55
Quadro A.12 – US 12 - Relatórios de Atividade	56
Quadro A.13 – US 13 - Gerenciamento de Cargos	56
Quadro A.14 – US 14 - Participação em Múltiplos Clubes	56
Quadro A.15 – US 15 - Participação em Salas de Bate-Papo	56
Quadro A.16 – US 16 - Enquetes e Votações	57
Quadro A.17 – US 17 - Progresso de Leitura	57
Quadro A.18 – US 18 - Histórico de mensagens nos chats	57

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CRUD	<i>Create, Read, Update and Delete</i>
EDA	<i>Event-Driven Architecture</i>
ElasticSearch	Mecanismo de busca e indexação distribuído
HTTP	<i>Hypertext Transfer Protocol</i>
ISBN	<i>International Standard Book Number</i>
JSON	<i>JavaScript Object Notation</i>
JWT	<i>JSON Web Token</i>
Kafka	Plataforma de mensageria distribuída
Keycloak	Plataforma de gerenciamento de identidade e acesso
MinIO	Serviço de armazenamento de objetos compatível com S3
MongoDB	Sistema de gerenciamento de banco de dados orientado a documentos
NGINX	Servidor web e proxy reverso
NoSQL	<i>Not Only SQL</i>
OAuth 2.0	<i>Open Authorization 2.0</i>
PKCE	<i>Proof Key for Code Exchange</i>
PostgreSQL (PgSQL)	Sistema de gerenciamento de banco de dados relacional
REST	<i>Representational State Transfer</i>
S3	<i>Simple Storage Service</i>
SOC	<i>Service-Oriented Computing</i>
SOA	<i>Service-Oriented Architecture</i>
SPAs	<i>Single Page Applications</i>
SSE	<i>Server-Sent Events</i>
SQLite	Sistema de gerenciamento de banco de dados relacional leve
TCC	Trabalho de Conclusão de Curso
USP	Universidade de São Paulo
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Problemática/Justificativa	21
1.2	Objetivos	23
1.2.1	Objetivos Específicos	24
2	REFERENCIAL TEÓRICO	25
2.1	Engenharia de Software	25
2.2	Arquitetura de Software	26
2.2.1	Arquitetura Monolítica	26
2.2.2	Arquitetura Orientada a Serviços	27
2.2.3	Arquitetura de Microserviços	27
2.2.4	Arquitetura Orientada a Eventos	30
2.3	OAuth 2.0	31
2.3.1	Princípios do OAuth 2.0	31
2.3.2	Funções do OAuth2.0	32
2.3.3	Escopos do OAuth 2.0	32
2.3.4	Tokens de acesso e código de autorização do OAuth 2.0	32
2.3.5	Fluxo geral de funcionamento do OAuth 2.0	33
2.3.6	Tipos de concessão no OAuth 2.0	33
2.4	Object Storage e o Modelo Amazon S3	34
3	TRABALHOS RELACIONADOS	36
3.1	Maratona.app	36
3.2	Bookclubs	36
3.3	Goodreads	36
3.4	Skoob	37

4	PROPOSTA	38
4.1	Processo de levantamento de requisitos	38
4.2	Aplicabilidade e Stakeholders	39
4.3	Arquitetura Proposta	39
4.3.1	Camada de Apresentação	41
4.3.2	API Gateway	41
4.3.3	Serviço de Autenticação e Autorização	41
4.3.4	Comunicação Orientada a Eventos (EDA)	41
4.3.5	Microserviços de Domínio	43
4.3.6	Integração com Serviços Externos	46
4.3.7	Persistência de Dados	46
5	CONSIDERAÇÕES FINAIS	48
	REFERÊNCIAS	50
	APÊNDICE A – USER STORIES	53

1 INTRODUÇÃO

A leitura é uma das atividades mais enriquecedoras para o desenvolvimento intelectual e crítico do ser humano, pois permite que os indivíduos reflitam sobre suas próprias condições de vida e considerem novas perspectivas. Além disso, a literatura tem o poder de nos conectar com o mundo e com as pessoas ao nosso redor.

Entretanto, a prática da leitura, tradicionalmente vista como uma atividade solitária, pode se tornar pouco atraente para muitos, especialmente em tempos de redes sociais. [Schmitz-Boccia \(2012\)](#) observa que os clubes de leitura surgem como uma solução eficaz para esse desafio, funcionando como grupos sociais que promovem a interação entre os leitores. Ao compartilhar suas impressões e opiniões, os participantes ampliam suas visões de mundo e são incentivados a explorar obras que, de outra forma, poderiam não fazer parte de suas escolhas individuais.

Com o advento das redes sociais, a leitura passou a ser influenciada por novos canais de comunicação, sendo o **TikTok**¹ um exemplo notável. Dentro desta plataforma, nichos como o **BookTok**² têm ganhado destaque, reunindo comunidades de leitores e promovendo uma rica troca de recomendações literárias. O BookTok é um espaço onde a literatura é amplamente discutida, e sua popularidade crescente tem incentivado muitas pessoas a retomarem ou iniciarem o hábito de leitura.

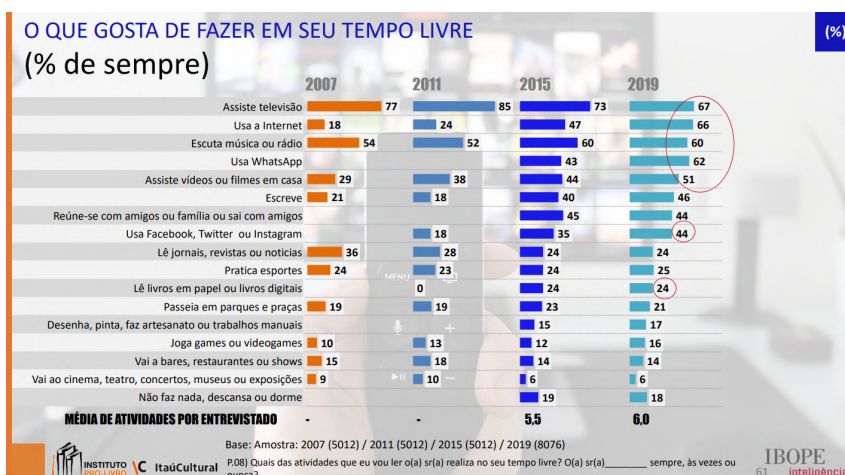
Como pontua [Cazarre \(2023\)](#), “A cultura de nicho permite, então, o desenvolvimento de grupos que, através de laços fortes de interesse, acabam por constituir comunidades e microculturas de afinidade.” Essa nova forma de interação literária mostra que, mesmo em uma era digital, onde a dispersão de atenção é constante, ainda há espaço para a prática da leitura. Redes como o TikTok funcionam como ferramentas poderosas de incentivo à leitura, demonstrando que o hábito de ler não precisa ser solitário, mas pode ser vivenciado em uma comunidade dinâmica e engajada.

Considerando a pesquisa "Retratos da Leitura no Brasil: 5ª Edição" realizada em 2019 pelo Instituto Pró-Livro, observa-se que 82% dos leitores gostariam de ler mais, sendo a falta de tempo (47%) o principal motivo para não conseguirem. Entre os que não leem, as causas mais comuns são a falta de tempo (34%) e o desinteresse pela leitura (28%). A pesquisa também revela que os leitores tendem a ocupar o tempo livre com outras atividades, como assistir televisão, usar a internet, e redes sociais, ou consumir vídeos e música.

¹ Disponível em <<https://tiktok.com>>

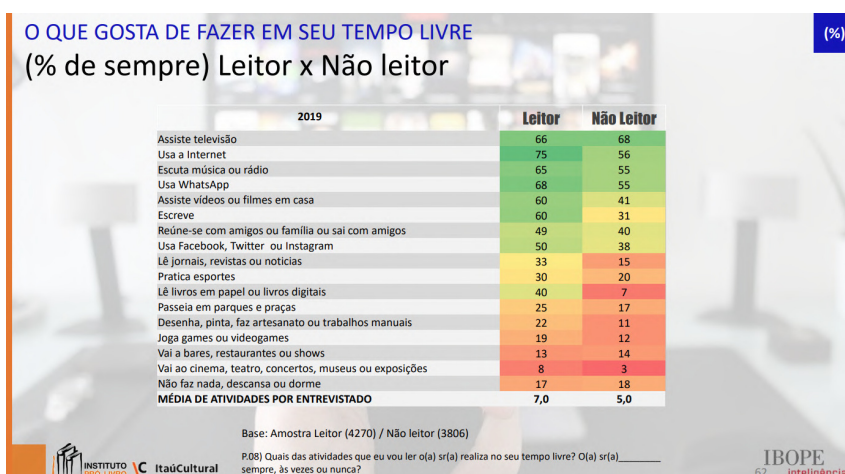
² Disponível em <<https://tiktok.com/tag/BookTok>>

Figura 1.1 – Retratos da leitura no Brasil - O que gosta de fazer no tempo livre



Fonte: (Instituto Pró-Livro, 2019)

Figura 1.2 – Retratos da leitura no Brasil - O que gosta de fazer no tempo livre - Leitor x Não Leitor



Fonte: (Instituto Pró-Livro, 2019)

A Figura 1.2 destaca a diferença na variedade de atividades culturais realizadas no tempo livre entre leitores e não-leitores. Os leitores se envolvem em uma maior diversidade de atividades (em média 7, comparados a 5 dos não-leitores), o que reforça como a leitura também serve como uma porta de entrada para outras formas de cultura.

Nesse contexto, o uso de redes sociais como o TikTok pode se tornar uma extensão benéfica das atividades de lazer, especialmente quando o conteúdo consumido é voltado para a promoção da leitura. Em vez de serem apenas distrações, as plataformas digitais podem inspirar leitores ao expor resenhas, recomendações literárias e discussões de obras populares.

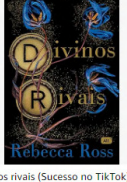
O BookTok, por exemplo, tem transformado a relação dos usuários com a literatura,

oferecendo um ambiente onde a leitura é celebrada e constantemente incentivada. Conteúdos engajadores e influenciadores literários compartilham suas experiências, despertando curiosidade e promovendo o interesse por novas leituras. Assim, a internet, muitas vezes vista como um obstáculo à leitura, revela seu potencial como uma aliada poderosa no desenvolvimento de hábitos literários, especialmente entre os jovens.

Na reportagem do Jornal da USP, publicada em 19 de dezembro de 2023, foi mencionado que a hashtag BookTok acumulava mais de 200 bilhões de visualizações (USP, 2023). Apenas algumas semanas depois, em 16 de janeiro de 2024, o G1 relatou que esse número já havia crescido para cerca de 215 bilhões de visualizações (G1, 2024). Esse aumento significativo, de 15 bilhões de visualizações em menos de um mês, demonstra a força e a expansão rápida dessa comunidade, destacando como a plataforma continua a transformar leitores ocasionais em ávidos consumidores de literatura, além de impulsionar vendas e a visibilidade de muitos autores.

Figura 1.3 – Resultados ao pesquisar Sucesso no tiktok na Amazon

Resultados

 AS COISAS QUE NUNCA SUPERAMOS Lucy Score Patrocinado As Coisas que Nunca Superamos (sucesso no TikTok) (Volume 1) por Lucy Score e Leticia Carvalho ★★★★☆ 807 Capa comum R\$ 59,90 De: R\$ 99,90 Receba até Amanhã, 19 de set. Frete GRÁTIS	 O legado Elle Kennedy O legado – Sucesso no TikTok: 5 Edição Português por Elle Kennedy, Juliana Romeiro, e outros. ★★★★☆ 2.514 Capa Comum R\$ 22,99 De: R\$ 69,90 Receba até sábado, 21 de set. Frete GRÁTIS Mais opções de compra R\$ 19,90 (34 ofertas de produtos novos e usados) Outros formatos: Kindle, Audiolivro	 ASSISTENTE DO VILÃO Hannah Nicole Maehrer Mais vendido Assistente do Vilão (Sucesso no TikTok) por Hannah Nicole Maehrer e Isabela Sampaio ★★★★☆ 1.799 Capa Comum R\$ 41,17 De: R\$ 64,90 Receba até Amanhã, 19 de set. Frete GRÁTIS Mais opções de compra R\$ 39,40 (49 ofertas de novos) Outro formato: Kindle	 MENTIROSOS E. Lockhart Mentirosos – Sucesso no TikTok Edição Português por E. Lockhart e Flávia Souto Maior ★★★★☆ 20.413 Capa Comum R\$ 36,84 De: R\$ 64,90 Receba até Amanhã, 19 de set. Frete GRÁTIS Mais opções de compra R\$ 9,90 (209 ofertas de produtos novos e usados) Idade: 15 anos e acima Outros formatos: Kindle, Audiolivro	 Divinos rivais Rebecca Ross Divinos rivais (Sucesso no TikTok): 1- Idioma : Português por Rebecca Ross e Sofia Soter ★★★★☆ 2.228 Capa Comum R\$ 45,00 De: R\$ 64,90 Receba até Amanhã, 19 de set. Frete GRÁTIS Mais opções de compra R\$ 35,00 (50 ofertas de produtos novos e usados) Outro formato: Kindle
 A ponte entre reinos Danielle L. Jensen A ponte entre reinos – Sucesso no TikTok: 1 Edição Português por Danielle L. Jensen e Guilherme Miranda ★★★★☆ 1.853 Capa Comum R\$ 52,20 De: R\$ 99,90	 ENFEITADOS F.T. Lukens Enfeitados – Sucesso no TikTok por F.T. Lukens ★★★★☆ 113 Capa Comum R\$ 26,90 De: R\$ 64,90 Receba até Amanhã, 19 de set. Frete GRÁTIS	 DIÁRIO DA SOMBRA Kella Shaheen Diário da sombra – Sucesso no TikTok: Um guia para explorar, transcender e integrar seu inconsciente por Kella Shaheen e Ligia Azevedo ★★★★☆ 50 Capa Comum R\$ 27,40 De: R\$ 49,90	 Uma tempestade de verão K.L. Walther Uma tempestade de verão: (Sucesso no TikTok) por K.L. Walther e Sofia Soter ★★★★☆ 2.101 Capa Comum R\$ 40,24 De: R\$ 64,90 Receba até sábado, 21 de set. Frete GRÁTIS	 Quebrando o gelo Hannah Grace Quebrando o gelo: (Sucesso no TikTok): 1 por Hannah Grace e Bruna Miranda ★★★★☆ 3.159 Capa Comum R\$ 54,99 De: R\$ 69,90 Promoção disponível de 20% off Receba até Amanhã, 19 de set.

Fonte: (Amazon Brasil, 2024)

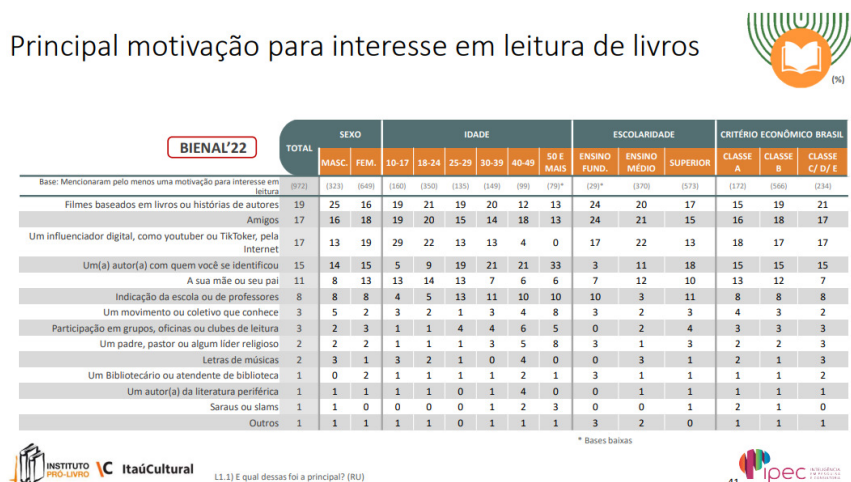
A Figura 1.3 mostra que empresas como a Amazon já tem adotado meios de chamar a atenção dos leitores já adeptos do Booktok, ao adicionar um pequeno texto no título de alguns dos seus produtos, como **Sucesso no TikTok**. Assim como diversas livrarias de todo o mundo também tem reservado espaços em suas lojas apenas com livros famosos do Booktok como visto nas Figuras 1.4 e 1.5.

Esse fenômeno de visibilidade impulsionado pelo BookTok tem gerado impactos significativos nas decisões de compra dos leitores. Como observado por [Cazarré \(2023\)](#), muitos leitores, especialmente da Geração Z, mudaram seus hábitos de compra de livros, preferindo fazer essas aquisições por meios digitais, impulsionados pelas recomendações que encontram no TikTok. Essas mudanças refletem o quanto plataformas como a Amazon têm se adaptado ao comportamento de compra dos leitores.

Além disso, a influência do BookTok vai além de apenas promover livros; ele transforma a experiência de compra em algo mais envolvente e dinâmico, muitas vezes levando os leitores a consumir obras que talvez não escolheriam por si mesmos. Isso reflete nos espaços físicos das livrarias atualmente, a curadoria agora incorpora o que está em alta nas redes sociais, alinhando suas ofertas com as tendências emergentes e reforçando a conexão entre o consumo online e o físico.

A pesquisa "Retratos da Leitura em eventos do livro e literatura", realizada na Bienal do Livro de São Paulo (2022), revelou que 17% dos entrevistados foram motivados a ler por influenciadores digitais, e 28% citaram redes sociais como a fonte da última indicação literária, Figuras 1.6 e 1.7.

Figura 1.6 – Retratos da leitura no Brasil - Principal motivação para interesse em leitura de livros



Fonte: (Instituto Pró-Livro, 2022)

Figura 1.7 – Retratos da leitura no Brasil - Último livro lido mais citados

Último livro lido – Mais citados



	TOTAL	SEXO		IDADE						ESCOLARIDADE		
		MASC.	FEM.	10-17	18-24	25-29	30-39	40-49	50 e MAIS	ENSINO FUND.	ENSINO MÉDIO	SUPERIOR
Base: Leu algum livro nos últimos 3 meses, ou está lendo algum livro	(979)	(327)	(652)	(160)	(352)	(135)	(151)	(101)	(80)	(29)*	(372)	(578)
Viu no YouTube, Instagram, Facebook ou TikTok	28	19	33	40	38	31	17	12	4	34	36	23
Amigo(a)	22	21	23	21	20	20	25	31	25	31	23	21
Algum professor ou professora	9	10	8	8	6	7	11	18	11	7	6	11
Viu matérias ou textos na televisão, jornais ou revistas	5	6	4	3	3	4	11	5	5	0	2	6
Marido, esposa ou companheiro(a)	3	6	2	1	3	3	6	2	4	3	3	3
Mãe ou responsável do sexo feminino	2	2	1	4	2	1	0	1	0	10	3	1
Filho(a), enteado(s) ou tutelados(as)	2	1	2	0	0	0	1	4	13	3	1	2
Pai ou responsável do sexo masculino	1	2	1	1	1	1	1	1	0	0	1	1
Outros	6	6	6	4	4	4	5	14	13	3	4	7
Não recebeu indicação/ Ninguém em especial	22	25	21	18	22	29	26	13	26	3	21	24
Não sabe/ Não respondeu (Esp.)	0	1	0	1	1	1	0	0	0	3	0	0

* Base baixa

* Base baixa



P28B_B) Quem indicou esse último livro que o(a) sr(a) leu ou o que está lendo? (LER OPÇÕES - RJ)



Fonte: (Instituto Pró-Livro, 2022)

A presença dos influenciadores digitais no cenário literário destaca o alcance dessas plataformas na democratização do acesso à leitura. Com conteúdos dinâmicos e acessíveis, esses influenciadores tornam o processo de descoberta de novos livros mais envolvente, especialmente para o público jovem. Além de apresentar novos títulos, esses influenciadores transformam a leitura em um fenômeno social, conectando leitores e promovendo discussões literárias. Dessa forma, as plataformas digitais, muitas vezes vistas como concorrentes da leitura, tornam-se ambientes férteis para o surgimento de uma nova geração de leitores.

1.1 PROBLEMÁTICA/JUSTIFICATIVA

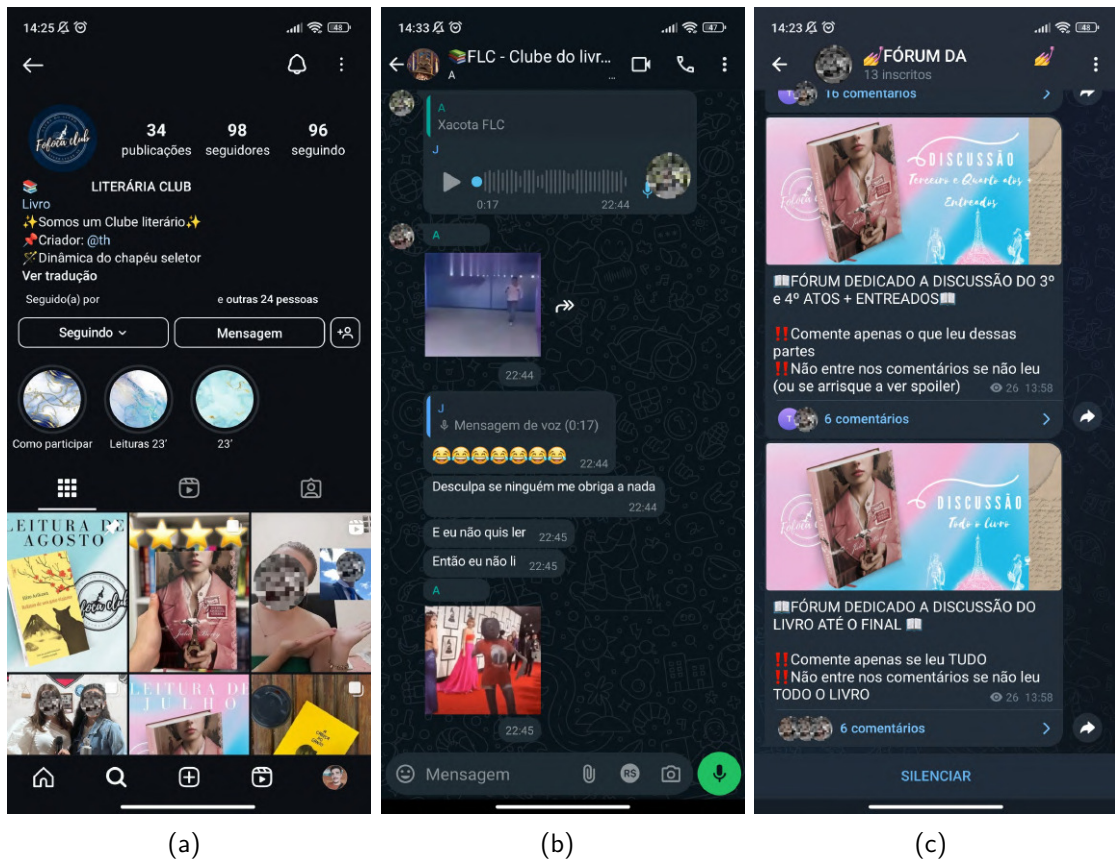
A constante popularidade dos clubes de leitura, como mostrado na Figura 1.8, que apresenta a tendência do termo “Clube do Livro” nos últimos cinco anos, evidencia a necessidade de uma ferramenta que impulse e facilite a organização e a interação entre seus membros.

Figura 1.8 – Tendências de buscas para o termo Clube do Livro

Fonte: ([Google](#), 2024)

Apesar dos benefícios evidentes dos clubes de leitura, ainda há uma lacuna significativa na oferta de ferramentas tecnológicas que facilitam a gestão e a interação contínua entre os membros desses clubes. Atualmente, os clubes de leitura utilizam uma variedade de plataformas e ferramentas, como redes sociais e aplicativos de mensagens, para coordenar suas atividades. Destacam-se os aplicativos Instagram (Figura 1.9 A), WhatsApp (Figura 1.9 B) e Telegram (Figura 1.9 C), conforme ilustrado nas Figuras

Figura 1.9 – Redes sociais utilizadas como plataforma para clube do livro



Fonte: Elaborado pelo autor (2024)

No entanto, essas soluções são fragmentadas e não foram projetadas especificamente para o gerenciamento de clubes de leitura, o que resulta em desafios como:

- **Falta de integração:** A necessidade de utilizar múltiplas plataformas para diferentes atividades, como discussões, organização de eventos, compartilhamento de progresso e proximidade entre os membros.
- **Dificuldade de gerenciamento:** A ausência de ferramentas específicas para o acompanhamento de atividades, metas de leitura e participação dos membros.
- **Interação limitada:** A comunicação entre os membros é muitas vezes dispersa e pouco engajante, dificultando a criação de uma comunidade coesa.

1.2 OBJETIVOS

O objetivo geral do presente trabalho é propor um sistema de gerenciamento de clubes de leitura, que visa integrar as necessidades de organização, comunicação, compartilhamento e

proximidade em uma única plataforma, proporcionando uma experiência mais completa para clubes de leitura e seus membros.

1.2.1 Objetivos Específicos

- Modelar uma plataforma que permita a criação de clubes de leitura, gerenciamento de membros e acompanhamento de participação.
- Descrever ferramentas para a organização e gerenciamento de eventos literários, como reuniões, desafios e discussões temáticas.
- Especificar funcionalidades de comunicação, como salas de bate-papo dedicadas e possibilidades de resenhas e opiniões sobre os livros durante o progresso de leitura.
- Propor ferramentas para acompanhar o progresso de leitura dos membros, estatísticas de participação e engajamento, auxiliando os administradores na gestão do clube.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os principais fundamentos teóricos que sustentam a proposta arquitetural do sistema LendoClub. São abordados conceitos relacionados à engenharia de software e à arquitetura de software, com foco nos estilos arquiteturais utilizados em sistemas distribuídos, como as arquiteturas monolítica, orientada a serviços, de microserviços e orientada a eventos.

Além disso, são discutidas tecnologias e mecanismos que apoiam esse modelo arquitetural, incluindo plataformas de mensageria, padrões de comunicação entre serviços, estratégias de segurança e modelos de armazenamento de dados em ambientes distribuídos. O objetivo deste capítulo é fornecer o embasamento conceitual necessário para justificar as escolhas arquiteturais adotadas na proposta do sistema, estabelecendo uma relação consistente entre os conceitos teóricos apresentados e a solução proposta nos capítulos seguintes.

2.1 ENGENHARIA DE SOFTWARE

A **engenharia de software** é a área responsável por aplicar princípios científicos, métodos e técnicas para o desenvolvimento, operação e manutenção de sistemas de software. Seu objetivo principal é produzir software de alta qualidade, que atenda às necessidades dos usuários, respeitando restrições de custo, prazo e qualidade.

Para o *Institute of Electrical and Eletronic Engineering* a engenharia de software é a aplicação sistemática, disciplinada e com abordagem quantitativa para o desenvolvimento, operação e manutenção de software ([Institute of Electrical and Electronics Engineers, 1990](#)).

Segundo [Sommerville \(1992\)](#) a engenharia de software envolve questões técnicas e não-técnicas, tais como a especificação do conhecimento, técnicas de projeto e implementação, conhecimentos dos fatores humanos pelo engenheiro de software e ainda, gestão de projetos.

No contexto dos sistemas de informação, a engenharia de software desempenha um papel fundamental ao garantir que soluções tecnológicas sejam capazes de apoiar processos organizacionais, facilitar a comunicação entre usuários e promover o acesso eficiente à informação. Sistemas de informação modernos precisam lidar com volumes crescentes de dados, múltiplos usuários simultâneos e integração com serviços externos, o que exige arquiteturas cada vez mais flexíveis e escaláveis.

2.2 ARQUITETURA DE SOFTWARE

A arquitetura de software pode ser definida como a estrutura fundamental de um sistema, composta por seus componentes, suas responsabilidades e as relações estabelecidas entre eles. Ela representa um conjunto de decisões de alto nível que influenciam diretamente atributos de qualidade como desempenho, escalabilidade, manutenibilidade, segurança e disponibilidade. De acordo com [Silveira \(2012\)](#), a arquitetura de software atua como um elo entre as necessidades de negócio e as especificações técnicas. Sua função é converter o entendimento dos casos de uso em estratégias viáveis de implementação dentro do sistema.

Decisões arquiteturais são difíceis de alterar após a consolidação do sistema, tornando essencial que sejam bem fundamentadas desde as etapas iniciais do projeto. Dessa forma, a arquitetura atua como um elo entre os requisitos do sistema e sua implementação técnica, orientando o desenvolvimento e a evolução da solução ao longo do tempo.

"Uma boa implementação, design ou arquitetura é aquela que permite modificações causando somente um impacto considerado justo a outras partes do sistema, e, ao mesmo tempo, diminuindo as ocorrências de tais situações, minimizando o custo de desenvolvimento e manutenção ([SILVEIRA, 2012](#))."

A excelência na arquitetura de software é fundamental para mitigar riscos de negócio durante o desenvolvimento técnico. Um design de qualidade deve ser resiliente o bastante para suportar a obsolescência tecnológica e as mudanças nas demandas dos usuários. Nesse contexto, o arquiteto precisa avaliar o impacto global de suas escolhas, equilibrando atributos de qualidade conflitantes — como desempenho e segurança — para harmonizar as necessidades do sistema com os objetivos comerciais. ([Microsoft, 2009](#))

A arquitetura de software compreende diversos modelos, tais como o monolítico, o orientado a serviços (SOA), o de microsserviços e o orientado a eventos, entre outros. Cada estilo arquitetural possui padrões e particularidades distintas. Nas seções seguintes, serão detalhadas as propriedades fundamentais das arquiteturas mais adotadas pela indústria de software.

2.2.1 Arquitetura Monolítica

A arquitetura monolítica caracteriza-se pela construção de um sistema como uma única unidade, na qual todos os módulos e funcionalidades são fortemente integrados. Nesse modelo, componentes como interface, regras de negócio e acesso a dados coexistem em um mesmo artefato de implantação.

Embora arquiteturas monolíticas possam ser mais simples de desenvolver inicialmente, especialmente em sistemas de pequeno porte, elas tendem a apresentar dificuldades de manutenção e escalabilidade à medida que o sistema cresce. Alterações em um módulo podem impactar outras partes do sistema, tornando a evolução mais complexa e aumentando o risco de falhas. Uma implantação monolítica representa um ponto único de falha: caso a aplicação apresente um erro, todos os serviços integrados ficam indisponíveis simultaneamente ([VILLAMIZAR et al., 2015](#)).

2.2.2 Arquitetura Orientada a Serviços

A Arquitetura Orientada a Serviços (SOA) é um modelo de design que permite que as funcionalidades de uma aplicação sejam disponibilizadas como um conjunto de serviços independentes, geralmente implementados como *Web Services*. Segundo [Erl \(2016\)](#), a essência do SOA reside no uso de interfaces padronizadas que podem ser invocadas, publicadas e descobertas, promovendo um baixo acoplamento. Diferente de modelos baseados em objetos ou componentes internos, a interação no SOA foca em esquemas de mensagens e interfaces de escopo definido, tratando o serviço como uma entidade lógica de negócio e não apenas como um provedor de componentes técnicos.

Este estilo arquitetural é capaz de encapsular processos empresariais em serviços interoperáveis, utilizando diversos protocolos de rede e formatos de dados para o intercâmbio de informações. Dessa forma, clientes e outros sistemas podem consumir serviços que residem localmente ou acessá-los remotamente via rede, garantindo flexibilidade na integração de sistemas heterogêneos ([KUMARI; RATH, 2015](#)).

Embora a SOA tenha contribuído significativamente para a evolução das arquiteturas distribuídas, sua adoção muitas vezes envolveu infraestruturas complexas e alto acoplamento entre serviços, o que dificultou sua aplicação em ambientes mais dinâmicos. Historicamente, aplicações que adotam a arquitetura SOA são voltadas para cenários complexos que exigem o compartilhamento de informações entre múltiplos processos.

2.2.3 Arquitetura de Microserviços

A arquitetura de microserviços é definida como um estilo arquitetural inspirado na computação orientada a serviços (SOC) que foca na decomposição funcional de sistemas em unidades autônomas. De acordo com [Dragoni et al. \(2016\)](#), um microserviço é um processo coeso e independente que interage exclusivamente por meio de troca de mensagens. O termo "coeso" refere-se ao fato de que cada serviço implementa apenas funcionalidades fortemente relacionadas ao domínio ou preocupação que se propõe a modelar.

Nesta arquitetura, o software é fragmentado em pequenos serviços independentes e

com responsabilidades bem definidas, facilitando processos isolados de teste e escala. Conforme apontam [O'Connor et al. \(2017\)](#), a comunicação entre esses módulos ocorre por interfaces de rede, podendo adotar tanto a comunicação síncrona (HTTP) quanto a assíncrona, baseada em tecnologias de mensageria como RabbitMQ ou Apache Kafka.

Este modelo se diferencia do monólito por permitir que cada componente seja implantado e executado de forma independente, possuindo inclusive suas próprias ferramentas de persistência de dados e memória. A arquitetura de microsserviços, portanto, caracteriza uma aplicação distribuída onde todos os seus módulos constituintes possuem essas propriedades de autonomia.

A transição do modelo monolítico para microsserviços visa mitigar problemas críticos de manutenção e escalabilidade. Conforme apontam [Dragoni et al. \(2016\)](#), as principais vantagens desta abordagem incluem:

- **Manutenibilidade:** O tamanho reduzido do código base de cada serviço limita o escopo de erros e facilita o teste de funcionalidades em isolamento.
- **Escalabilidade Granular:** Diferente do monólito, que exige a duplicação de toda a aplicação para escalar, os microsserviços permitem instanciar ou descartar serviços específicos de acordo com a carga de demanda.
- **Independência Tecnológica:** Não há imposição de uma linguagem ou paradigma específico; cada serviço pode ser implementado com a tecnologia mais adequada, desde que respeite os protocolos de comunicação da rede.
- **Implantação Contínua:** O uso de contêineres e pipelines de automação permite que atualizações sejam enviadas para produção em segundos, sem a necessidade de reiniciar o sistema completo.

É necessário considerar que a adoção de microsserviços impõe desafios operacionais e custos adicionais. Segundo [Fowler e Lewis \(2014\)](#), entre esses impactos destacam-se a sobrecarga computacional decorrente da execução de múltiplos processos isolados, a latência inerente à comunicação via rede e o aumento na complexidade de manutenção, especialmente quando são empregadas diferentes linguagens de programação nos microsserviços.

2.2.3.1 API Gateway em Arquiteturas de Microsserviços

Em arquiteturas de microsserviços, a decomposição do sistema em diversos serviços independentes traz benefícios como escalabilidade e flexibilidade evolutiva, mas também introduz novos desafios relacionados à comunicação, segurança e gerenciamento de interfaces. Nesse

contexto, o **API Gateway** surge como um componente central para organizar e mediar o acesso aos microsserviços.

Segundo [Zhao et al. \(2018\)](#), o API Gateway pode ser entendido como um servidor especializado que atua como único ponto de entrada para as requisições destinadas ao conjunto de microsserviços, abstraindo do cliente os detalhes internos da arquitetura distribuída. Dessa forma, o cliente não precisa conhecer a localização, a quantidade ou a granularidade dos serviços existentes, interagindo apenas com o gateway.

Uma das principais motivações para o uso de um API Gateway em arquiteturas de microsserviços está relacionada à diferença de granularidade entre as APIs fornecidas pelos serviços e as necessidades dos clientes. Enquanto os microsserviços tendem a expor APIs mais específicas e de baixo nível, os clientes geralmente demandam operações mais agregadas. O API Gateway atua como uma camada intermediária capaz de adaptar essas interfaces, reduzindo o número de chamadas necessárias e simplificando a lógica do cliente.

Além de atuar como ponto de acesso unificado, o API Gateway concentra funcionalidades transversais que, se distribuídas entre os microsserviços, aumentariam o acoplamento e a complexidade do sistema. Entre essas funcionalidades destacam-se autenticação, autorização, balanceamento de carga, controle de fluxo, monitoramento, cache e registro de logs. [Zhao et al. \(2018\)](#) ressaltam que, ao transferir essas responsabilidades para o gateway, os microsserviços podem se concentrar exclusivamente na lógica de negócio, tornando-se mais simples e coesos.

Outro aspecto relevante do API Gateway é sua contribuição para a resiliência do sistema. Em cenários de alta carga ou falha de serviços, o gateway pode aplicar mecanismos como limitação de requisições, degradação de serviços e circuit breaker, evitando o efeito cascata conhecido como “avalanche de serviços”. Essas estratégias permitem que o sistema continue operando de forma controlada, mesmo diante de falhas parciais.

No que se refere à segurança, o API Gateway desempenha um papel fundamental ao centralizar os processos de autenticação e autorização. De acordo com [Zhao et al. \(2018\)](#), é comum a adoção de protocolos padronizados, como OAuth 2.0 e OAuth 2.0 combinado com JWT, permitindo que o gateway valide as credenciais dos clientes e repasse aos microsserviços apenas requisições já autenticadas. Essa abordagem reduz a duplicação de lógica de segurança e facilita a manutenção das políticas de acesso.

Do ponto de vista arquitetural, o uso de um API Gateway também contribui para a evolução do sistema ao longo do tempo. Como o gateway encapsula a estrutura interna da aplicação, mudanças na organização dos microsserviços — como divisão, fusão ou substituição de serviços — podem ser realizadas sem impactar diretamente os clientes. Essa característica é

especialmente relevante em sistemas que passam por evolução contínua e adaptação frequente aos requisitos do domínio.

Dessa forma, o API Gateway configura-se como um componente essencial em arquiteturas de microsserviços, oferecendo uma solução estruturada para o gerenciamento de APIs, segurança, comunicação e controle de tráfego. Sua adoção, quando fundamentada em boas práticas arquiteturais, contribui significativamente para a organização e a sustentabilidade de sistemas distribuídos modernos.

2.2.4 Arquitetura Orientada a Eventos

A Arquitetura Orientada a Eventos (EDA) é um estilo de arquitetura focado na produção, detecção e consumo de eventos como principal meio de comunicação entre serviços. Diferente dos modelos tradicionais baseados em requisição-resposta (como o HTTP REST), a EDA promove o desacoplamento extremo, permitindo que os sistemas processem informações de forma assíncrona e em tempo real ([ODOFIN et al., 2022](#)).

O modelo baseia-se em três componentes fundamentais:

1. **Produtores de Eventos:** Responsáveis por capturar uma mudança de estado (o evento) e publicá-la no sistema.
2. **Barramento de Mensagens (Broker):** Atua como o intermediário que recebe e distribui os eventos.
3. **Consumidores de Eventos:** Serviços que reagem aos eventos pertinentes às suas responsabilidades.

O trabalho de [Chou \(2008\)](#) descreve que a arquitetura orientada a eventos representa um novo paradigma, tendo como princípios:

- **Padrão de mensagens assíncronas:** Diferente do modelo tradicional de requisição/resposta (síncrono), a EDA baseia-se no padrão de publicação e subscrição (publish/subscribe). Nesse cenário, os produtores enviam notificações aos interessados sem interromper ou bloquear seu próprio fluxo de processamento à espera de um retorno.
- **Mensagens autônomas:** Os eventos são transmitidos como mensagens independentes que carregam toda a informação necessária para a execução da tarefa. Segundo [Chou \(2008\)](#), esses eventos não devem possuir dependências de hardware ou de sessões específicas, focando estritamente na comunicação de mudanças de estado de negócio entre os diferentes domínios da organização.

- **Sistemas distribuídos com alto desacoplamento:** O uso de mensagens assíncronas encapsula a complexidade interna dos sistemas. Esse modelo reduz a necessidade de interfaces rígidas ou definições complexas de APIs, exigindo apenas um formato semântico padronizado (como XML ou JSON) para garantir a interoperabilidade entre os módulos.
- **O Receptor possui o controle do fluxo de mensagens:** Na EDA, o consumidor do evento detém o controle sobre a carga de trabalho que pode processar. Os sistemas de mensageria permitem verificar o status do processamento e oferecem escalabilidade, possibilitando que um único evento seja distribuído e processado por diversos nós simultaneamente, conforme a capacidade do receptor.
- **Propagação em tempo real:** A eficácia desta arquitetura está intrinsecamente ligada à sua capacidade de publicar e consumir informações em tempo real. [Chou \(2008\)](#) ressalta que, sem a propagação instantânea, o modelo perde sua vantagem competitiva, assemelhando-se à eficiência limitada das arquiteturas tradicionais baseadas em processamento por lotes.

2.3 OAUTH 2.0

O **OAuth 2.0** é um padrão amplamente adotado para autorização delegada, projetado para permitir que aplicações obtenham acesso controlado a recursos protegidos em nome de um usuário, sem a necessidade de compartilhamento direto de credenciais. Introduzido como sucessor do OAuth 1.0, o OAuth 2.0 consolidou-se como o padrão de fato para autorização em aplicações web, móveis e sistemas distribuídos modernos.

Diferentemente de protocolos de autenticação, o OAuth 2.0 tem como foco principal a autorização, isto é, a concessão de permissões específicas para que um cliente execute determinadas ações sobre recursos protegidos. Essa distinção é fundamental, uma vez que o protocolo não se responsabiliza por verificar a identidade do usuário final, mas sim por controlar quais recursos podem ser acessados e em quais condições ([Auth0, 2026](#)).

2.3.1 Princípios do OAuth 2.0

O OAuth 2.0 baseia-se no conceito de acesso consentido, no qual o proprietário do recurso autoriza explicitamente que uma aplicação cliente acesse determinados recursos em seu nome. Essa abordagem reduz riscos de segurança ao eliminar a necessidade de que o cliente armazene ou manipule diretamente credenciais sensíveis, como senhas.

O protocolo opera por meio do uso de tokens de acesso, que representam a autorização concedida ao cliente. Esses tokens possuem escopo limitado, validade temporal definida e podem ser revogados, oferecendo maior controle sobre o acesso aos recursos. Embora o OAuth

2.0 não imponha um formato específico para os tokens, o uso de JSON Web Tokens (JWT) é amplamente difundido por permitir a inclusão de informações estruturadas e verificáveis no próprio token (Auth0, 2026).

2.3.2 Funções do OAuth2.0

A especificação do OAuth 2.0 define um conjunto de papéis que compõem o ecossistema de autorização:

- **Proprietário do recurso:** Entidade que possui os recursos protegidos e concede autorização para seu acesso, geralmente representada pelo usuário final;
- **Cliente:** Aplicação que solicita acesso aos recursos protegidos em nome do proprietário do recurso;
- **Servidor de autorização:** Responsável por autenticar o proprietário do recurso, obter seu consentimento e emitir tokens de acesso para o cliente;
- **Servidor de recursos:** componente que hospeda os recursos protegidos e valida os tokens apresentados pelo cliente antes de conceder acesso.

Essa separação de responsabilidades contribui para uma arquitetura mais segura e modular, especialmente em sistemas distribuídos e baseados em microsserviços.

2.3.3 Escopos do OAuth 2.0

Os escopos desempenham um papel central no OAuth 2.0, pois permitem definir com precisão quais ações um cliente está autorizado a realizar. Cada escopo representa um conjunto específico de permissões associadas a determinados recursos ou operações.

2.3.4 Tokens de acesso e código de autorização do OAuth 2.0

O OAuth 2.0 utiliza diferentes tipos de credenciais temporárias para garantir que o acesso aos recursos seja concedido de forma segura e controlada. O processo de emissão de privilégios não deve, idealmente, expor o token de acesso diretamente durante a fase de consentimento do usuário. O código de autorização atua como uma credencial temporária de uso único. Em vez de emitir o *Access Token* imediatamente após a aprovação do usuário, o Servidor de Autorização retorna um código ao cliente. O cliente, então, realiza uma chamada segura (canal *back-channel*) para trocar esse código pelo token definitivo.

2.3.5 Fluxo geral de funcionamento do OAuth 2.0

Antes de iniciar o processo de autorização, o cliente deve estar previamente registrado no servidor de autorização, obtendo uma identificação própria e, em alguns casos, um segredo de cliente. A partir desse registro, o fluxo geral do OAuth 2.0 pode ser descrito da seguinte forma:

1. o cliente solicita autorização ao servidor de autorização, informando suas credenciais, escopos desejados e um endereço de redirecionamento;
2. o servidor de autorização autentica o cliente e solicita o consentimento do proprietário do recurso;
3. após o consentimento, o servidor retorna ao cliente um código de autorização ou um token de acesso, conforme o tipo de concessão adotado;
4. o cliente utiliza o token de acesso para requisitar recursos ao servidor de recursos;
5. o servidor de recursos valida o token e, se autorizado, fornece o recurso solicitado.

Esse fluxo garante que o acesso aos recursos ocorra de forma controlada, rastreável e segura.

2.3.6 Tipos de concessão no OAuth 2.0

O OAuth 2.0 define diferentes tipos de concessão, cada um adequado a um cenário específico de uso:

- **Concessão de código de autorização:** Considerada uma das formas mais seguras, especialmente para aplicações web e móveis, pois utiliza um código intermediário trocado por um token de acesso;
- **Concessão de código de autorização com PKCE:** Extensão do fluxo anterior, adicionando mecanismos de proteção contra interceptação, sendo recomendada para aplicações móveis e SPAs;
- **Concessão implícita:** Fluxo simplificado no qual o token é retornado diretamente ao cliente, atualmente desaconselhado devido a vulnerabilidades de segurança;
- **Concessão de credenciais do proprietário do recurso:** utilizada apenas em cenários altamente confiáveis, nos quais o cliente tem acesso direto às credenciais do usuário;
- **Concessão de credenciais do cliente:** Indicada para comunicação entre aplicações e microsserviços, sem interação direta com o usuário;

- **Fluxo de autorização do dispositivo:** Voltado para dispositivos com restrições de entrada, como televisores inteligentes;
- **Concessão de token de atualização:** permite a renovação de tokens de acesso expirados sem nova interação do usuário.

A escolha adequada do tipo de concessão é fundamental para garantir segurança e compatibilidade com o contexto da aplicação.

2.4 OBJECT STORAGE E O MODELO AMAZON S3

Em aplicações modernas que lidam com dados como imagens, documentos e outros conteúdos não estruturados, o modelo tradicional de armazenamento baseado em sistemas de arquivos ou bancos de dados relacionais tende a apresentar limitações relacionadas à escalabilidade e à flexibilidade de acesso. Esse cenário é ainda mais evidente em sistemas distribuídos e baseados em microsserviços, nos quais diferentes componentes da aplicação precisam acessar dados de forma independente e desacoplada.

Nesse contexto, o **object storage** surge como um modelo de armazenamento projetado para atender a esse tipo de demanda, permitindo o gerenciamento eficiente de grandes volumes de dados não estruturados. Diferentemente dos modelos tradicionais, o object storage não organiza os dados em estruturas hierárquicas de diretórios ou blocos, mas sim em objetos independentes, cada um identificado por um identificador único e acompanhado de metadados associados. (AWS, 2026)

De acordo com Wang (2022), esse modelo favorece ambientes distribuídos ao possibilitar maior escalabilidade e tolerância a falhas, uma vez que os objetos podem ser armazenados e replicados em diferentes nós sem dependência de uma estrutura centralizada. O autor destaca que a separação entre dados e metadados é um dos principais fatores que contribuem para a eficiência e a flexibilidade desse tipo de armazenamento, especialmente em sistemas de grande porte.

No âmbito da computação em nuvem, o object storage tornou-se amplamente adotado por provedores de serviços, sendo o Amazon Simple Storage Service (**Amazon S3**)¹ um dos exemplos mais consolidados desse modelo. Segundo a Amazon Web Services, o S3 é um serviço de armazenamento de objetos voltado para alta durabilidade, disponibilidade e escalabilidade, permitindo que aplicações armazenem e recuperem dados por meio de interfaces padronizadas baseadas em HTTP.

¹ Disponível em <<https://aws.amazon.com/pt/s3/>>

O modelo adotado pelo Amazon S3 organiza os dados em buckets e objects, nos quais cada objeto é acessado por meio de uma chave única. Além disso, o serviço permite a associação de metadados aos objetos armazenados, o que possibilita maior controle sobre permissões, tipos de conteúdo e políticas de acesso. Conforme discutido por [Wang \(2022\)](#), o uso eficiente de metadados é um elemento central para o desempenho e a organização de sistemas de object storage.

Outro aspecto relevante do Amazon S3 é sua integração natural com arquiteturas distribuídas e de microsserviços. O acesso aos objetos ocorre por meio de APIs REST, o que facilita o consumo do serviço por diferentes aplicações e componentes, independentemente da linguagem ou plataforma utilizada. Essa característica torna o object storage particularmente adequado para o armazenamento de recursos compartilhados, como imagens de perfil, arquivos de apoio e conteúdos multimídia.

Além disso, serviços de object storage como o Amazon S3 são projetados para oferecer alta tolerância a falhas, utilizando mecanismos de replicação automática dos dados em múltiplas zonas de disponibilidade. Essa abordagem reduz a necessidade de gerenciamento manual de backups e contribui para a confiabilidade do sistema como um todo.

3 TRABALHOS RELACIONADOS

Este capítulo aborda plataformas e sistemas existentes que compartilham características com o conceito de clube do livro, destacando suas funcionalidades e limitações. As plataformas analisadas foram identificadas por meio de mecanismos de busca na web, bem como através do conhecimento prévio do autor.

O Skoob¹ e o Goodreads² são exemplos de redes sociais literárias conhecidas pelo autor e já utilizadas para fins pessoais. O Maratona.app³ foi descoberto através de criadores de conteúdo literário no TikTok, enquanto o Bookclubs⁴ foi identificado por meio de pesquisas na web. As palavras-chave utilizadas para a pesquisa incluem "aplicativo de clubes do livro", "book clubs" e "clubes do livro".

3.1 MARATONA.APP

O Maratona.app é uma plataforma dedicada à organização de leituras coletivas, que promove a interação entre leitores e criadores de conteúdo literário. Embora o Maratona.app seja eficaz, leituras coletivas são atividades de curto prazo com foco em um livro específico, não oferecendo um suporte mais extensivo para a gestão contínua e a longo prazo de clubes de leitura.

3.2 BOOKCLUBS

O Bookclubs oferece uma proposta muito parecida com o LendoClub, fornecendo ferramentas para a criação e gestão de clubes de leitura. No entanto, o Bookclubs não é uma solução nacional, sendo utilizado e popular principalmente em países de língua inglesa. Isso pode limitar o acesso e a relevância para os usuários brasileiros, que podem preferir soluções mais adaptadas às suas necessidades.

3.3 GOODREADS

Goodreads é uma rede social literária popular, onde usuários podem avaliar livros, escrever resenhas, criar listas de leitura e participar de grupos temáticos. Com uma base de usuários global, o Goodreads oferece diversas funcionalidades voltadas para interação social em torno da leitura. No entanto, sua abordagem é mais individualista, centrada no leitor como

¹ Disponível em <<https://skoob.com.br>>

² Disponível em <<https://goodreads.com>>

³ Disponível em <<https://maratona.app/>>

⁴ Disponível em <<https://bookclubs.com>>

indivíduo, e menos voltada para clubes de leitura, o que pode ser uma limitação para grupos que buscam uma experiência mais colaborativa.

3.4 SKOOB

O Skoob é a principal rede social literária no Brasil, permitindo que leitores avaliem livros, escrevam resenhas e interajam com outros usuários. Apesar de sua popularidade, assim como o Goodreads, o Skoob também não oferece funcionalidades específicas para gestão de clubes de livros e leituras coletivas.

4 PROPOSTA

O aplicativo proposto busca oferecer uma solução para o gerenciamento de clubes de leitura, criando um ambiente integrado onde os membros podem interagir, monitorar seu progresso de leitura e participar de atividades organizadas. Entre as funcionalidades destacadas, o sistema contará com ferramentas de gestão de clubes e membros, permitindo a criação de clubes de leitura, a administração de seus participantes e o acompanhamento detalhado da participação de cada um.

Além disso, o LendoClub facilitará a organização de atividades literárias, possibilitando a criação e gerenciamento de eventos como reuniões, desafios de leitura e discussões temáticas. A interação e comunicação entre os membros será amplamente incentivada, onde os participantes poderão compartilhar suas opiniões sobre os livros durante seu progresso de leitura.

4.1 PROCESSO DE LEVANTAMENTO DE REQUISITOS

O processo de levantamento de requisitos foi baseado na experiência pessoal do autor, juntamente com conversas informais com os participantes de dois clubes do livro em que o autor participa. Os requisitos do sistema foram documentados no formato de *user stories*, como demonstrado no Quadro 4.1. A escolha objetiva preservar a visão do produto que os participantes dos clubes tinham, além de ser escrita em linguagem simples e objetiva, facilitando a compreensão do contexto e o porquê de cada uma ser priorizada no *backlog*.

Quadro 4.1 – US07 - Adição de Livros do mês

US 07 - Adição de Livros do mês			
Como um administrador, quero adicionar novos livros ao feed do clube para que membros possam vê-los e avaliá-los. O cadastro deve incluir capa do livro, título, autor, gênero, ISBN e sinopse.			
Story Points	3	Prioridade	Alta
TA1.1	Verificar se livros podem ser adicionados.		
TA1.2	Validar campos obrigatórios.		

Fonte: Elaborado pelo autor (2026)

Uma lista completa das *user stories* levantadas para este trabalho com informações relevantes como *story points* e prioridade pode ser encontrada no [Apêndice A - User Stories](#).

4.2 APLICABILIDADE E STAKEHOLDERS

A potencialidade de aplicabilidade do sistema é significativa, considerando o crescimento contínuo das comunidades de leitura e o aumento da demanda por ferramentas especializadas. Os principais *stakeholders* do aplicativo incluem **leitores**, **editoras**, **livrarias** e **instituições de ensino**, todos com interesses específicos que o sistema atenderá de maneira estratégica.

Os **leitores** são o principal grupo de usuários do aplicativo. Eles buscam uma plataforma que facilite a organização de suas leituras, a interação com outros leitores, e a participação em clubes de leitura. A possibilidade de acompanhar o progresso de leitura, participar de discussões, e compartilhar opiniões sobre livros são funcionalidades que atendem diretamente às necessidades desse grupo.

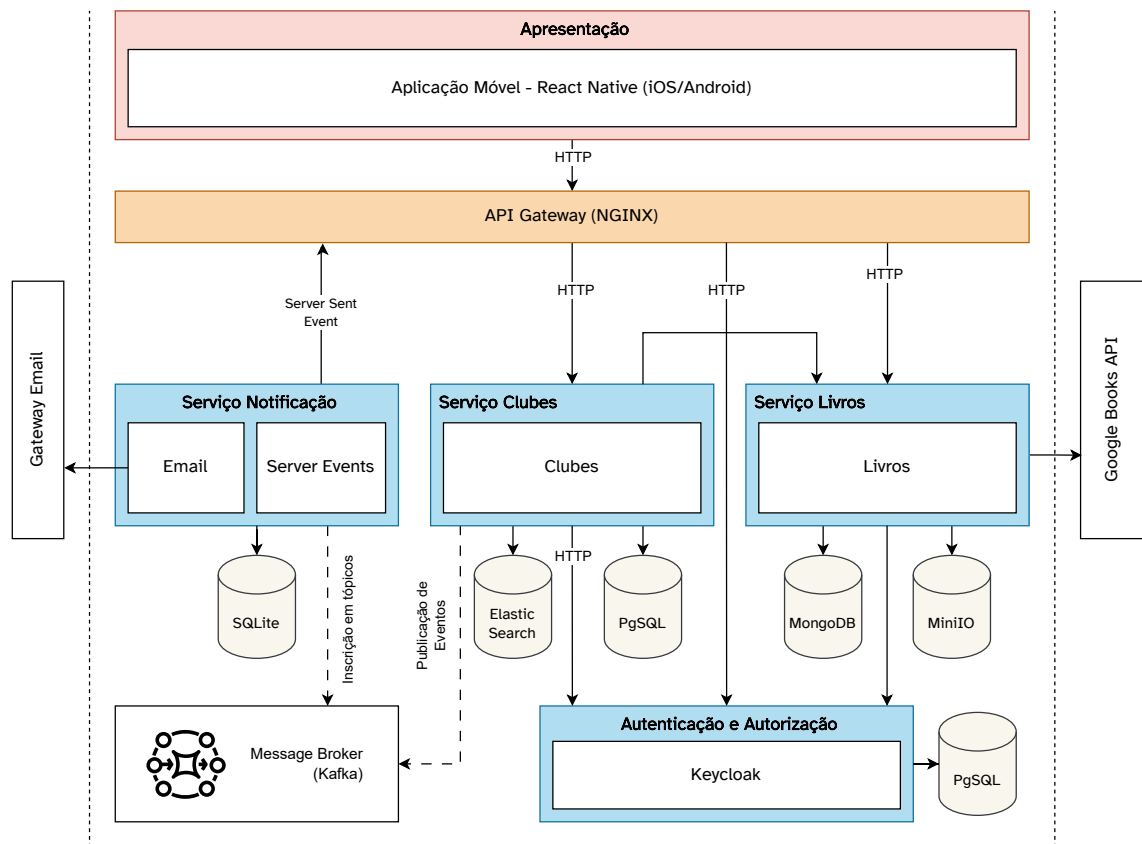
Editoras e livrarias podem utilizar o aplicativo como uma ferramenta estratégica para engajar seus clientes e promover obras literárias. Através de clubes de leitura patrocinados ou parcerias com autores, essas organizações podem criar uma conexão mais forte com o público leitor, além de incentivar a venda de livros. O aplicativo oferecerá uma plataforma onde essas entidades podem organizar eventos literários, desafios de leitura, e discussões que promovam seus produtos.

Escolas, universidades e institutos representam *stakeholders* importantes que podem utilizar o aplicativo para promover a leitura entre seus membros. Essas instituições podem criar clubes de leitura destinados a estudantes, docentes e servidores, facilitando a organização de grupos de leitura e incentivando a discussão sobre obras literárias. O uso do aplicativo permitirá que essas instituições alcancem objetivos educacionais e culturais de forma mais estruturada.

4.3 ARQUITETURA PROPOSTA

A arquitetura proposta para o LendoClub adotará o estilo de microsserviços, no qual o sistema será dividido em serviços independentes, cada um responsável por um contexto específico do domínio. Como demonstrado na figura 4.1 a seguir. Essa abordagem visa promover maior escalabilidade, manutenibilidade e flexibilidade evolutiva em comparação a arquiteturas monolíticas tradicionais.

Figura 4.1 – Arquitetura do projeto



Fonte: Elaborado pelo autor (2026)

O acesso ao sistema será realizado por meio de uma aplicação móvel desenvolvida em React Native, que se comunicará exclusivamente com um API Gateway. Este componente atuará como ponto único de entrada, sendo responsável pelo roteamento das requisições, controle de segurança e abstração da topologia interna dos serviços.

Cada microsserviço possuirá sua própria base de dados, adotando o padrão de persistência poliglota, no qual diferentes tecnologias de armazenamento serão utilizadas de acordo com a natureza dos dados (MARTINS et al., 2024). A comunicação entre os serviços ocorrerá tanto de forma síncrona, via HTTP, quanto de forma assíncrona, por meio de troca de mensagens utilizando um broker de eventos.

Para autenticação e autorização, a arquitetura propõe a utilização de um servidor de identidade baseado no protocolo OAuth 2.0, centralizando o controle de acesso e evitando o acoplamento entre os serviços de domínio e as regras de segurança.

4.3.1 Camada de Apresentação

A interface com o usuário final será materializada através de uma **Aplicação Móvel** que será desenvolvida com o framework **React Native**, permitindo o compartilhamento de lógica de negócio entre as plataformas iOS e Android enquanto mantém o desempenho próximo ao nativo. Para mitigar problemas de latência e permitir o funcionamento em condições de conectividade limitada, a aplicação utilizará o **SQLite** como motor de persistência local, armazenando dados essenciais diretamente no dispositivo. Um diferencial técnico desta camada será a implementação de **Server-Sent Events (SSE)**, que estabelecerá um canal de comunicação unidirecional do servidor para o cliente, possibilitando que o aplicativo receba atualizações de progresso de leitura e novas mensagens em tempo real, sem a necessidade de requisições constantes

4.3.2 API Gateway

O **API Gateway** será um componente central da arquitetura proposta, atuando como intermediário entre os clientes e os microsserviços. Sua principal função é abstrair a complexidade da arquitetura distribuída, oferecendo ao cliente uma interface unificada para acesso às funcionalidades do sistema.

Entre as responsabilidades atribuídas ao API Gateway destacam-se o roteamento de requisições, a aplicação de políticas de segurança, o controle de acesso e o encaminhamento das chamadas aos serviços apropriados. Na proposta arquitetural, o gateway é representado por uma solução baseada em **NGINX**, configurada para operar como ponto de entrada único do sistema.

4.3.3 Serviço de Autenticação e Autorização

A autenticação e autorização dos usuários serão tratadas de forma centralizada por meio de um serviço dedicado, baseado no uso do protocolo **OAuth 2.0**. Na arquitetura proposta, esse papel é desempenhado por uma solução como o **Keycloak**¹, responsável por gerenciar identidades, credenciais, tokens de acesso e permissões.

Esse serviço atuará de forma integrada ao **API Gateway** e aos microsserviços, garantindo que apenas requisições autenticadas e autorizadas tenham acesso às funcionalidades do sistema. Dessa forma, os microsserviços não precisam implementar lógica própria de autenticação, concentrando-se exclusivamente em suas responsabilidades de negócio.

4.3.4 Comunicação Orientada a Eventos (EDA)

A infraestrutura de comunicação do LendoClub será fundamentada no paradigma de **Arquitetura Orientada a Eventos (Event-Driven Architecture - EDA)**. Diferente dos

¹ Disponível em <<https://keycloak.org/>>

modelos tradicionais onde os sistemas dependem exclusivamente de chamadas síncronas que podem gerar gargalos e acoplamento excessivo, a EDA permite que os microsserviços reajam a fatos significativos (eventos) ocorridos no ecossistema de forma independente e assíncrona.

O núcleo desta estratégia é o **Apache Kafka**, que atua como um barramento de eventos de alta performance e durabilidade. No LendoClub, o Kafka funcionará como um log distribuído onde os serviços "produtores" publicam mensagens sobre alterações de estado. Por exemplo, quando o **Serviço de Clubes** registra uma nova interação ou o **Serviço de Livros** atualiza uma obra, um evento é disparado para um tópico específico no Kafka. Os serviços "consumidores", como o **Serviço de Notificações**, monitoram esses tópicos e processam as informações de acordo com sua lógica interna, sem que o serviço de origem precise conhecer a existência ou o estado do destinatário.

Esta abordagem oferece três vantagens críticas para a plataforma:

1. **Resiliência:** Se o serviço de notificações estiver temporariamente indisponível, as mensagens permanecem retidas no Kafka e são processadas retroativamente assim que o serviço for restabelecido, garantindo que nenhum aviso de interação seja perdido.
2. **Escalabilidade:** Permite que o sistema lide com picos de tráfego de forma eficiente, pois o processamento de tarefas pesadas (como envio de e-mails ou indexação de buscas) é retirado do fluxo principal de resposta ao usuário.
3. **Desacoplamento:** facilita a evolução do sistema, permitindo a adição de novos microsserviços que consumam os mesmos eventos sem a necessidade de alterar o código dos serviços já existentes.

Para fluxos que exigem uma resposta imediata e bidirecional — como a validação de credenciais no momento do login — a arquitetura complementa o modelo de eventos com comunicações síncronas via HTTP/REST, criando um ambiente híbrido que equilibra agilidade e robustez técnica.

4.3.4.1 Kafka

O Apache **Kafka**² é definido como uma plataforma de streaming de dados distribuída, projetada para processar volumes massivos de eventos com alta eficiência. Originalmente concebido pelo LinkedIn em 2011 como um sistema de mensageria, o Kafka evoluiu para uma infraestrutura completa de processamento de fluxos, fundamentada na abstração de um log de transações distribuído (NARKHEDE et al., 2017).

² Disponível em <<https://kafka.apache.org/>>

Como ecossistema de streaming, o Kafka destaca-se por oferecer baixa latência, alto rendimento e robusta tolerância a falhas. Segundo a documentação oficial ([Kafka, 2026](#)), a plataforma é capaz de fornecer respostas em milissegundos, o que a torna ideal para sustentar aplicações voltadas à experiência do usuário e para a integração de sistemas em tempo real. Suas capacidades fundamentais podem ser divididas em três pilares:

- **Sistema de Mensageria (*Messaging System*)**: Atua no padrão publicar-subscriver (publish-subscribe), permitindo a troca de dados entre aplicações de forma assíncrona e desacoplada.
- **Sistema de Armazenamento (*Storage System*)**: Diferente de filas tradicionais, o Kafka permite o armazenamento persistente e tolerante a falhas dos fluxos de dados, funcionando como um banco de dados de eventos.
- **Processamento de Fluxos (*Stream Processing*)**: Oferece ferramentas para processar, transformar e reagir a fluxos de eventos no momento em que ocorrem, em vez de depender de processamentos em lote.

O Apache Kafka propõe uma estrutura operacional baseada em papéis distintos: aplicações que atuam como fontes de dados (*producers*) e aquelas que consomem as informações geradas (*consumers*). Nesse modelo, todo o fluxo de mensagens é intermediado pelo *Broker*, que centraliza o controle e a persistência dos dados entre os diversos serviços responsáveis pelo processamento das streams.

4.3.5 Microserviços de Domínio

A arquitetura proposta contemplará a divisão do sistema em diferentes microserviços de domínio, cada um responsável por um conjunto específico de funcionalidades.

4.3.5.1 Serviço de Notificações

O **Serviço de Notificações** atuará como o componente reativo e de engajamento do LendoClub, sendo responsável por garantir que informações pertinentes cheguem aos usuários de forma oportuna e multicanal. Dentro de uma arquitetura de microserviços orientada a eventos, este serviço funciona predominantemente como um consumidor do barramento de mensagens **Apache Kafka**, monitorando tópicos específicos gerados pelos demais serviços de domínio (Clubes e Livros).

A lógica operacional deste serviço baseia-se na escuta de gatilhos disparados na plataforma. Por exemplo, quando o Serviço de Clubes publica um evento de "Novo Tópico de Discussão" ou "Reunião Agendada", o Serviço de Notificações processa esse evento, identifica

os membros afetados e determina a melhor via de entrega. Para notificações de alta prioridade e imediatas, o serviço utiliza Server-Sent Events (SSE) para enviar alertas diretamente à Aplicação Móvel, permitindo que o usuário visualize a notificação em tempo real enquanto navega no aplicativo.

Além da comunicação interna, o serviço será responsável pela integração com Gateway de E-mail. Esta funcionalidade é essencial para fluxos de recuperação de conta, ou convites para novos membros que ainda não possuem o aplicativo instalado. Para gerenciar essas preferências e garantir que o usuário não seja sobrecarregado, o serviço manterá uma estrutura de persistência leve para armazenar configurações de notificação e logs de envio, assegurando que a entrega respeite as escolhas de privacidade de cada membro.

O desacoplamento proporcionado pelo uso do Kafka é especialmente benéfico para este componente: em cenários de alta carga ou falha temporária nos provedores de e-mail externos, as notificações ficam retidas na fila de mensagens e são processadas assim que a conexão é restabelecida, garantindo a confiabilidade da entrega (*at-least-once delivery*) e a integridade da experiência do usuário no LendoClub.

4.3.5.2 Serviço de Clubes

O **Serviço de Clubes** constituirá o núcleo funcional do ecossistema LendoClub, sendo o componente responsável por orquestrar toda a lógica de negócio relativa à criação, manutenção e interação dentro das comunidades de leitura. Este microsserviço será concebido para operar de forma autônoma, gerindo o ciclo de vida completo dos clubes, desde o registo inicial dos metadados (nome, descrição, categorias e regras) até à gestão complexa da hierarquia de membros.

A persistência de dados deste serviço será suportada por uma instância dedicada de **PostgreSQL (PgSQL)**, garantindo a integridade referencial e a consistência transacional necessária para operações críticas, como a adesão de novos membros ou a moderação de permissões de administrador. Para além das operações padrão de CRUD (*Create, Read, Update, Delete*), o serviço se integrará nativamente com o **ElasticSearch**, indexando os dados dos clubes para permitir que os utilizadores realizem buscas avançadas por critérios variados e pesquisas de texto completo (*full-text search*), assegurando uma descoberta de conteúdo eficiente e performática.

No que concerne à interatividade, o serviço será o motor das dinâmicas sociais da plataforma. Ele gerirá as comunicações internas e as interações diretas entre os membros, servindo de mediador para fóruns de discussão e debates literários. Para manter o sistema desacoplado, sempre que ocorre uma ação relevante dentro de um clube — como o agendamento

de um novo encontro ou a publicação de um tópico — o serviço publicará um evento no **Apache Kafka**. Este modelo permite que outros módulos, como o Serviço de Notificações, reajam de forma assíncrona, garantindo que os participantes recebam atualizações em tempo real sem comprometer o tempo de resposta da operação principal.

Finalmente, a segurança das operações dentro deste serviço será validada em conformidade com as políticas do **Keycloak**. Cada requisição enviada ao Serviço de Clubes será verificada para assegurar que apenas membros autorizados possam realizar edições ou aceder a conteúdos restritos de clubes privados, mantendo a privacidade e a organização dos grupos conforme delineado nos requisitos funcionais do projeto.

4.3.5.3 Serviço de Livros

O **Serviço de Livros** será o componente responsável pela gestão do catálogo literário do LendoClub, atuando como o provedor central de metadados para todas as funcionalidades que envolvem obras bibliográficas. Diferente de outros módulos, este serviço operará sob uma estratégia de dados híbrida e integração intensa com serviços de terceiros, garantindo que o usuário tenha acesso a informações detalhadas, capas de livros e sinopses atualizadas.

A persistência de dados neste microsserviço utilizará o **MongoDB**, um banco de dados orientado a documentos. Esta escolha justifica-se pela natureza semiestruturada dos dados de livros, que podem variar significativamente em termos de atributos (como diferentes edições, múltiplos autores ou categorias variáveis). O uso do MongoDB confere a flexibilidade necessária para armazenar esses objetos complexos sem a rigidez de um esquema relacional tradicional. Para buscas de alto desempenho, os dados serão também indexados no **ElasticSearch**, permitindo que o sistema responda a consultas complexas e buscas por palavras-chave com baixa latência.

Um ponto crítico desta subseção é a integração com a **Google Books API**. O serviço foi projetado para atuar como um mediador inteligente: quando um livro é pesquisado ou adicionado a um clube, o sistema consumirá a API da Google para buscar informações oficiais, enriquecendo a base de dados local. Para otimizar o consumo de banda e melhorar a performance da aplicação móvel, o serviço gerenciará o armazenamento de mídias (como imagens de capas) utilizando soluções de **Object Storage** (MinIO para ambiente local e AWS S3 para produção), evitando que o aplicativo precise realizar requisições externas repetitivas para fontes de terceiros.

Em termos de comunicação interna, o Serviço de Clubes interage com o Serviço de Livros via **HTTP**.

4.3.6 Integração com Serviços Externos

A principal integração de dados ocorre com a **Google Books API**. Através deste serviço, o **Serviço de Livros** consegue realizar consultas em tempo real para recuperar metadados bibliográficos precisos, como números de ISBN, sinopses detalhadas, categorias literárias e links para as artes das capas. Esta conexão é realizada via protocolo HTTPS, garantindo que o catálogo do aplicativo esteja sempre alinhado com o maior repositório de livros do mundo, facilitando a busca e a organização das leituras pelos administradores dos clubes.

No âmbito da comunicação externa, a arquitetura se integrará a um **Gateway de E-mail**. Esta ponte é vital para garantir que o ciclo de engajamento do usuário não se limite apenas ao ambiente do aplicativo móvel. O gateway será acionado pelo **Serviço de Notificações** para fluxos críticos de transação e segurança, como a validação de novos cadastros, recuperação de senhas e o envio de resumos de atividades para usuários que optam por receber atualizações via correio eletrônico.

Adicionalmente, para a gestão de ativos digitais e arquivos de mídia, o sistema será projetado para suportar **Object Storage** em nuvem, especificamente o **AWS S3** (ou o **MinIO** em cenários de desenvolvimento e infraestrutura local). Esta integração permitirá que o LendoClub armazene e recupere grandes volumes de dados não estruturados de maneira escalável e segura, garantindo que o carregamento de imagens e documentos na aplicação móvel seja otimizado através de redes de entrega de conteúdo (CDNs), reduzindo a carga direta nos servidores de aplicação.

4.3.7 Persistência de Dados

A estratégia de armazenamento do LendoClub adota o conceito de **Persistência Poliglota**, que consiste na utilização de diferentes tecnologias de banco de dados para resolver problemas específicos dentro de uma arquitetura de microsserviços. Em vez de forçar um único modelo de dados para toda a aplicação, cada microserviço deterá e gerenciará sua própria base de dados, utilizando o motor que oferece o melhor desempenho e integridade para o seu domínio de negócio.

4.3.7.1 Bancos de Dados Relacionais (PostgreSQL)

Para os dados que exigem alta integridade referencial e conformidade ACID (Atomicidade, Consistência, Isolamento e Durabilidade), a arquitetura utilizará o **PostgreSQL (PgSQL)**. Este banco de dados é o alicerce do **Serviço de Clubes**, onde é necessário gerenciar relações complexas entre usuários, perfis de administrador, permissões e associações a grupos. A robustez do PgSQL garantirá que transações críticas, como a entrada de um membro em

um clube privado, ocorram de forma segura e consistente.

4.3.7.2 Bancos de Dados Orientados a Documentos (MongoDB)

No **Serviço de Livros**, a persistência será realizada através do **MongoDB**, um banco de dados NoSQL orientado a documentos. Esta escolha é justificada pela natureza heterogênea dos metadados literários. Como as informações provenientes da Google Books API podem variar em estrutura de um título para outro, o MongoDB permitirá armazenar esses objetos JSON de forma flexível, sem a necessidade de migrações de esquema (*schema migrations*) constantes, facilitando a evolução do catálogo.

4.3.7.3 Object Storage (MinIO e AWS S3)

Para o armazenamento de arquivos binários e dados não estruturados, como imagens de capa de livros, fotos de perfil e documentos compartilhados nos clubes, o sistema utilizará o conceito de **Object Storage**. Em ambientes de desenvolvimento e infraestrutura local (on-premise), utilizará o **MinIO**, que oferece uma API compatível com o padrão de mercado. Para o ambiente de produção em larga escala, a arquitetura está preparada para integrar-se ao **AWS S3**, garantindo alta disponibilidade e durabilidade dos arquivos.

4.3.7.4 Indexação e Busca Full-Text (ElasticSearch)

Complementando a persistência, o **ElasticSearch** atuará como uma camada de recuperação de dados de alta performance. Ele não substitui os bancos de dados principais, mas funciona como um motor de busca secundário que indexa as informações do PostgreSQL e do MongoDB. Isso permitirá que o LendoClub ofereça filtros avançados, buscas por proximidade fonética e pesquisas em grandes volumes de texto com respostas em milissegundos, algo que bancos de dados tradicionais teriam dificuldade em processar sob carga intensa.

4.3.7.5 Persistência Local (SQLite)

Finalmente, na camada de apresentação, a **Aplicação Móvel** fará uso do **SQLite**. Este banco de dados leve reside diretamente no dispositivo do usuário, servindo como um cache inteligente para armazenar dados de sessão, preferências e informações básicas de leitura. Essa estratégia de persistência local será vital para garantir que a interface permaneça responsiva e que informações essenciais estejam disponíveis mesmo em cenários de instabilidade de rede.

5 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo principal apresentar uma proposta arquitetural para um sistema de gerenciamento de clubes de leitura, considerando os desafios atuais relacionados à promoção da leitura, à organização de comunidades leitoras e ao uso de tecnologias digitais como mediadoras desse processo. Diferentemente de abordagens focadas exclusivamente na implementação, o estudo concentrou-se na definição e detalhamento da arquitetura de software, buscando evidenciar decisões técnicas coerentes com os requisitos do domínio e com boas práticas consolidadas na literatura.

Ao longo do desenvolvimento do trabalho, foi possível contextualizar a importância da leitura e dos clubes de leitura no cenário contemporâneo, bem como o papel das plataformas digitais no incentivo ao hábito de ler. A partir desse contexto, foram levantados os requisitos funcionais e não funcionais do sistema, que serviram de base para a definição de uma arquitetura capaz de atender a aspectos como escalabilidade, manutenibilidade, segurança e flexibilidade evolutiva.

A arquitetura proposta fundamenta-se no estilo de microsserviços, aliada a princípios de arquitetura orientada a eventos, utilizando tecnologias amplamente adotadas no mercado. A separação do sistema em microsserviços de domínio — como os serviços de clubes, livros e notificações — contribui para o desacoplamento entre responsabilidades e para a possibilidade de evolução independente de cada componente. A integração com serviços externos, como a Google Books API, e o uso de mecanismos especializados, como full-text search e object storage, reforçam a adequação da arquitetura às demandas específicas do domínio da aplicação.

Do ponto de vista acadêmico, o trabalho contribui ao apresentar uma visão estruturada de como conceitos teóricos relacionados à arquitetura de software podem ser aplicados de forma prática na proposição de um sistema realista, ainda que não implementado em sua totalidade. A proposta arquitetural elaborada pode servir como base tanto para trabalhos futuros quanto para uma eventual implementação do sistema em um contexto prático ou experimental.

Como limitações, destaca-se o fato de que o escopo do trabalho concentrou-se na proposta e no detalhamento arquitetural, não contemplando a implementação completa do sistema nem a validação empírica da arquitetura por meio de testes de desempenho, escalabilidade ou usabilidade.

Como trabalhos futuros, sugere-se a implementação gradual dos microsserviços propostos, bem como a realização de experimentos que permitam avaliar o comportamento da arquitetura em cenários reais de uso.

Por fim, conclui-se que os objetivos propostos neste trabalho foram alcançados, uma vez que foi possível definir e descrever uma arquitetura de software coerente, fundamentada teoricamente e alinhada às necessidades do domínio de clubes de leitura, contribuindo para a discussão acadêmica sobre o uso de arquiteturas modernas no desenvolvimento de sistemas voltados à promoção da leitura.

REFERÊNCIAS

Amazon Brasil. **Amazon Brasil**. 2024. Acesso em: 21 set. 2024. Disponível em: <<https://www.amazon.com.br>>.

Auth0. **What is OAuth 2.0?** 2026. Disponível em: <<https://auth0.com/pt/intro-to-iam/what-is-oauth-2>>. Acesso em: 26 jan. 2026. Disponível em: <<https://auth0.com/pt/intro-to-iam/what-is-oauth-2>>.

AWS. **What is Object Storage?** 2026. Disponível em: <<https://aws.amazon.com/what-is/object-storage/>>. Acesso em: 26 jan. 2026. Disponível em: <<https://aws.amazon.com/what-is/object-storage/>>.

CAZARRÉ, R. R. **Comunidades Virtuais no TikTok: As Implicações do BookTok no Consumo de Livros da Geração Z da Grande Porto Alegre**. Trabalho de Conclusão de Curso — Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil, 2023. Disponível em: <<http://hdl.handle.net/10183/258859>>.

CHOU, D. **Using Events in Highly Distributed Architectures**. 2008. Disponível em: <<https://msdn.microsoft.com/en-us/library/dd129913.aspx>>. Acesso em: 26 jan. 2026. Disponível em: <<https://msdn.microsoft.com/en-us/library/dd129913.aspx>>.

DRAGONI, N.; GIALLORENZO, S.; LLUCH-LAFUENTE, A.; MAZZARA, M.; MONTESI, F.; MUSTAFIN, R.; SAFINA, L. Microservices: yesterday, today, and tomorrow. **CoRR**, abs/1606.04036, 2016. Disponível em: <<https://arxiv.org/abs/1606.04036>>.

EMERGEMAG. **BookTok movimentando mercado editorial e literário**. 2023. Acesso em: 21 set. 2024. Disponível em: <<https://emergemag.com.br/booktok-movimentando-mercado-editorial-e-literario/>>.

ERL, T. **Service-Oriented Architecture (Paperback): Concepts, Technology, and Design**. [S.l.]: Prentice Hall, 2016. (The Prentice Hall Service Technology Series).

FOWLER, M.; LEWIS, J. **Microservices: a definition of this new architectural term**. 2014. Acesso em: 26 jan. 2026. Disponível em: <<https://martinfowler.com/articles/microservices.html>>.

G1. **BookTok: como TikTok está transformando jovens em leitores e autores em best-sellers**. 2024. Acesso em: 21 set. 2024. Disponível em: <<https://g1.globo.com/pop-arte/noticia/2024/01/16/booktok-como-tiktok-esta-transformando-jovens-em-leitores-e-autores-em-best-sellers.ghml>>.

Google. **Tendências de Busca**. 2024. Acesso em: 12/08/2024. Disponível em: <<https://trends.google.com>>.

Institute of Electrical and Electronics Engineers. **IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990)**. New York, NY, USA: IEEE, 1990. ISBN 1-55937-067-X. Disponível em: <https://www.informatik.htw-dresden.de/~hauptman/SEI/IEEE_Standard_Glossary_of_Software_Engineering_Terminology%20.pdf>.

Instituto Pró-Livro. **Retratos da Leitura no Brasil - 5ª Edição**. 2019. Acesso em: 21 set. 2024. Disponível em: <<https://www.prolivro.org.br/5a-edicao-de-retratos-da-leitura-no-brasil-2/a-pesquisa-5a-edicao/>>.

Instituto Pró-Livro. **Perfil dos Leitores da Bienal do Livro SP 2022**. 2022. Acesso em: [data de acesso]. Disponível em: <https://www.prolivro.org.br/wp-content/uploads/2022/08/Pesquisaltau%CC%81Cultural_IPL_PerfilLeitoresdaBienaldoLivroSP22-2-1.pdf>.

Kafka. **Introduction**. 2026. Disponível em: <<https://kafka.apache.org/intro>>. Acesso em: 26 jan. 2026. Disponível em: <<https://kafka.apache.org/intro>>.

KUMARI, S.; RATH, S. K. Performance comparison of soap and rest based web services for enterprise application integration. In: **International Conference on Advances in Computing, Communications and Informatics (ICACCI)**. [S.l.]: [s.n.], 2015. p. 1656–1660.

MARTINS, L. S. S.; SILVA, J. V. F. d.; SCHERER, A. P. Z. Persistência poliglota: o armazenamento de dados e suas possibilidades. Porto Alegre, 2024.

Microsoft. **Microsoft application architecture guide**. 2009. Disponível em: <<https://msdn.microsoft.com/en-us/library/ff650706.aspx>>. Acesso em: 26 jan. 2026. Disponível em: <<https://msdn.microsoft.com/en-us/library/ff650706.aspx>>.

NARKHEDE, N.; SHAPIRA, G.; PALINO, T. **Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale**. [S.l.]: O'Reilly Media, 2017.

O'CONNOR, R. V.; ELGER, P.; CLARKE, P. M. Continuous software engineering—a microservices architecture perspective. **Journal of Software: Evolution and Process**, [S.l.], v. 29, n. 11, p. 1–12, 2017.

ODOFIN, O. T.; OWOADE, S.; OGBUEFI, E.; OGEAWUCHI, J. C.; ADANIGBO, O. S.; GBENLE, T. P. Integrating event-driven architecture in fintech operations using apache kafka and rabbitmq systems. **International Journal of Multidisciplinary Research and Growth Evaluation**, v. 03, n. 04, p. 635–643, 2022. Disponível em: <https://www.allmultidisciplinaryjournal.com/uploads/archives/20250604133710_MGE-2025-3-208.1.pdf>. Acesso em: 26 jan. 2026. Disponível em: <https://www.allmultidisciplinaryjournal.com/uploads/archives/20250604133710_MGE-2025-3-208.1.pdf>.

SCHMITZ-BOCCIA, A. Clubes de leitura: a construção de sentidos em situações de leitura colaborativa. **Veras**, ISE Vera Cruz, v. 2, n. 1, p. 97, maio 2012. ISSN 2236-5729. Disponível em: <<http://dx.doi.org/10.14212/veras.vol2.n1.ano2012.art81>>.

SILVEIRA, P. **Introdução à arquitetura e design de software: uma visão sobre a plataforma Java**. Rio de Janeiro, Brasil: Elsevier, 2012.

SOMMERVILLE, I. **Software Engineering**. 4. ed. Wokingham, England: Addison-Wesley, 1992. ISBN 9780201565294.

USP, J. da. **BookTok: fenômeno das redes sociais impacta na venda e visibilidade de obras literárias**. 2023. Acesso em: 21 set. 2024. Disponível em: <<https://jornal.usp.br/atualidades/booktok-fenomeno-das-redes-sociais-impacta-na-venda-e-visibilidade-de-obras-literarias/>>.

VILLAMIZAR, M. et al. Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud. In: **10th Computing Colombian Conference**. [S.l.]: [s.n.], 2015. p. 583–590.

WANG, Z. **Architectural Design and Analysis of Microservice-Based Systems**. Tese (Doutorado) — University of Auckland, Auckland, New Zealand, 2022. Tese (Doutorado em Ciência da Computação). Disponível em: <<https://researchspace.auckland.ac.nz/handle/2292/XXXXX>>.

ZHAO, J. T.; JING, S. Y.; JIANG, L. Z. Management of api gateway based on micro-service architecture. In: **Journal of Physics: Conference Series**. IOP Publishing, 2018. v. 1087, n. 3, p. 032032. Disponível em: <<https://doi.org/10.1088/1742-6596/1087/3/032032>>.

APÊNDICE A – USER STORIES

Esse apêndice agrupa todas as *user stories* levantadas inicialmente.

Quadro A.1 – US 01 - Registro de Usuário

US 01 - Registro de Usuário			
Como um usuário, quero me registrar no sistema para poder acessar todas as funcionalidades disponíveis. O registro deve incluir campos de email, confirmação de email, senha, confirmação de senha, nome, apelido, sexo, gênero, data de nascimento, país, estado e cidade.			
Story Points	3	Prioridade	Alta
TA1.1	Verificar se o usuário pode se registrar com sucesso.		
TA1.2	Validar campos obrigatórios e únicos.		

Fonte: Elaborado pelo autor (2026)

Quadro A.2 – US 02 - Login de Usuário

US 02 - Login de Usuário			
Como um usuário registrado, quero fazer login no sistema para acessar minhas informações pessoais e atividades. O login deve ser feito usando email e senha.			
Story Points	3	Prioridade	Alta
TA2.1	Verificar se o login é realizado com credenciais corretas.		
TA2.2	Testar mensagens de erro para falhas.		

Fonte: Elaborado pelo autor (2026)

Quadro A.3 – US 03 - Criação de Clube

US 03 - Criação de Clube			
Como um usuário, quero criar um novo clube de livros para organizar leituras e atividades com outros membros. A criação do clube deve incluir nome do clube, descrição, regras e critérios de adesão.			
Story Points	6	Prioridade	Alta
TA3.1	Testar a criação de novos clubes.		
TA3.2	Verificar campos obrigatórios.		
TA3.3	Verificar a associação do criador como administrador.		

Fonte: Elaborado pelo autor (2026)

Quadro A.4 – US 04 - Criação de Salas de Bate-Papo

US 04 - Criação de Salas de Bate-Papo			
Como um administrador de clube, quero criar salas de bate-papo específicas para discussões e atividades do clube. Cada sala deve ter um título, descrição e lista de participantes.			
Story Points	4	Prioridade	Média
TA4.1	Testar a criação de salas de bate-papo.		

Fonte: Elaborado pelo autor (2026)

Quadro A.5 – US 05 - Aprovação de Registros

US 05 - Aprovação de Registros			
Como um administrador, quero aprovar ou rejeitar novos registros de usuários para garantir a segurança do clube. O administrador deve receber notificações de novos registros pendentes.			
Story Points	5	Prioridade	Média
TA5.1	Testar a funcionalidade de aprovação e rejeição de usuários.		
TA5.2	Verificar se o administrador recebe notificação de novos registros.		

Fonte: Elaborado pelo autor (2026)

Quadro A.6 – US 06 - Edição de Perfil

US 06 - Edição de Perfil			
Como um usuário, quero poder editar meu perfil para atualizar minhas informações pessoais, incluindo email, senha, nome, apelido, sexo, gênero, data de nascimento, país, estado e cidade.			
Story Points	3	Prioridade	Média
TA6.1	Verificar se as informações do perfil podem ser atualizadas.		
TA6.2	Testar persistência dos dados.		

Fonte: Elaborado pelo autor (2026)

Quadro A.7 – US 07 - Adição de Livros do mês

US 07 - Adição de Livros do mês			
Como um administrador, quero adicionar novos livros ao feed do clube para que membros possam vê-los e avaliá-los. O cadastro deve incluir capa do livro, título, autor, gênero, ISBN e sinopse.			
Story Points	3	Prioridade	Alta
TA7.1	Verificar se livros podem ser adicionados.		
TA7.2	Validar campos obrigatórios.		

Fonte: Elaborado pelo autor (2026)

Quadro A.8 – US 08 - Pesquisa de Livros

US 08 - Pesquisa de Livros			
Como um administrador, quero pesquisar livros para adicionar ao feed do clube, a pesquisa poderá ser feita por título, autor e gênero.			
Story Points	3	Prioridade	Média
TA8.1	Testar funcionalidade de pesquisa.		
TA8.2	Verificar se resultados batem.		

Fonte: Elaborado pelo autor (2026)

Quadro A.9 – US 09 - Criação de Eventos

US 09 - Criação de Eventos			
Como um administrador, quero criar eventos e reuniões de leitura para organizar as atividades do clube. O evento deve incluir título, descrição, data, hora e local.			
Story Points	5	Prioridade	Média
TA9.1	Testar a criação e edição de eventos.		
TA9.2	Verificar a exibição no feed.		
TA9.3	Testar se usuários do clube recebem notificação de eventos.		

Fonte: Elaborado pelo autor (2026)

Quadro A.10 – US 10 - Lembretes de Eventos

US 10 - Lembretes de Eventos			
Como um usuário, quero receber lembretes de eventos para não esquecer das reuniões e eventos do clube. Os lembretes devem ser enviados em forma de notificação do app.			
Story Points	4	Prioridade	Média
TA10.1	Testar envio de lembretes de eventos como notificação.		

Fonte: Elaborado pelo autor (2026)

Quadro A.11 – US 11 - Avaliações e Comentários

US 11 - Avaliações e Comentários			
Como um usuário, quero poder avaliar os livros do mês do clube quando terminar a leitura. As avaliações devem conter uma nota e um comentário, com possibilidade de edição.			
Story Points	4	Prioridade	Média
TA11.1	Testar funcionalidade de avaliação e comentários.		
TA11.2	Verificar exibição das avaliações e comentários.		

Fonte: Elaborado pelo autor (2026)

Quadro A.12 – US 12 - Relatórios de Atividade

US 12 - Relatórios de Atividade			
Como um administrador, quero acessar relatórios sobre as atividades do clube, como avaliação dos livros por participante e se o participante terminou o livro ou não.			
Story Points	5	Prioridade	Alta
TA12.1	Testar geração e exibição de relatórios.		
TA12.2	Verificar precisão dos dados.		

Fonte: Elaborado pelo autor (2026)

Quadro A.13 – US 13 - Gerenciamento de Cargos

US 13 - Gerenciamento de Cargos			
Como um administrador, quero gerenciar os cargos de usuários para ter a possibilidade de transformar outros usuários em administradores.			
Story Points	5	Prioridade	Alta
TA13.1	Testar atribuição e modificação de cargos.		
TA13.2	Somente validar mediante senha.		

Fonte: Elaborado pelo autor (2026)

Quadro A.14 – US 14 - Participação em Múltiplos Clubes

US 14 - Participação em Múltiplos Clubes			
Como um usuário, quero poder participar de múltiplos clubes de livros. O sistema deve permitir fácil navegação entre os diferentes feeds dos clubes dos quais participo.			
Story Points	5	Prioridade	Alta
TA14.1	Verificar a participação em múltiplos clubes.		
TA14.2	Testar a exibição correta dos clubes em que o usuário participa.		

Fonte: Elaborado pelo autor (2026)

Quadro A.15 – US 15 - Participação em Salas de Bate-Papo

US 15 - Participação em Salas de Bate-Papo			
Como um usuário, quero participar de salas de bate-papo do clube para discutir o progresso dos livros. As salas devem suportar mensagens de texto em tempo real.			
Story Points	4	Prioridade	Média
TA15.1	Testar participação e interações em salas de bate-papo.		

Fonte: Elaborado pelo autor (2026)

Quadro A.16 – US 16 - Enquetes e Votações

US 16 - Enquetes e Votações			
Como um administrador, quero criar enquetes e votações para obter a opinião dos membros do clube sobre livros, eventos ou outras decisões do clube.			
Story Points	4	Prioridade	Média
TA16.1	Testar a criação, edição e participação em enquetes.		
TA16.2	Verificar a exibição dos resultados das votações.		

Fonte: Elaborado pelo autor (2026)

Quadro A.17 – US 17 - Progresso de Leitura

US 17 - Progresso de Leitura			
Como um usuário, quero poder atualizar meu progresso de leitura.			
Story Points	3	Prioridade	Baixa
TA17.1	Testar funcionalidade de progresso de leitura.		
TA17.2	Verificar exibição correta do progresso.		

Fonte: Elaborado pelo autor (2026)

Quadro A.18 – US 18 - Histórico de mensagens nos chats

US 18 - Histórico de mensagens nos chats	
Como um novo participante de um clube de leitura, quero ter acesso ao histórico completo de mensagens em todos os bate-papos criados pelo administrador, para que eu possa me inteirar das discussões anteriores e participar de forma mais engajada.	
Story Points	5
Prioridade	Alta
TA18.1	Verificar se o histórico de mensagens está sendo mantido em todos os bate-papos.
TA18.2	Testar se um novo participante consegue visualizar todas as mensagens anteriores ao seu ingresso no chat.
TA18.3	Confirmar que as mensagens aparecem em ordem cronológica e são atualizadas em tempo real para todos os membros.

Fonte: Elaborado pelo autor (2026)

	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)

TCC finalizado

Assunto:	TCC finalizado
Assinado por:	Danrlei Silva
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Danrlei de Lira Silva, DISCENTE (202212010003) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS, em 20/02/2026 10:43:35.

Este documento foi armazenado no SUAP em 20/02/2026. 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: 1771827
Código de Autenticação: 5e22a0d79c

