

Campus Ji-Paraná
Curso Superior de Tecnologia em Análise e Desenvolvimento de
Sistemas

PROJETO DO SISTEMA BIBLIOTECA HELENA RAMAZZOTTE
Cliente: Igreja Missionária Unida do Brasil

FABIANO BRITO DE SOUZA

**PROJETO DO SISTEMA BIBLIOTECA HELENA RAMAZZOTTE
1.0**

Projeto de Conclusão de Curso SUPERIOR EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS, como parte dos requisitos necessários para a aprovação em Tecnólogo em Análise e Desenvolvimento de Sistemas. Área de Concentração: Ciências Exatas e da Terra.

Orientador (a): Prof. Me. Gleison Guardia

Ji-Paraná, 2025

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO.

Souza, Fabiano Brito de.
Projeto do sistema biblioteca Helena Ramazzotte / Fabiano Brito de
Souza. - Ji-Paraná, 2025.
117 f.

Orientador(a): Me. Gleison Guardia.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e
Tecnologia de Rondônia - IFRO, Ji-Paraná, 2025.

1. Biblioteca digital. 2. Sistemas web. 3. Metodologias ágeis. 4.
Gerenciamento de acervo. 5. Comunidade de leitura. I. Guardia, Gleison
(orient.). II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia
- IFRO. III. Título.

Bibliotecário(a) Responsável: Cleuza Diogo Antunes, CRB-11/864



ATA DE DEFESA DE MONOGRAFIA

Na data 19/08/2025 realizou-se a sessão pública de defesa da Monografia intitulada **SISTEMA BIBLIOTECA HELENA RAMAZZOTTE** apresentada pelo aluno **Fabiano Brito de Souza (2022202060031)** do Curso **Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (Ji-Paraná)**. Os trabalhos foram iniciados às **20:36** pelo Professor **Gleison Guardia** presidente da banca examinadora, constituída pelos seguintes membros:

- **Gleison Guardia** (Orientador)
- **Jefferson Antonio dos Santos** (Examinador Interno)
- **Karan Luciano Silva di Pernis** (Examinador Externo)

A banca examinadora, tendo terminado a apresentação do conteúdo da Monografia, passou à arguição do candidato. Em seguida, os examinadores reuniram-se para avaliação e deram o parecer final sobre o trabalho apresentado pelo aluno, tendo sido atribuído o seguinte resultado:

APROVADO

Nota: 90

Proclamados os resultados pelo presidente da banca examinadora, foram encerrados os trabalhos e, para constar, eu **Gleison Guardia** lavrei a presente ata que assino juntamente com os demais membros da banca examinadora.

Ji-PARANÁ / RO, 19/08/2025

Documento assinado eletronicamente por **Fabiano Brito de Souza**, Discente, em 20/08/2025, às 21:21, conforme horário oficial de Rondônia, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Gleison Guardia**, Orientador, em 20/08/2025, às 20:38, conforme horário oficial de Rondônia, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Jefferson Antonio dos Santos**, Examinador Interno, em 20/08/2025, às 20:42, conforme horário oficial de Rondônia, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Karan Luciano Silva di Pernis**, Examinador Externo, em 21/08/2025, às 10:47, conforme horário oficial de Rondônia, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

PROJETO DO SISTEMA BIBLIOTECA HELENA RAMAZZOTTE

1.0

Fabiano Brito de Souza

Gleison Guardia

Resumo: O presente trabalho descreve o desenvolvimento e a implementação de um sistema de gerenciamento de biblioteca baseado em tecnologias web, direcionado à Biblioteca Helena Ramazzotte, da Igreja Missionária Unida do Brasil. O projeto surgiu da necessidade de superar as limitações do modelo manual de controle de empréstimos, devoluções e usuários, realizado por meio de anotações e planilhas, que geravam erros e ineficiência. A aplicação proposta foi estruturada com tecnologias modernas, como Node.js, Next.js, React, MySQL e CSS, garantindo responsividade, segurança e escalabilidade. O sistema oferece recursos de cadastro de usuários e livros, relatórios gerenciais, notificações automáticas e criação de comunidades de leitura, promovendo engajamento entre os leitores. O desenvolvimento seguiu metodologias ágeis, em especial o Scrum, permitindo organização do backlog, prototipagem e validação com os usuários. Os resultados apontam que a solução contribui para a eficiência administrativa, aumenta a confiabilidade das informações e proporciona uma experiência mais interativa e acessível para leitores e gestores da biblioteca.

Palavras-chave: Biblioteca digital; Sistemas web; Metodologias ágeis; Gerenciamento de acervo; Comunidade de leitura.

Resumo: This work describes the development and implementation of a web-based library management system for the Helena Ramazzotte Library, belonging to the Igreja Missionária Unida do Brasil. The project emerged from the need to overcome the limitations of the manual control model, which used notes and spreadsheets to manage loans, returns, and users, resulting in errors and inefficiency. The proposed application was structured with modern technologies such as Node.js, Next.js, React, MySQL, and CSS, ensuring responsiveness, security, and scalability. The system provides features such as user and book registration, management reports, automated notifications, and the creation of reading communities, fostering engagement among readers. The development followed agile methodologies, especially Scrum, allowing backlog organization, prototyping, and validation with users. The results indicate that the solution contributes to administrative efficiency, increases the reliability of information, and provides a more interactive and accessible experience for readers and library managers.

Palavras-chave: Digital library; Web systems; Agile methodologies; Collection management; Reading community.

LISTA DE FIGURAS

Figura 01: Tela Inicial JesusCopy	13
Figura 02: Tela inicial Livrarias Família Cristã	14
Figura 03: Controle de livros	16
Figura 04: Tela de comunidade do novo sistema	18
Figura 05: Ranking linguagens mais utilizadas	20
Figura 06: Estrutura aplicação web	24
Figura 07: Api rest	27
Figura 10: Exemplo Assinatura	29
Figura 11: Exemplo JWT decodificada.....	30
Figura 12: Exemplo HTML estrutura certa	32
Figura 13: Exemplo de estruturação com tags semânticas:	34
Figura 14: SSR e SSG	36
Figura 15 : Diagrama de caso uso	58
Figura 16: Diagrama de Banco de Dados	81
Figura 17: de Diagrama de Classes.....	82
Figura 18: Página de login	84
Figura 19: Página de login validação de campos vazios	84
Figura 20: Página de login validação de usuário e senha.....	85
Figura 21: Página de cadastre-se aba dados de usuário.....	86
Figura 22: Página de cadastre-se aba de endereço do usuário	86
Figura 23: Página de menu lateral usuário: Administrador	87
Figura 24: Página de menu lateral usuário: Voluntário	88
Figura 25: Página de menu lateral usuário: Leitor	89
Figura 26: Página de home	90
Figura 27: Página de detalhes do livro.....	90
Figura 28: Página de detalhes do livro se o livro estiver indisponível.....	91
Figura 29: Página de detalhes do livro aba avaliações.....	91
Figura 30: Página de cadastrar livro parte1	92
Figura 31: Página de cadastrar livro parte2	92
Figura 32: Página de aprovar empréstimo	93
Figura 33: Página empréstimos	94
Figura 34: Página de usuário	94
Figura 35: Página de detalhes do usuário.....	95
Figura 36: Página de detalhes do usuário mudar nível de usuário	95
Figura 37: Página de sugira ou doe um livro parte 1	96
Figura 38: Página de sugira ou doe um livro parte 2	96
Figura 39: Página de criar comunidade	97
Figura 40: Página de comunidade	98
Figura 41: Página de relatórios	98

Figura 42: Página de acesso rápido à comunidade	99
Figura 43: Página de perfil do usuário	99
Figura 44: Página de notificações	100
Figura 45: Página de indicações semanais.....	101

LISTA DE QUADROS

Quadro 01 - Equipe de Desenvolvimento	42
Quadro 02 - Product Backlog	42
Quadro 03: Fazer Login	50
Quadro 04: Cadastre-se	51
Quadro 05: Esqueceu senha?	51
Quadro 06: Redefinir Senha	52
Quadro 07: Home	52
Quadro 08: Cadastrar Livro	52
Quadro 09: Consultar Livro	53
Quadro 10: Lista de Empréstimos	54
Quadro 11: Aprovar Empréstimos	54
Quadro 12: Sugere/doe um livro	54
Quadro 13: Criar Comunidade	55
Quadro 14: Comunidade	55
Quadro 15: Usuários	56
Quadro 16: Consultar Usuário	56
Quadro 17: Conta do Usuário	56
Quadro 18: Relatórios	57
Quadro 19: Notificações	57

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface (Interface de Programação de Aplicações)

CSS – Cascading Style Sheets (Folhas de Estilo em Cascata)

HTML – HyperText Markup Language (Linguagem de Marcação de Hipertexto)

SP – São Paulo

PR – Paraná

IDE – Integrated Development Environment (Ambiente de Desenvolvimento Integrado)

REST – Representational State Transfer (Transferência de Estado Representacional)

JWT – JSON Web Token (Token Web baseado em JSON)

SQL – Structured Query Language (Linguagem de Consulta Estruturada)

MDN – Mozilla Developer Network (Rede de Desenvolvedores Mozilla)

HTTP – Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto)

AWS – Amazon Web Services (Serviços Web da Amazon)

SSR – Server-Side Rendering (Renderização no Lado do Servidor)

SSG – Static Site Generation (Geração de Site Estático)

SUMÁRIO

1. TEMA	10
1.1. DELIMITAÇÃO DO TEMA	10
1.2. PROBLEMÁTICA	10
2. JUSTIFICATIVA	11
3. HIPÓTESE	11
4. OBJETIVOS	11
4.1. OBJETIVO GERAL	11
4.2. OBJETIVOS ESPECÍFICOS	11
5. FUNDAMENTAÇÃO TEÓRICA	12
5.1. ANÁLISE DE SISTEMAS DE BIBLIOTECAS EXISTENTES	12
5.1.1. Bibliotecas Cristãs e sua Relevância	12
5.1.2. JesusCopy	13
5.1.3. LivrariaS Família Cristã	14
5.2. COMPARAÇÃO ENTRE O SISTEMA ATUAL E SISTEMA NOVO	14
5.2.1. Sistema antigo/atual	15
5.2.2. Sistema novo	16
5.3. LINGUAGENS DE PROGRAMAÇÃO	18
5.3.1. Paradigma de Programação	20
5.3.2. Javascript	21
5.3.3. Typescript	22
5.4. DESENVOLVIMENTO WEB	23
5.4.1. Back-end	24
5.4.1.1. Node.js	24
5.4.1.2. APIs RESTful	25
5.4.1.3. Gerenciamento de autenticação com JWT (JSON Web Token)	27
5.4.2. Front-end	30
5.4.2.1. HTML	31
5.4.2.2. CSS	34
5.4.2.3. Next.js e React	34
5.5. Banco de dados	36
5.5.1. MySQL	37
5.6. Metodologias ágeis	37
5.6.1. Scrum	38
6. METODOLOGIA	41
6.1. ESCOPO DE PESQUISA	41
6.2. METODOLOGIA ÁGEIS	41
6.2.1. SCRUM	41
6.2.1.1. EQUIPE DE DESENVOLVIMENTO	42
6.2.1.2. PRODUCT BACKLOG	42
6.3. LEVANTAMENTO DE REQUISITOS	44
6.3.1. ENTREVISTAS COM CLIENTE	44

6.3.1.1. Entrevista 1	44
6.3.1.2. Entrevista 2	47
6.3.1.3. Entrevista 3	48
6.3.2. DOCUMENTO DE REQUISITOS	50
6.4. CASO DE USO	58
6.4.1. Diagrama de caso uso	58
6.4.2. Diagrama de caso uso expandido	59
6.4.3. Diagrama de Banco de Dados	80
6.4.4. Diagrama de classes	81
7. RESULTADO E DISCUSSÕES	83
7.1. Aplicação front-end	83
7.1.1. Página de login	83
7.1.2. Página de cadastre-se	85
7.1.3. Menu Lateral	87
7.1.4. Página de home	89
7.1.5. Página de detalhes do livro	90
7.1.6. Página de cadastrar livro	92
7.1.7. Página de aprovar empréstimo	93
7.1.8. Página de empréstimos	93
7.1.9. Página de usuário	94
7.1.10. Página de detalhes do usuário	95
7.1.11. Página de de sugira ou doe um livro	96
7.1.12. Página de criar comunidade	97
7.1.13. Página de comunidade	97
7.1.14. Página de relatórios	98
7.1.15. Página de acesso rápido à comunidade	99
7.1.16. Página de perfil do usuário	99
7.1.17. Página de notificação	100
7.1.18. Página de indicações semanais	100
7.1.19. Página de indicações semanais(home)	101
8. CRONOGRAMA	102
REFERÊNCIAS	104
9. ANEXOS	107

1. TEMA

Desenvolvimento e Implementação de um Sistema de Gerenciamento de Biblioteca Baseado em Tecnologias Web.

1.1. DELIMITAÇÃO DO TEMA

Com o aumento do uso de tecnologias digitais, cresce a demanda por soluções que otimizem processos administrativos e promovam uma melhoria ao usuário. As bibliotecas, como espaços de conhecimento e cultura, precisam acompanhar essa evolução tecnológica para atender às necessidades de organização, eficiência e acessibilidade.

Neste contexto, um sistema digital de gerenciamento de bibliotecas pode facilitar o controle de empréstimos, devoluções, a categorização de livros e o gerenciamento de usuários. A implementação deste sistema visa agilizar processos, reduzir a necessidade de controle manual e aumentar a segurança e acessibilidade das informações, beneficiando tanto a equipe administrativa quanto os usuários.

1.2. PROBLEMÁTICA

A biblioteca da igreja missionária unida do Brasil do bairro jardim presidencial enfrenta desafios importantes na gestão e organização de livros, empréstimos e usuários devido ao uso de métodos manuais e planilhas Excel para registro e controle. Esse modelo tradicional resulta em processos desatualizados e ineficientes, dificultando o controle dos livros emprestados, o monitoramento dos prazos de devolução e a visualização precisa dos livros disponíveis.

A ausência de um sistema digital unificado limita o acesso a informações em tempo real sobre o status dos livros e usuários, o que pode gerar confusões, atrasos e insatisfação. Além disso, a falta de categorização estruturada e funcionalidades específicas de controle de empréstimos aumenta a carga de trabalho para o administrador, expondo o sistema a riscos de inconsistências e perda de dados.

Esses problemas evidenciam a necessidade de uma solução tecnológica que proporcione um gerenciamento de biblioteca eficiente, acessível e seguro.

2. JUSTIFICATIVA

A necessidade de modernização nos processos de gerenciamento de bibliotecas se torna evidente com o aumento do acervo e da quantidade de usuários. Um sistema de gerenciamento digital fornece controle mais preciso sobre empréstimos e devoluções, melhor organização de dados e maior acessibilidade à informação, beneficiando tanto a administração da biblioteca quanto seus usuários.

3. HIPÓTESE

A implementação de uma aplicação web para gerenciamento da biblioteca, integrando controle de empréstimos, cadastro de usuários e livros, reduzirá significativamente os erros manuais nos registros e aumentará em aproximadamente 70% a eficiência do controle de empréstimos, conforme indicadores de confiabilidade dos dados e tempo médio de atendimento, quando comparados ao sistema manual atual. Dessa forma, espera-se que a equipe obtenha um controle mais seguro e ágil das operações, beneficiando a experiência dos usuários e promovendo um ambiente mais organizado.

4. OBJETIVOS

4.1. OBJETIVO GERAL

Desenvolver uma aplicação web para gerenciamento do acervo e das atividades da biblioteca, com foco em melhorar o controle de empréstimos, devoluções e organização de usuários, melhorando a eficiência administrativa.

4.2. OBJETIVOS ESPECÍFICOS

- Analisar os fluxos atuais da biblioteca e identificar pontos críticos na gestão da biblioteca.
- Identificar e priorizar as necessidades dos usuários e administradores.
- Pesquisar e selecionar as tecnologias adequadas para o desenvolvimento de um sistema web responsivo e seguro.
- Implementar funcionalidades específicas para gerenciamento do estoque de livros, cadastro de usuários e categorização de livros.

- Testar e ajustar o sistema com feedback dos usuários e administradores, garantindo uma interface intuitiva e de fácil uso.
- Desenvolver e implementar o sistema.

5. FUNDAMENTAÇÃO TEÓRICA

Visando o uso de tecnologias web como base para a proposta de intervenção, irei apresentar uma série de conceitos das tecnologias que serão utilizadas, e suas fundamentações teóricas. Dentre elas as linguagens de programação JavaScript e TypeScript. O banco de dados utilizarei o MySQL. No Back-End, será utilizado o Node.js, que será responsável pela comunicação entre a API e o banco de dados. Já no Front-End, será utilizado o framework Next.js, que é baseado na biblioteca React. A estilização da interface será feita com CSS, garantindo um design responsivo e atraente.

5.1. ANÁLISE DE SISTEMAS DE BIBLIOTECAS EXISTENTES

Para o desenvolvimento dos novos sistemas, foram realizadas algumas análises em sites de bibliotecas, a fim de identificar os padrões de boas práticas, funcionalidades relevantes e inspirações a serem incorporadas ao projeto. Foram estudadas bibliotecas cristãs e acadêmicas, como uma sugestão de inspiração pelos administradores, a Loja JesusCopy, além de outros modelos de bibliotecas cristãs.

5.1.1. Bibliotecas Cristãs e sua Relevância

Segundo Luz da sabedoria (2024), isso resume o que é uma biblioteca cristã:

A biblioteca cristã é um espaço dedicado ao armazenamento e à disseminação de literatura religiosa, especialmente voltada para a fé cristã. Este tipo de biblioteca pode incluir Bíblias, livros de teologia, literatura devocional, estudos bíblicos e obras de autores cristãos renomados. O objetivo principal é fornecer recursos que ajudem os fiéis a aprofundar seu conhecimento sobre a doutrina cristã e a viver sua fé de maneira mais informada e consciente. (LUZ DA SABEDORIA, 2024, pg 1).

Continuando, o autor fala sobre as variedades que uma biblioteca cristã pode conter, não se limitando apenas a bíblias em suas diferentes traduções, mas pode conter também livros de história da igreja, literatura missionária e obras de ficção cristã, devocionais e entre outros. Segundo Luz da Sabedoria (2024), as bibliotecas

cristãs desempenham um papel importante na formação espiritual dos indivíduos e das comunidades pois oferecem acesso a matérias que podem enriquecer a vida espiritual, promover o estudo das escrituras e incentivar a reflexão sobre questões morais e éticas à luz da fé cristã, além de promover estudos bíblicos e devocionais.

5.1.2. JesusCopy

A loja JesusCopy é um e-commerce na venda e distribuição de livros cristãos, bíblias e acessórios cristãos. Ela faz parte do movimento JesusCopy, liderada pelo pastor Douglas Gonçalves e sua esposa Val Gonçalves.

Por acreditarmos no que Jesus quer fazer na nossa geração e apostarmos na leitura como um instrumento de aperfeiçoamento, prosseguimos com o propósito de nos tornarmos todas cópias de Jesus através da propagação de Bíblias e livros que irão edificar o próximo (JESUSCOPY, 2025, pg 1).

A indicação dos administradores para analisar e ter inspirações nesse site é porque é um site moderno e intuitivo, proporcionando uma experiência de navegação agradável. A plataforma oferece recomendações personalizadas de livros, tem vários livros atuais, bíblias estilizadas, um design legal entre outras coisas.

Figura 01: Tela Inicial JesusCopy



Fonte: JESUSCOPY, 2025.

5.1.3. LivrariaS Família Cristã

As Livrarias Família Cristã é uma rede de livrarias evangélicas que teve início em 2002, fundada por uma família cristã em Marília-SP. Com o crescimento das vendas e demandas por materiais cristãos, a empresa expandiu suas operações para Londrina-PR, onde abriu a maior loja do segmento no estado do Paraná. Além das lojas físicas, a livraria lançou sua plataforma digital, permitindo a venda e distribuição de livros e produtos cristãos para todo o Brasil (FAMÍLIA CRISTÃ, 2025). A missão da livraria é:

Proclamar a palavra de Deus através do nosso trabalho para que o Brasil se encha do pleno conhecimento da Glória do SENHOR, como as águas cobrem o mar. (FAMÍLIA CRISTÃ, 2025, pg 1).

Embora a livraria tenha o maior foco comercial, a análise vai trazer insights de design, pois a plataforma é bem intuitiva e tem umas funcionalidades bem interessantes de serem implementadas, como destaque do mês, sistema como, é do seu gosto? Então veja estes produtos similares, frequentemente lidos juntos entre outros detalhes.

Figura 02: Tela inicial Livrarias Família Cristã



Fonte: LIVRARIAS FAMÍLIA CRISTÃ, 2025.

5.2. COMPARAÇÃO ENTRE O SISTEMA ATUAL E SISTEMA NOVO

A gestão eficiente de uma biblioteca exige que tudo esteja bem-organizado: empréstimos, devoluções e o acervo em geral. Só que, na biblioteca da igreja, o

sistema atual ainda depende muito de métodos manuais, como anotações em papel e planilhas de Excel, o que acaba limitando bastante as coisas e podendo gerar erros humanos e perdas de dados. A ideia de criar um sistema web de gerenciamento vem justamente para resolver isso, trazendo mais praticidade, modernizando os processos e garantindo mais segurança, agilidade e facilidade tanto para os administradores quanto aos leitores que terão mais acessibilidade aos livros.

5.2.1. Sistema antigo/atual

O sistema que a biblioteca usa hoje ainda depende muito de planilhas no Excel e de anotações feitas à mão. Isso não só gera mais trabalho considerável para os administradores, pois também aumenta as chances de erros, perda de dados e muita correria para manter tudo organizado, as informações em dia e ter precisão nas informações.

Algumas características e limitações do sistema atual incluem:

- Registro manual de empréstimos e devoluções: A administração precisa preencher informações manualmente, o que aumenta o risco de erros, como anotar o nome do livro errado, quantidade que tem estoque, esquecer de anotar quando foi emprestado o livro entre outros erros que um ser humano pode cometer. (ver Anexos 1 a 9.)
- Dificil acesso do leitor: Para o leitor saber se um livro está disponível ele precisa ir até a biblioteca e procurar o livro no acervo.
- Falta de controle automatizado: O sistema não possui mecanismos para alertas de prazos de devolução ou identificação de usuários inadimplentes, resultando em atrasos e dificuldades na recuperação de livros, levando algumas vezes até a perda do livro.

Com isso, percebe-se que o sistema manual impacta negativamente a eficiência do gerenciamento da biblioteca, gerando retrabalho e falta de confiabilidade nas informações.

Um dos pontos bons da biblioteca é a organização, os livros são separados por categorias e cada 'categoria é organizada por etiquetas de cores, facilitando assim a

localização do livro caso ele esteja no acervo da biblioteca (ver na figura 03 e anexo 10)

Figura 03: Controle de livros

1	A	B	C	D	E	F	G	H	I
2		Título	Autor	Editora	Tema	Devocional			
3	1	Milhões Transculturais - Uma perspectiva cultural	Ralph D. Winter/ Steven. Hawthorne	Mundo Cristão	Crucifixo	Visão Cristã			
4	1	8 bilhões em uma igreja contextualizada	Collin W. Dye	Renascor	Vida Cristã	Missional			
5	1	Armadaz do Coração de Deus - Ao seu Coração	Kay Arthur	Key Arthur	Devocional	Homens			
6	1	Redeiras Duras - Sabia e Firme	Mark Fisher	Case	Empresarial	Planejamento			
7	1	101 Ideias Criativas - Para Mulheres	Mary Ann Cox e Carol Sue Markh	Hagnos	Mulheres	Mulher			
8	1	101 Ideias Criativas - Para Grupos Pequenos	Wilson Soares Cardoso	AD Santos	Crucifixo	Família			
9	1	101 Ideias Criativas - Para Profissionais	Mary Ann Cox e Carol Sue Markh	Hagnos	Crucifixo	Novidades			
10	1	101 Ideias Criativas - Para Grupos Pequenos	Mary Ann Cox e Carol Sue Markh	Hagnos	Crucifixo				
11	1	101 Ideias Criativas - Para Grupos Pequenos	Antonio Tadeu Ayres	Vida	Crucifixo	Biografia			
12	1	101 Ideias Criativas - Para Grupos Pequenos	Sergio Moraes	Pratico	Crucifixo	Livros biblicos			
13	1	101 Ideias Criativas - Para Grupos Pequenos	Alfredo Halpern	Abeli	Validados	Laboracao e arte			
14	1	101 Ideias Criativas - Para Grupos Pequenos	Dong Yu Lan	Avore de Vida	Vida Cristã				
15	2	101 Ideias Criativas - Para Grupos Pequenos	José Paulo Fachini	Aborada	Crucifixo				
16	2	101 Ideias Criativas - Para Grupos Pequenos	José Paulo Fachini	Aborada	Crucifixo				
17	1	101 Ideias Criativas - Para Grupos Pequenos	John Diano	Rom Pastor	Crucifixo				
18	1	101 Ideias Criativas - Para Grupos Pequenos	Hagnos Reconhecimento	Hagnos Reconhecimento	Crucifixo				
19	1	101 Ideias Criativas - Para Grupos Pequenos	Francisco Manoel de Melo e Roseny Scheidgger de Melo	GraffNort	Validados				
20	1	101 Ideias Criativas - Para Grupos Pequenos	Joselak Lima	AD Santos	Crucifixo				
21	1	101 Ideias Criativas - Para Grupos Pequenos	R.R Soares	Graca Editorial	Crucifixo				
22	2	101 Ideias Criativas - Para Grupos Pequenos	Valdeci L. Silva Jr.	SOCEP	Vida Cristã				
23	1	101 Ideias Criativas - Para Grupos Pequenos	Francisco Manoel de Melo e Roseny Scheidgger de Melo, Luciana Shroedger de Melo	SOCEP	Vida Cristã				
24	1	101 Ideias Criativas - Para Grupos Pequenos	Autenas Diversos	Força Editorial	Crucifixo				
25	1	101 Ideias Criativas - Para Grupos Pequenos	Larry Lee	Edora Bedrina	Crucifixo				
26	1	101 Ideias Criativas - Para Grupos Pequenos	Clayton M. Bicena	Clayton M. Bicena	Crucifixo				
27	1	101 Ideias Criativas - Para Grupos Pequenos	Paris Redhead	Betânia	Vida Cristã				
28	1	101 Ideias Criativas - Para Grupos Pequenos	Case	Case	Crucifixo				
29	1	101 Ideias Criativas - Para Grupos Pequenos	Adriana Helen Borges	Gráfica Print	Mulheres				
30	1	101 Ideias Criativas - Para Grupos Pequenos	Adriana Helen Borges	Gráfica Print	Mulheres				
31	1	101 Ideias Criativas - Para Grupos Pequenos	Jobo Manoel Falzardo	Visão	Crucifixo				
32	1	101 Ideias Criativas - Para Grupos Pequenos	Maria Cecília	Kellogg	Crucifixo				
33	1	101 Ideias Criativas - Para Grupos Pequenos	Craig Hill	Universidade da Família	Validados				

Fonte: Administrador da Biblioteca

5.2.2. Sistema novo

Já o novo sistema de gerenciamento da biblioteca tem como foco trazer uma interface moderna e intuitiva, a aplicação web permitirá uma gestão eficiente do acervo, bem como o engajamento dos leitores por meio de uma comunidade virtual, onde poderão interagir, compartilhar metas de leitura e acompanhar o progresso coletivo.

Principais diferenciais do novo sistema:

1. Comunidade de Leitores

- Os usuários poderão adicionar amigos dentro do sistema e formar grupos de leitura, colaborando para atingir objetivos específicos, como ler determinado livro em conjunto, ler determinado livro em uma quantidade de dias, ler uma quantidade de livros por um período. (ver figura 04)
- Será disponibilizado um ranking, destacando os leitores mais ativos, incentivando a competitividade saudável e o engajamento na leitura.

- Gráficos de progresso serão exibidos para que os leitores acompanhem sua evolução e motivem-se a alcançar suas metas.

2. Relatórios Gerenciais para o Administrador

- A nova solução permitirá ao administrador acessar relatórios detalhados sobre:
 - Desempenho dos leitores (quantidade de livros lidos, tempo médio de leitura).
 - Quantidade de empréstimos ativos e devoluções pendentes.
 - Dados estatísticos sobre as categorias mais procuradas da biblioteca.
 - Dados estatísticos de qual gênero ler mais.
 - Dados estatísticos de qual faixa etária ler mais
- Os relatórios visam ajudar os administradores na tomada de decisão estratégica, podendo cada vez mais tomar decisões de melhorias baseada na análise dos dados.

3. Perfis de Usuários e Controle de Acesso

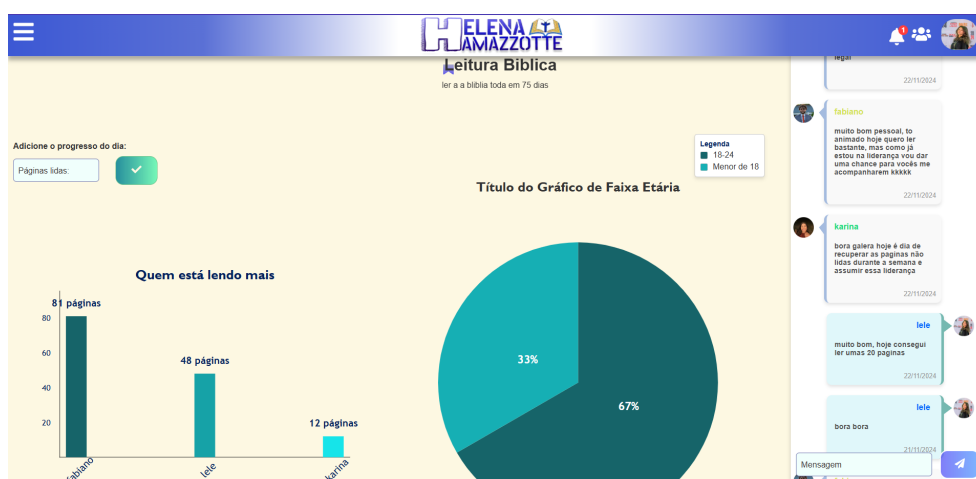
O novo sistema contará com três níveis de acesso, garantindo segurança e organização:

- **Administrador:**
 - Controle total do sistema.
 - Gestão dos voluntários e leitores.
- **Voluntário:**
 - Acesso um pouco mais limitado.
 - Ajudará nos empréstimos e devoluções de livros.
 - Não terá permissão para excluir ou cadastrar novos livros e não poderá visualizar relatórios administrativos.
- **Leitor:**
 - Acesso ao catálogo de livros disponíveis para consulta e solicitação de empréstimos.
 - Acessos às comunidades, podendo interagir com amigos, acompanhando desafios e rankings.
 - Visualização do próprio histórico de leitura e progresso em desafios.

4. Automação e Segurança de Dados

- Notificações automáticas para prazos de devolução, não apenas para o usuário como para o administrador, pois assim ele poderá tomar as cabíveis atitudes de procurar o leitor e entender o motivo da não devolução.
- Registro de logs de atividades para rastrear ações importantes e prevenir acessos indevidos.

Figura 04: Tela de comunidade do novo sistema



Fonte: Próprio Autor

Com essas melhorias tende a tornar-se o sistema mais moderno, acessível e engajador, facilitando o controle administrativo da biblioteca e beneficiando não apenas os administradores, mas também os leitores, que terão uma experiência interativa e motivadora com seus amigos.

5.3. LINGUAGENS DE PROGRAMAÇÃO

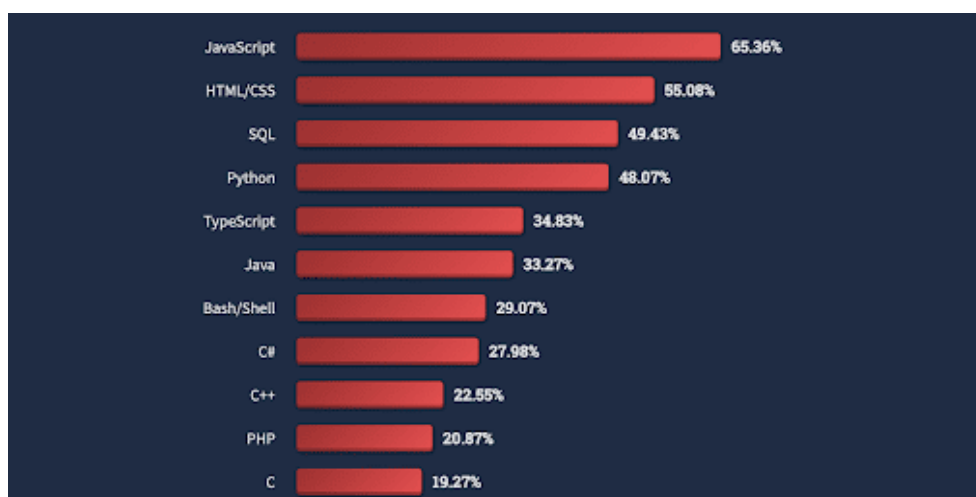
Santana (2024) menciona que linguagens de programação nada mais são do que um idioma com regras escritas que garante essa comunicação entre ser humano e o computador, mas o computador não consegue se adaptar tão facilmente como um ser humano que se adapta a vários idiomas, e é por isso que são criadas várias linguagens. A programação se resume a um conjunto de instruções passadas ao

computador para que ele execute uma determinada tarefa, desde algo simples até a criação de aplicativos complexos.

Aprofundando no assunto, as linguagens de programação podem ser classificadas em baixo nível e alto nível. As linguagens de de baixo nível, como o assembly, se trata das primeiras que foram criadas, são conhecidas por sua sintaxe ser mais próxima da linguagem de máquina, o que faz sentido pois seu propósito é que seja mais perceptível a máquina, seu uso ainda é importante na atualidade por ser a responsável por programar hardwares dos mais variados usos, com exemplos sistemas embarcados e microcontroladores. Apesar de sua complexidade, elas proporcionam alto desempenho e eficiência, fatores essenciais para aplicações críticas (SOUZA; FLORIAN; DALLILO, 2023).

Por outro lado, podemos citar as linguagens de alto nível que são as mais atuais, e se caracterizam por sua semelhança com a nossa linguagem humana, desse modo facilita a escrita, manutenção e legibilidade do código, sua sintaxe carrega variáveis, controle de fluxo, módulos, funções e paradigmas de programação como orientado a objetos, estruturado, funcional. Segundo Santana (2024) as linguagens de alto nível que mais se destacam e são utilizadas é o python, javascript, java, typescript entre outras que são amplamente utilizadas no desenvolvimento de aplicações web, inteligências artificiais e softwares empresariais.

Figura 05: Ranking linguagens mais utilizadas



Fonte: Santana, 2024

5.3.1. Paradigma de Programação

Segundo Sebesta (2018) paradigmas são modelos ou moldes que são usados para estruturar um problema que pode ser representado por dados, variáveis e instruções. Na resolução desses problemas pode ser utilizado diferentes paradigmas como estruturado, funcional e orientado a objetos.

Logo de início temos o paradigma estruturado que teve sua criação na década de 60 e tem como principal característica a organização do código por meio de estrutura de controle, funções e sub-rotinas, com isso modulariza a solução do problema, facilitando sua compreensão e manutenção, mas por outro lado essa separação entre funções e dados pode gerar mais trabalho na manutenção, pois uma simples mudança nos dados exige alterações em diversas partes do código. Segundo Souza, Florian e Dallilio (2023) esse paradigma é amplamente utilizado em linguagens como C, Pascal e Cobol.

Em seguida temos o paradigma funcional que como o nome já diz funcional se baseia na composição de funções matemáticas que não vão modificar o estado global dos sistemas, com isso tende a reduzir os efeitos colaterais e ainda facilitam os testes de código. Essas funções no paradigma funcional são as classes que podem ser passadas como parâmetros para outras funções e retornadas como resultado, segundo Farroco et al. (2020) essa abordagem é comum em linguagens como o haskell, lisp e erlang pois são sistemas que exigem o paralelismo e alto desempenho

E por fim temos o paradigma orientado a objeto que surgiu para facilitar todos os problemas citados anteriormente por meio de objetos que tem como objetivo encapsular atributos (dados) e métodos (comportamentos), em sua sintaxe tem conceitos como herança, polimorfismo e encapsulamento que permite maior reutilização de código. Segundo Jungthon e Goulart (2020), as linguagens que mais utilizam a orientação a objeto é o java, C++, python, typescript.

5.3.2. Javascript

O javascript também conhecido como js é uma linguagem de programação de alto nível interpretada, multiparadigma e dinâmica, muito utilizada no desenvolvimento

web para deixar páginas mais animadas e interativas e o que faz ser uma linguagem muito utilizada é o fato de pode ser utilizado tanto no front quanto back. O js foi criado em 1995 por Brendan Eich, originalmente criado para ser executado diretamente no navegador do usuário sendo assim uma linguagem *client-side*, ou seja, processada pelo computador do cliente em vez de no servidor (JAISON, 2012). Inicialmente seu nome era *livescript* mas uma parceria com a Sun Microsystem, empresa responsável pelo java, o nome *livescript* foi alterado para javascript, isso foi uma jogada de marketing feita para chamar a atenção devido a fama do java na época (PEREIRA; SILVÉRIO, 2024).

Uma das principais funções do javascript no ambiente web é possibilitar a interação entre o usuário e a interface do site, manipulando elementos HTML e do DOM, isso permite validações de usuário, animações, requisições assíncronas entre outras diversas funcionalidades que deixam as aplicações web mais dinâmicas. Segundo Pereira e Silverio (2024) o que torna o javascript muito utilizado é sua versatilidade que não se limita apenas ao front, pode ser utilizada no back com lado do servidor (*Server-Side*) juntamente com o node.js, pode ser utilizada tanto em mobiles e sistemas embarcados também.

Segundo Jaison (2012) um dos maiores problemas na época é sua interpretação pelos navegadores, pois cada navegador pode processar e rodar o código de maneira diferente o que pode levar a inconsistências. Para minimizar esses problemas em 2015 foi criado o padrão ECMAScript mais conhecido com ES6 que define as diretrizes da linguagem e desde então o javascript passou por grandes evoluções, incorporando recursos como *arrow functions*, *classes*, *async/await* e *destructuring*, garantindo assim maior compatibilidade e eficiências entre os diversos e diferentes navegadores (MDN, 2023).

Como citado acima, o javascript tem muita versatilidade e pode ser utilizado em diversas áreas da tecnologia, além de ter uma variedade de frameworks e bibliotecas o que torna ele uma linguagem muito popular e utilizada mundialmente. Segundo Pesquisas do *Stack Overflow* (2022, pg 1) em sua pesquisa anual mostra que atualmente o javascript é uma das linguagens mais utilizadas em todo o mundo.

5.3.3. Typescript

O typescript é uma linguagem de programação desenvolvida pela microsoft em 2012 como uma extensão para melhorar algumas falhas e vulnerabilidades que tem no javascript, ele oferece alguns recursos avançados como tipagem estática, interfaces, classes e suporte a linguagem aprimorada à programação orientada a objetos (MICROSOFT, 2019).

Segundo Neves (2024), o *typescript* é um subconjunto do *javascript*, ou seja, todo código javascript válido também pode ser executado em *typescript*, mas uma grande vantagem do *typescript* no seu sistema de tipos estáticos que nos permite detectar erros enquanto "codamos", falhas que antes só surgia na hora que rodava o código com essa melhoria de ver os erros na própria IDE ajuda muito em projetos robustos tornando assim mais fácil a manutenibilidade e qualidade do software.

Outro diferencial do typescript é seu suporte avançado à orientação a objetos que oferece recursos como controle de acesso (público, privado, protegido), interfaces e herança, essas características que auxiliam na modulação do código e na reutilização do código e na reutilização de componentes (HILMAN, 2024).

Por ser transpilado para javascript puro o typescript é compatível com qualquer ambiente que suporte o javascript tornando assim sua adoção flexível em diferentes cenários. Ele é muito utilizado no front com os frameworks react, angular e vue.js mas também pode ser utilizado no backend com o node.js. Como vimos na figura 03 o typescript já ocupa os top 10 linguagens mais utilizadas e vem cada vez mais se tornando popular.

5.4. DESENVOLVIMENTO WEB

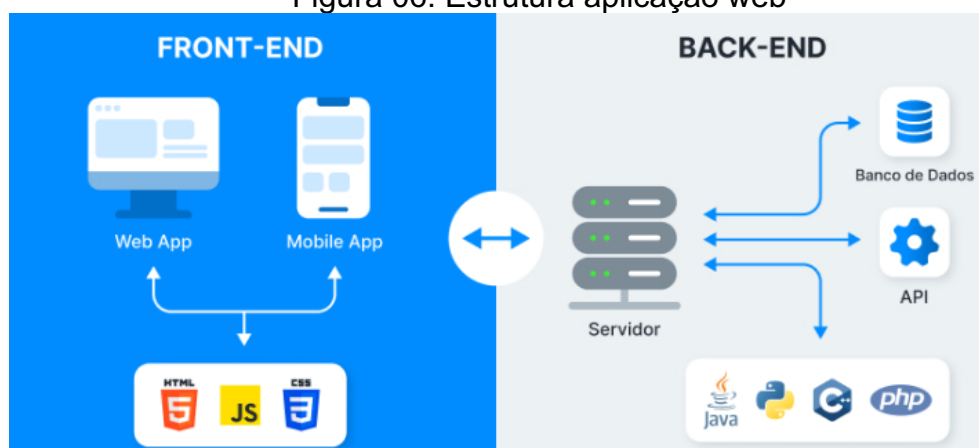
Segundo Valença (2023), o desenvolvimento web pode ser definido como a construção de software baseado na internet, e essa aplicação é dividida principalmente entre o front-end e back-end. O front-end é um termo utilizado para referir ao desenvolvimento de interfaces para aplicações web. Essas interfaces visuais, o usuário final vai ter contato direto com elas, geralmente elas podem conter, textos, botões, caixa de diálogo, menus, ícones, barras de navegação, entre outras

coisas, para montar essa estrutura, estilizar, adicionar interatividade às páginas, criar animações e entre outras coisas é utilizado as tecnologias como HTML, CSS e javascript. A outra parte se trata do Back-End que trabalha numa camada que se demonstra invisível ao cliente, porém é primordial para a lógica de funcionamento da aplicação, para o gerenciamento dos dados, e controlador do fluxo de informações, para isso é utilizado linguagens como python, PHP, ruby, java e javascript(node.js) (MENEZES, 2024).

Dentro desse universo temos várias ferramentas que são alicerces para acelerar o desenvolvimento que são os frameworks e bibliotecas, os mais conhecidos e utilizados são o react, angular, vue.js, django e laravel que oferecem componentes e estruturas pré-definidas. Além disso, temos o banco de dados que tem o papel fundamental de armazenar e recuperar eficiente das informações processadas pelas aplicações, os mais conhecidos são o MySQL, PostgreSQL e MongoDB (MENEZES, 2024)

Aprofundando no assunto vemos o avanço da tecnologia e o crescimento da demanda por soluções digitais e com isso vem o uso de novas práticas, como arquitetura de microsserviços, Progressive Web Apps (PWAs) e APIs RESTful, que permite o desenvolvimento de sistemas mais escaláveis e de alta performance. Segundo Valença (2023), os profissionais da área precisam estar sempre atualizados e estudando para lidar com a rápida evolução da tecnologia e conseguir atender o mercado de forma eficiente.

Figura 06: Estrutura aplicação web



Fonte:UDS, 2024

5.4.1. Back-end

De acordo com Santana (2024) o back-end é a parte escondida do usuário, ele não apenas gerencia a lógica de negócio e a comunicação entre servidor e cliente, mas vai muito além disso, envolvendo tecnologias que vão garantir a escalabilidade, segurança e eficiência das aplicações web, temos também ferramentas para otimizar o desempenho e a organização dos sistemas que são os sistemas de cache e arquiteturas em microsserviços.

Se aprofundando nesse aspecto da segurança no back-end podemos implementar autenticação e autorizações de cada usuário, para esse controle de acesso temos o uso de tokens como JWT e OAuth, proteção contra ataques SQL Injection e Cross-Site (XSS). Já para uma boa distribuição e escalabilidade da aplicação na nuvem podemos utilizar containers (Docker) e a orquestração de servidores (Kubernetes) (SANTANA, 2024).

5.4.1.1. Node.js

Segundo a documentação oficial

Node.js® é um ambiente de execução de JavaScript disponível para várias plataformas, de código aberto e gratuita, que permite aos programadores criar servidores, aplicações da Web, ferramentas de linha de comando e programas de automação de tarefas(NODE.JS, 2024, pg 1).

Criado em 2009 por Ryan Dahl, o node.js utiliza o motor v8 do google chrome para interpretação javascript no servidor.

Segundo a MDN (2023, pg 1), do ponto de vista do desenvolvimento de um servidor web, o node.js possui vários benefícios:

- Performance excelente. Node foi projetado para otimizar a taxa de transferência e a escalabilidade em aplicações web. É uma ótima combinação para resolver muitos problemas comuns no desenvolvimento da web (por exemplo, aplicações em tempo real).
- O código é escrito em "JavaScript simples e antigo". Isso significa menos tempo gasto para lidar com mudanças de código entre navegador e servidor web, não sendo necessária uma mudança na linguagem.
- JavaScript é uma linguagem de programação relativamente nova e apresenta algumas vantagens quando comparadas a outras linguagens tradicionais de servidor (por exemplo Python, PHP, etc.). Muitas outras linguagens novas e populares compilam/convertem em JavaScript,

permitindo que você também use essas linguagens, como TypeScript, CoffeeScript, ClosureScript, Scala, LiveScript, etc.

- O Gerenciador de Pacotes do Node (NPM, na sigla em inglês) provê acesso a centenas de milhares de pacotes reutilizáveis. NPM possui a melhor coleção de dependências e pode ser usado para automatizar a maior parte da cadeia de ferramentas de compilação.
- É portátil, com versões para diferentes sistemas operacionais, como Microsoft Windows, OS X, Linux, Solaris, FreeBSD, OpenBSD, WebOS e NonStop. Além disso, tem excelente suporte de muitos provedores de hospedagem na web, que muitas vezes fornecem documentação e infraestrutura específica para hospedar sites desenvolvidos em Node.
- Possui uma comunidade de desenvolvedores e um ecossistema muito ativo, com muitas pessoas dispostas a ajudar.

Você pode utilizar o Node.js para criar um simples servidor web, utilizando o pacote node HTTP (MDN, 2023).

Segundo Clemente (2023), em resumo, o node.js mudou a forma como construímos aplicações web, graças a sua eficiência, escalabilidade e versatilidade, o node.js se estabeleceu como uma boa escolha confiável para desenvolvedores web em todo o mundo.

5.4.1.2. APIs RESTful

Segundo a MDN (2023), REST(Representational State Transfer) refere-se a um grupo de restrições de design dentro da arquitetura de software que gera sistemas distribuídos eficientes, confiáveis e escaláveis. Um sistema é denominado RESTful quando você aceita todas essas restrições.

Segundo a documentação da AWS (2024, pg 1)

A API RESTful é uma interface que dois sistemas de computador usam para trocar informações de forma segura pela internet. A maioria das aplicações de negócios precisa se comunicar com outras aplicações internas e de terceiros para executar várias tarefas. Por exemplo, para gerar contracheques mensais, seu sistema interno de contas precisa compartilhar dados com o sistema bancário de seu cliente a fim de automatizar o faturamento e se comunicar com uma aplicação interna de planilha de horas. As APIs RESTful suportam essa troca de informações porque seguem padrões de comunicação de software seguros, confiáveis e eficientes.

As API RESTful incluem os seguintes benefícios:

Escalabilidade

Os sistemas que implementam APIs REST podem ser escalados com eficiência, porque a REST otimiza as interações entre cliente e servidor. A

ausência de estado remove a carga do servidor, porque o servidor não precisa reter informações de solicitações anteriores do cliente. O cache bem gerenciado elimina parcial ou completamente algumas interações entre cliente e servidor. Todos esses recursos permitem a escalabilidade sem causar gargalos de comunicação que reduzem a performance.

Flexibilidade

Os serviços da Web RESTful permitem a separação total do cliente do servidor. Eles simplificam e desacoplam vários componentes do servidor para que cada parte possa evoluir independentemente. Mudanças de plataforma ou tecnologia na aplicação do servidor não afetam a aplicação do cliente. A capacidade de camadas de funções de aplicações aumenta ainda mais a flexibilidade. Por exemplo, os desenvolvedores podem fazer alterações na camada de banco de dados sem reescrever a lógica da aplicação.

Independência

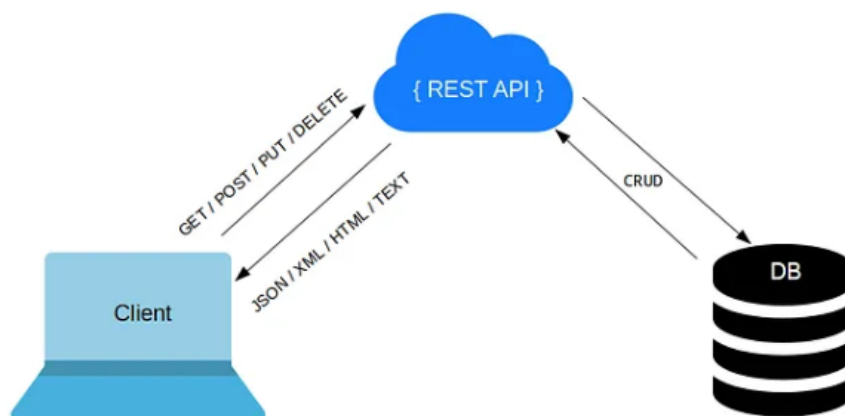
APIs REST são independentes da tecnologia usada. Você pode escrever aplicações de cliente e servidor em várias linguagens de programação sem afetar o design da API. Também é possível alterar a tecnologia subjacente em ambos os lados sem afetar a comunicação. (AWS, 2024, pg 1)

Segundo a AWS (2024), o cliente entra em contato com o servidor usando a API para fazer uma requisição, essa solicitação tem um identificador de recurso exclusivo, método (GET, POST, PUT, DELETE), cabeçalho HTTP, dados e parâmetros. Estas são as etapas gerais para qualquer chamada API REST:

1. O cliente envia uma solicitação ao servidor. O cliente segue a documentação da API para formatar a solicitação de modo que o servidor entenda.
2. O servidor autentica o cliente e confirma que o cliente tem o direito de fazer essa solicitação.
3. O servidor recebe a solicitação e a processa internamente.
4. O servidor retorna uma resposta ao cliente. A resposta contém informações que indicam ao cliente se a solicitação foi bem-sucedida. A resposta também inclui informações solicitadas pelo cliente.

Os detalhes de solicitação e resposta da API REST variam um pouco, dependendo de como os desenvolvedores da API projetam a API. (AWS, 2024. pg 1)

Figura 07: Api rest



Fonte: Medium, 2020

5.4.1.3. Gerenciamento de autenticação com JWT (JSON Web Token)

Segundo a documentação oficial do JWT (2024, pg 1) essa é definição do que é o JSON Web Token:

JSON Web Token (JWT) é um padrão aberto (RFC 7519) que define uma maneira compacta e autocontida de transmitir informações com segurança entre as partes como um objeto JSON. Essas informações podem ser verificadas e confiáveis porque são assinadas digitalmente. JWTs podem ser assinados usando um segredo (com o algoritmo HMAC) ou um par de chaves pública/privada usando RSA ou ECDSA.

Embora os JWTs possam ser criptografados para também fornecer sigilo entre as partes, vamos nos concentrar em tokens assinados. Os tokens assinados podem verificar a integridade das declarações contidas neles, enquanto os tokens criptografados ocultam essas declarações de outras partes. Quando os tokens são assinados usando pares de chaves pública/privada, a assinatura também certifica que apenas a parte que detém a chave privada é a que a assinou.

Conforme está na documentação do JWT (2024, pg1), um token é composto por 3 partes: Header (cabeçalho), payload (carga útil), signature (assinatura). O cabeçalho normalmente consiste em duas partes: o tipo do token, que é JWT, e o algoritmo de assinatura usada, por exemplo:

Figura 08: Exemplo cabeçalho.

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Fonte: JWT, 2024.

Já a caga útil armazena as informações do usuário, como ID, nome e permissões de acesso. existem 3 tipos de payload:

- **Reivindicações registradas:** São um conjunto de reivindicações predefinidas que não são obrigatórias, mas recomendadas, para fornecer um conjunto de reivindicações úteis e interoperáveis. Algumas delas são: iss (emissor), exp (tempo de expiração), sub (assunto), aud (público) e outras.
- **Reivindicações públicas:** elas podem ser definidas à vontade por aqueles que usam JWTs. Mas para evitar colisões, elas devem ser definidas no IANA JSON Web Token Registry ou definidas como um URI que contém um namespace resistente a colisões.
- **Reivindicações privadas:** são reivindicações personalizadas criadas para compartilhar informações entre partes que concordam em usá-las e não são reivindicações registradas ou públicas (JWT, 2024, pg 1).

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "admin": true  
}
```

Fonte: JWT, 2024

Já a assinatura é gerada a partir de uma chave secreta ou de um par de chaves públicas/privadas, garantindo a integridade e proteção do token.

Figura 10: Exemplo Assinatura

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  secret)
```

Fonte: JWT, 2024

Por fim junta tudo e gera uma saída com três string Base64-URL separadas por pontos que podem ser facilmente passadas em ambientes HTML e HTTP.

Figura 0: Exemplo JWT

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaXNTb2NpYWwiOnRydWV9.4pcPyMD09o1PSyXnrXCjTwXyr4BsezdI1AVTmud2fU4
```

Fonte: JWT, 2024.

Figura 11: Exemplo JWT decodificada.

The screenshot shows the JWT.io decoder interface. At the top, there is a navigation bar with the JWT logo, links for 'Debugger', 'Libraries', 'Introduction', and 'Ask', and a 'Crafted by Auth0 by Okta' badge. Below the navigation bar, the 'Algorithm' is set to 'HS256'. The main content is divided into two columns: 'Encoded' and 'Decoded'. The 'Encoded' column displays the JWT token: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaXNTb2NpYWwiOnRydWV9.4pcPyMD09o1PSyXnrXCjTwXyr4BsezdI1AVTmud2fU4`. The 'Decoded' column shows the decoded structure:

```

HEADER:
{
  "alg": "HS256",
  "typ": "JWT"
}

PAYLOAD:
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}

VERIFY SIGNATURE
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
)  secret base64 encoded
  
```

At the bottom left, there is a 'Signature Verified' status with a checkmark icon. At the bottom right, there is a blue button labeled 'SHARE JWT'.

Fonte: JWT, 2024

5.4.2. Front-end

Segundo Aquino (2024) o front-end é a parte onde o usuário final vê e interage:

É a área que se concentra na criação das interfaces visuais e interativas de um site ou aplicação, aquelas com as quais os usuários interagem diretamente. Em outras palavras, é o processo de desenvolvimento de tudo o que um usuário vê e utiliza em uma página da web, incluindo layouts, botões, menus de navegação, animações, e outros elementos visuais e funcionais.

O objetivo do desenvolvimento front-end é garantir que a interface seja atraente, intuitiva, rápida e responsiva. Isso é alcançado por meio de uma combinação de linguagens de programação como HTML, CSS e JavaScript, além do uso de frameworks e bibliotecas que facilitam a criação de aplicações interativas e dinâmicas. (AQUINO, 2024, pg 1)

5.4.2.1. HTML

O HTML (HyperText Markup) é uma linguagem de marcação padrão para a criação de páginas web. O HTML existe desde 1991 e atualmente está na versão 5, com várias funcionalidades que tem como objetivo principal demarcar a estrutura de uma página web, essa estrutura é formada por um conjunto de elementos(tags).

As tags são formadas por uma estrutura própria, iniciam com o sinal “menor de que””, em seguida vem o nome daquele elemento e por fim, o sinal “maior que”.

Podem ser dispostas em tags que precisam de fechamento e tags que fecham sozinhas (self-closing).

O fechamento de uma tag será definido com uma barra (/), sendo que no caso das tags de autofechamento, não há necessidade da presença desse caractere. (MOURA; HENRIQUE, 2024, pg1).

Aprofundando no assunto, temos uma estrutura básica para que os navegadores intérprete e renderiza o conteúdo de forma correta:

<!DOCTYPE html>

Informa ao navegador que esse documento é do tipo HTML e indica sua versão. Quando está escrito apenas html, indica que é a mais recente.

<html>

Representa a raiz do documento, serve com um container que engloba todos os outros elementos HTML.

<body>

É onde fica todo o conteúdo de texto, imagem e vídeos, em que o usuário ou usuária vê e interage, nele os conteúdos são estruturados pelas demais tags do HTML.

<script>

Esse elemento contém instruções de script ou aponta para um arquivo de script externo por meio do atributo src.

<head>

Compreende as informações do documento que serão interpretadas pelo navegador (metadados). Como, por exemplo, título do documento, links para folhas de estilo etc.

<meta>

Define metadados, ou seja, informações sobre dados de um documento HTML. As tags vão dentro do elemento e são usadas para especificar o conjunto de caracteres, o autor ou autora do documento, as configurações da janela de visualização etc.

<link>

É uma tag vazia, que contém apenas atributos e faz a relação do documento HTML com recursos externos.

É comumente usado para vincular uma folha de estilo externa, também é usada para definir o favicon da página (ícone da aba do navegador), como outros recursos.

<style>

Essa tag é usada para declarar estilos (CSS) para um documento. (MOURA; HENRIQUE, 2024, pg 1)

Figura 12: Exemplo HTML estrutura certa

```
● ● ●  
  
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta http-equiv="X-UA-Compatible" content="IE=edge">  
  <meta name="viewport" content="width=device-width, initial  
  <title>Document</title>  
</head>  
<body>  
  
</body>  
</html>
```

Fonte: ALURA, 2024

E por fim temos as tags semânticas e tags sem semântica. Segundo Moura e Henrique (2024), as tags semânticas são tags que possuem significados e que vão dar sentido a informação de texto ao navegador e buscadores, usar tags semânticas é uma boa prática para ajudar no entendimento do código, ajuda na otimização do site, além de ranquear melhor nos buscadores como o Google. Lista das tags semânticas:

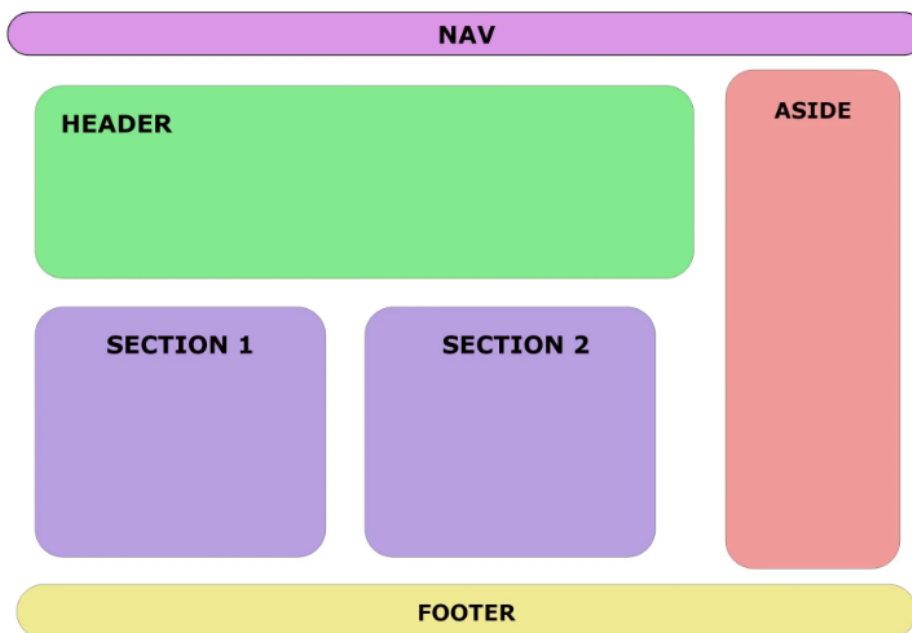
- <header>
- <main>
- <footer>
- <section>
- <article>
- <aside>
- <nav>

- ``
- ``
- ``

Já as tags sem semântica não definem um significado para aquele texto, normalmente são utilizadas apenas para fins de estilização e separação. Lista de tags sem semântica:

- `<div>`
- ``
- ``
- `<i>`

Figura 13: Exemplo de estruturação com tags semânticas:



Fonte: ALURA, 2024

5.4.2.2. CSS

O CSS (Cascading Style Sheets) é uma linguagem de folhas de estilos utilizada para dar estilo a página web. Com o CSS conseguimos criar designs responsivos, garantindo que a interface se adapte a diferentes dispositivos e tamanhos de telas. Segundo a documentação MDN (2024), o CSS vai funcionar em conjunto com o HTML, aplicando estilos aos elementos(tags), ou seja, vamos deixar a página com um visual agradável ao usuário final, colocando: cores, fontes, espaçamentos entre parágrafos, caixas de textos, tabelas, variações de layouts, ajustes de imagens, animações entre outras coisas.

5.4.2.3. Next.js e React

Segundo a documentação oficial do next.js:

Next.js é um framework React para construir aplicativos web full-stack. Você usa React Components para construir interfaces de usuário, e Next.js para recursos e otimizações adicionais.

Por baixo dos panos, o Next.js também abstrai e configura automaticamente as ferramentas necessárias para o React, como empacotamento, compilação e muito mais. Isso permite que você se concentre em construir seu aplicativo em vez de gastar tempo com configurações (VERCEL,2025, pg 1).

Segundo Clemente (2024), um dos principais benefícios do Next.js é o SSR (Server-Side Rendering), o React é uma SPA (Single Page Application), ou seja, todo o conteúdo é gerado no navegador do usuário, já o SSR permite que o conteúdo seja renderizado no servidor e entregue para o usuário.

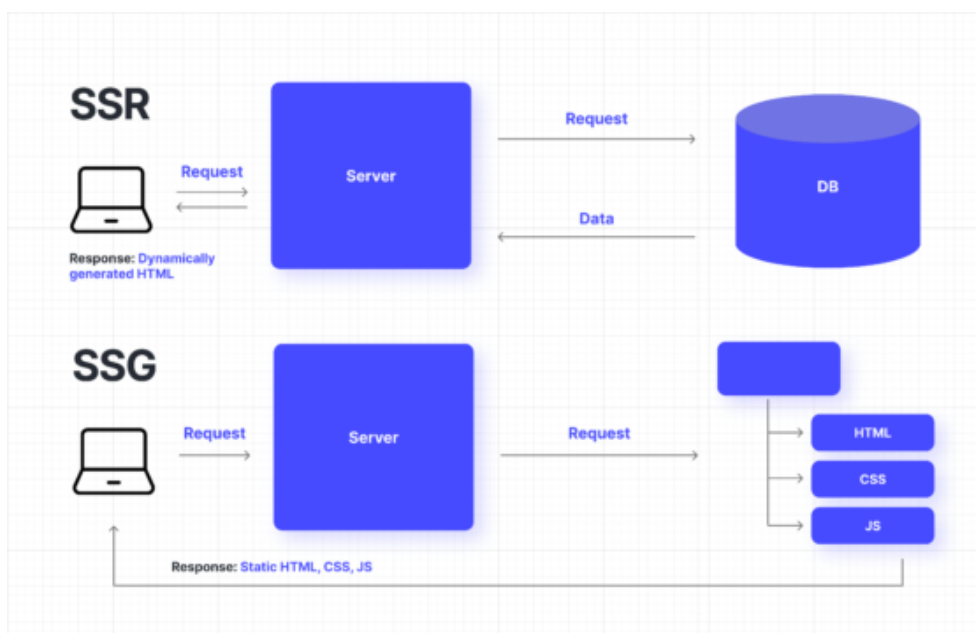
O Next.js com o sistema SSR veio para solucionar o problema que é quando os desenvolvedores colocam muito código javascript no lado do cliente, isso pesa na renderização. Segundo Nascimento (2022, pg 1) esse o funcionamento do SSR:

O SSR funciona da seguinte maneira, ao acessarmos um site por exemplo: alura.com.br enviamos uma requisição para o servidor para acessar a página inicial do site, então o servidor manda uma requisição para uma rest api por exemplo pedindo essa página inicial. Uma vez que os dados são recebidos o servidor renderiza uma página HTML com o conteúdo e envia para o cliente exibindo assim o conteúdo na página, se a pessoa acessar outra rota por exemplo: alura.com.br/artigos o processo continua o mesmo.

Também temos o sistema SSG (Static Site Generation), que segundo a documentação oficial, VERCEL (2024, pg 1):

Se uma página usa Static Generation, o HTML da página é gerado no momento da compilação. Isso significa que, na produção, o HTML da página é gerado quando você executa next build. Esse HTML será então reutilizado em cada solicitação. Ele pode ser armazenado em cache por um CDN.

Figura 14: SSR e SSG



Fonte: MEDIUM, 2024

5.5. Banco de dados

Segundo a documentação oficial da ORACLE (2025) essa é definição de banco de dados:

Um banco de dados é uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (DBMS). Juntos, os dados e o DBMS, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados.

Os dados nos tipos mais comuns de bancos de dados em operação atualmente são modelados em linhas e colunas em uma série de tabelas para tornar o processamento e a consulta de dados eficientes. Os dados podem ser facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados. A maioria dos bancos de dados usa a linguagem de consulta estruturada (SQL) para escrever e consultar dados.

O SQL (Structured Query Language, Linguagem de consulta estruturada) é uma linguagem de programação usada por quase todos os bancos de dados, com ela

conseguimos consultar, manipular e definir dados e fornecer controle de acesso. Desde sua criação na década de 1960 até hoje os bancos de dados passaram por algumas evoluções e hoje em dia temos diversos tipos de banco de dados, como: banco de dados relacionais, banco de dados NoSQL, data warehouse, banco de dados em nuvem, entre outros tipos.

5.5.1. MySQL

Segundo a documentação oficial da ORACLE (2020, pg 1) essa é definição do banco de dados MySQL:

MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto baseado em SQL. Ele foi projetado e otimizado para aplicativos da web e pode ser executado em qualquer plataforma. Como surgiram requisitos novos e diferentes com a internet, o MySQL tornou-se a plataforma preferida para desenvolvedores da web e aplicativos baseados na web. Como foi projetado para processar milhões de consultas e milhares de transações, o MySQL é uma escolha popular para empresas de comércio eletrônico que precisam gerenciar várias transferências de dinheiro. A flexibilidade sob demanda é o principal recurso do MySQL.

Hoje em dia o MySQL é altamente usado por seus benefícios como: facilidade de uso e desempenho, confiabilidade e segurança, licença de código aberto, entre outros benefícios. Essa é a história de evolução do MySQL:

A primeira versão do MySQL Server foi lançada em 1995 pela empresa sueca MySQL AB, fundada por David Axmark, Allan Larsson e Michael Widenius. O MySQL leva o nome da filha de Widenius, chamada My. O projeto MySQL foi lançado como código aberto em 2000, sob a GNU General Public License (GPL). Em 2001, o MySQL havia atingido mais de 2 milhões de instalações ativas; em 2004, o software estava sendo baixado mais de 30.000 vezes por dia. O MySQL foi adquirido pela Sun Microsystems em 2008 e, quando a Oracle adquiriu a Sun em 2009, também assumiu a propriedade do MySQL. Hoje, o MySQL é o sistema de banco de dados relacional de código aberto mais amplamente usado. (AWS, 2024, pg 1)

5.6. Metodologias ágeis

Segundo Bastos (2023), as metodologias ágeis são técnicas de aceleração de entregas em projetos, elas auxiliam na gestão e organização de demandas com base em cronogramas e fácil visualização de etapas e na divisão de tarefas e são muito utilizadas na área da tecnologia, atualmente existem inúmeras metodologias ágeis, como: Kanban, Lean, Scrum, Extreme Programming (XP), Test Driven Development

(TDD), Feature Driven Development (FDD), entre outras, cada uma com diferentes abordagens, a fim de atender às mais diversas demandas e tipos de projetos.

Apesar de ter várias metodologias ágeis, elas apresentam de um modo geral características em comum como: Melhoria de performance na empresa, Aumento da satisfação do público, por proporcionar uma boa experiência e uma maior entrega de valor, Redução de erros, Melhoria contínua, Foco em resultados (BASTOS, 2023)

5.6.1. Scrum

O Scrum é uma das metodologias mais populares pois foca em agilidade e entrega, além de promover transparência, supervisão e adaptação, permitindo que as equipes entreguem continuamente e façam ajustes rápidos caso necessário. É assim que o scrum funciona:

O conceito divide os projetos em ciclos curtos chamados de sprints, que duram de duas a quatro semanas, e envolvem quatro cerimônias principais:

- sprint planning (reunião de planejamento): momento em que o responsável pelo produto apresenta os requisitos desejados para o projeto e a equipe define tarefas;
- daily meetings (reuniões diárias): reuniões rápidas para acompanhar o desenvolvimento das atividades e possíveis dificuldades;
- sprint review (reunião de revisão): demonstração de resultados ao responsável pelo produto e feedbacks para validar a entrega;
- sprint retrospective (reunião de retrospectiva): análise sobre o desempenho da equipe e os resultados do projeto.

Cada sprint tem uma duração específica e um conjunto de objetivos a serem alcançados. As entregas realizadas ao final de cada ciclo visam agregar valor ao negócio (CNN, 2023, pg 1).

A metodologia é pautada em série de princípios, que servem como um guia para a implementação, mas podendo ser adaptado conforme o projeto. Principais princípios:

- foco no valor: o time deve priorizar atividades que geram mais valor para o cliente, conforme o backlog do produto;
- adaptação: capacidade de adaptar o projeto de acordo com as mudanças no processo e necessidades do cliente;
- transparência: as informações do projeto, como escopo, prioridades e andamento, devem ser acessíveis a todos os envolvidos no processo;
- inspeção: o trabalho realizado deve ser acompanhado, verificado e avaliado ao longo do ciclo, em reuniões diárias, de revisão e retrospectiva;
- colaboração: o trabalho da equipe deve ser colaborativo e integrado, com compartilhamento de conhecimento, soluções e responsabilidades;

- autonomia: a equipe deve ter autonomia para organizar as tarefas e definir como executar cada uma delas, de acordo com orientações do líder e do responsável pelo produto. (CNN,2023, pg 1)

Ainda Segundo a CNN (2023, pg 1), o Scrum possui alguns artefatos que são:

- Product backlog:

Product backlog, ou backlog do produto, é uma lista de funcionalidades a serem desenvolvidas durante os sprints, normalmente definida na reunião de planejamento, realizada com todos os envolvidos no processo.

Essa lista é organizada por prioridades de acordo com o objetivo e as necessidades do projeto.

- Sprint backlog:

Essa é uma subcategoria do product backlog, que tem como função definir a lista de funcionalidades específicas de cada sprint, bem como fazer a distribuição de funções e responsabilidades.

A determinação acontece também na reunião de planejamento e pode ser avaliada durante as sprint reviews, que ocorrem ao final de cada sprint e servem para analisar erros, acertos e pontos de melhoria para próximos projetos.

- Incremento:

O incremento, também conhecido como meta de sprint, é a entrega originada de um ciclo. É a versão utilizável final do produto apresentada ao cliente, que será avaliada e receberá um feedback.

Ele é uma maneira de agregar valor ao produto e pode ser adicionado de acordo com a conclusão dos itens do backlog. São detalhes que podem ser integrados ao produto, contribuindo para a experiência do consumidor.

Outro artefato citado pelo artigo é o gráfico burndown, ou gráfico de queima, que acompanha o andamento de cada sprint. Ele é o responsável por indicar quanto trabalho foi realizado e o que ainda precisa ser feito no projeto.

- Membros de uma equipe do Scrum:

A estrutura dessa metodologia ágil também conta com um grupo de profissionais, responsáveis por executar as sprints.

Essa equipe se reúne durante os eventos do método, como a reunião de planejamento e as reuniões diárias, com o objetivo de alinhar as metas, acompanhar o progresso das ações e fazer adaptações quando necessário.

Nesse cenário, o estudo da UFCG destaca três papéis da estrutura.

- Responsável pelo produto:

O product owner, ou responsável pelo produto, tem a função de definir e priorizar as funcionalidades do produto, representando as necessidades dos clientes e dos stakeholders.

- Líder Scrum:

O líder, também chamado de scrum master, é o facilitador e guardião do processo. Ele é o responsável por ajudar a equipe a seguir os princípios e as práticas da metodologia, bem como por resolver conflitos.

Além disso, esse profissional também preza pela melhoria contínua da equipe e busca remover possíveis impedimentos ao longo do processo.

- Equipe de desenvolvimento:

A equipe de desenvolvimento é o grupo de profissionais responsáveis pela execução das tarefas e produzem os entregáveis do projeto.

Esse time é auto-organizado e atua de maneira colaborativa para alcançar os objetivos definidos em cada sprint.

6. METODOLOGIA

Com base nos objetivos deste trabalho e em todas as tecnologias apoiadas no referencial teórico, a metodologia aplicada será baseada em metodologias ágeis, utilizarei o scrum como framework para organização do projeto. Além disso, será abordado neste capítulo a metodologia aplicada através do Levantamento de requisitos, Diagrama de Casos de Uso, Diagrama de Classes e na Modelagem dos Dados.

6.1. ESCOPO DE PESQUISA

Em busca de uma solução para o problema, a metodologia aplicada foi embasada em uma pesquisa feita em uma biblioteca cristã, a pesquisa foi conduzida com base em entrevistas e observação diretas dos processos internos da biblioteca da igreja. A coleta dos dados foi realizada por meio de reuniões com os administradores da biblioteca, onde o foco era identificar as principais demandas do sistema, com isso, podendo assim listar os requisitos que deve conter no sistema. Deste modo, depois de listar os requisitos foi possível desenvolver os casos de uso do sistema. Com o material comportamental do sistema coletado foi viável elaborar a Modelagem de Dados que engloba o diagrama lógico do banco de dados e o dicionário de dados, finalizando com o diagrama de classes.

6.2. METODOLOGIA ÁGEIS

De metodologia foi utilizado o scrum, que permite uma abordagem iterativa e incremental, garantindo maior flexibilidade e eficiência no processo. Utilizando o Product Backlog com as funcionalidades priorizadas. O processo incluiu planejamento, desenvolvimento em sprints, revisões contínuas e validações da entrega.

6.2.1. SCRUM

O desenvolvimento do sistema seguirá a metodologia Scrum, dividindo o trabalho em ciclos curtos chamados de sprint, onde serão realizadas as entregas de forma incremental e ao fim de cada entrega terão as reuniões (sprint review) para o

acompanhamento do progresso e revisões para verificar se precisa de alguma melhoria para o próximo ciclo (sprint retrospective).

6.2.1.1. EQUIPE DE DESENVOLVIMENTO

A equipe de desenvolvimento será composta pelos seguintes papéis e atribuições, de acordo com a metodologia ágil Scrum estabelecida para este projeto:

Quadro 01 - Equipe de Desenvolvimento

Sprints	Nome	Função	Atribuições
Todas	Fabiano Brito de Souza	Scrum Master	Liderar a equipe durante o andamento do projeto.
	Fabiano Brito de Souza	Desenvolvedor	Desenvolver o software.
	Karina Galego Caetano Oliveira, Letícia Galego Caetano Oliveira	Product Owner	Representante do cliente que ajuda no entendimento do negócio
	Gleison Guardia	Analista	Revisar toda a documentação para garantir a conformidade e qualidade das informações.

6.2.1.2. PRODUCT BACKLOG

Neste tópico serão documentados todos os itens do Product Backlog criados ao longo do desenvolvimento do projeto.

Quadro 02 - Product Backlog

Item	Descrição	Estimativa de Esforço	Nível de Prioridade
1	Desenvolver o documento de Especificação do Projeto	40h	100
2	Construção do Documento de Requisitos de Software	60h	90
3	Elaboração do Diagrama de Caso de Uso	60h	70
4	Escrever o Diagrama de Caso de Uso	40h	70
5	Criar Diagrama de Classes	60h	70
6	Criar Diagramas de Sequência	60h	80
7	Modelar Diagrama de Entidade-Relacionamento do Banco de Dados	40h	80
8	Revisar e organizar todo o projeto	50h	100
9	Desenvolver da Tela de Login	15h	100
10	Desenvolver da Tela de Cadastrar usuário	15h	100
11	Desenvolver da Tela de Perdeu Senha	8h	100
12	Desenvolver da Tela de Redefinir Senha	8h	100
13	Desenvolver da Tela de Home	24h	100
14	Desenvolver da Tela de Menu Lateral	6h	100
15	Desenvolver da Tela de Cadastrar Livro	10h	100
16	Desenvolver da Tela de Consultar Livro	15h	100
17	Desenvolver da Tela de Aprovar Empréstimos	10h	100
18	Desenvolver da Tela de Devolver Livro	10h	100
19	Desenvolver da Tela de Lista de Usuários	6h	100
20	Desenvolver da Tela de Detalhes do Usuário	8h	100
21	Desenvolver da Tela de Doe um livro	8h	80
22	Desenvolver da Tela de Criar Comunidade	15h	90
23	Desenvolver da Tela de Consultar	24h	90

	Comunidade		
24	Desenvolver da Tela de Relatórios	12h	70
25	Desenvolver da Tela de Notificações	10h	70

6.3. LEVANTAMENTO DE REQUISITOS

6.3.1. ENTREVISTAS COM CLIENTE

6.3.1.1. Entrevista 1

Número da Entrevista: 01

Cliente Entrevistado: Atos Fiuza, Karina Galego

Data: 28/10/2024

Nesta primeira entrevista o Scrum Master se reuniu com os administradores da biblioteca para entender os desafios atuais para gerenciar o acervo da biblioteca, como funcionam os fluxos de empréstimos e devoluções de livros, tentando entender quais são as necessidades do dia a dia. Durante a entrevista, foram discutidos os seguintes pontos chaves:

Objetivos e Visão Geral do Sistema

1. Qual é o objetivo principal do sistema que estamos construindo?
 - Resposta: O objetivo do sistema é permitir a localização eficiente dos livros nas bibliotecas, visualizar os livros emprestados, monitorar o inventário de livros disponíveis, e gerenciar o controle de usuários e livros.

2. Quem será o usuário principal do sistema e quais são suas funções e responsabilidades? Quais serão as divisões de acesso (Administrador, Leitor)?
 - Resposta:

- Administrador: Acesso total ao sistema, incluindo funcionalidades como cadastro, exclusão de livros, e geração de relatórios.
- Voluntário: Permissão para registrar empréstimos e devoluções, porém sem acesso a relatórios e exclusão de livros.
- Leitor (Usuário Normal): Acesso ao catálogo de livros, com opção de reservar títulos, mas sem permissões administrativas.

Cadastro e Gerenciamento de Usuários

1. Quais informações serão necessárias para o cadastro de um novo usuário? Qual é o critério para dados obrigatórios e convenientes?
 - Resposta: O sistema deve coletar informações como nome completo (evitando apelidos), idade, gênero, estado civil e igreja local (este último apenas para membros).
2. Alguma funcionalidade específica deve ser disponibilizada ao administrador ou usuários com permissões especiais?
 - Resposta: Sim, o administrador terá acesso a funções exclusivas, como gerenciar permissões de outros usuários, visualizar e gerar relatórios, e excluir ou editar registros de livros

Cadastro e Gerenciamento de Livros

1. Quais informações sobre os livros devem ser cadastradas (ex.: capa, título, autor, status de disponibilidade)?
 - Resposta: O banco de dados atual já cobre as informações necessárias para cada livro.
2. Quais critérios de categorização ou filtros são importantes na busca de livros?
 - Resposta: Cada livro deverá ser identificado com um fundo específico conforme a categoria. Além disso, deve haver uma categoria principal, já que os livros podem pertencer a mais de duas categorias.
3. Como o sistema deve lidar com categorias novas? As categorias devem ser predefinidas ou podem ser criadas pelo usuário?
 - Resposta: Algumas categorias serão predefinidas, mas será possível adicionar novas categorias conforme a necessidade.

Empréstimos e Devoluções de Livros

1. Qual é a política de prazo para empréstimos de livros?
 - Resposta: O prazo padrão para empréstimo é de 30 dias, com possibilidade de renovação. Outros usuários podem reservar o livro para retirada a partir do dia anterior ao vencimento (dia 29), caso o prazo não seja renovado.

Relatório e Visualização de Dados

1. Existem relatórios específicos que você gostaria de visualizar? Se sim, quais dados esses relatórios devem conter?
 - Resposta: Relatórios sobre os dados de devolução, dados de leitura, frequência de devoluções pontuais e identificação de usuários que não cumprem os prazos de devolução

Interface do Usuário e Design

1. Você prefere um sistema mais simples e minimalista ou algo visualmente mais completo?
 - Resposta: Preferimos um design visualmente mais completo, com opções de fundo claro e escuro.
2. Existe algum sistema ou site cuja interface possa servir como inspiração?
 - Resposta: O site JesusCopy é uma referência em termos de interface e experiência do usuário.

Notificações e Alertas

1. O sistema deve enviar notificações para os usuários ou administradores? Em caso afirmativo, quais eventos devem acionar essas notificações (ex.: vencimento do prazo de empréstimo, alerta para retirada de livro)?
 - Resposta: Sim, o sistema deve enviar notificações. Dez dias antes do vencimento do empréstimo, os usuários receberão uma mensagem de lembrete. Também haverá notificações para novos livros, incluindo uma seção exclusiva para destacar as aquisições recentes.

2. Qual seria o canal de preferência para essas notificações: e-mail, notificações internas no sistema ou outra forma?
 - Resposta: Preferimos uma barra de notificações no sistema, com notificações por e-mail para empréstimos que estejam atrasados há mais de 10 dias.

Ideias Adicionais do Cliente

1. Será interessante incluir uma página de sugestões para livros que deveriam estar na biblioteca, onde os leitores podem indicar o nome do livro ou de autores que deveriam ter na biblioteca
2. Os usuários poderão fazer doações para a biblioteca, como a compra de livros por meio de Pix ou doações físicas

6.3.1.2. Entrevista 2

Número da Entrevista: 02

Cliente Entrevistado: Karina Galego, Leticia Galego

Data: 20/11/2024

Nesta segunda entrevista, o scrum master se reuniu com os administradores da biblioteca com algumas telas já estavam desenvolvidas o foco principal da reunião foi validar e coletar feedbacks sobre o design das telas. Durante a entrevista, o scrum master apresentou cada tela e explicou o funcionamento de cada ação, permitindo que os administradores interagissem com o sistema e sugerir melhorias, o principal ponto que foi levantado é que o sistema deveria passar o tom de um ambiente de biblioteca e livros, essas foram as melhorias citadas:

1. Alterar a cor de fundo para uma paleta de cor mais próximo do amarelo claro remetendo a cor de uma página, cor sugerida” #fdf8e2”.
2. Na tela de login ter um botão para ver a senha e quando clicar nesse botão passar uma marca texto remetendo a realidade de um livro.

3. Alterar as cores dos gráficos para uma paleta de verde.
4. Ajustar os cards de emprestar e devolver livro para ser cards pequenos e tendo apenas botões de “ok” ou “recusar” e esses botões serem apenas um emoji.
5. Ajustar o botão de enviar mensagem na parte de comunidade sendo apenas o emoji do avião.

6.3.1.3. Entrevista 3

Número Entrevista: 03

Cliente Entrevistado: Atos Fiuza, Karina Galego, Leticia Galego

Data: 09/02/2025

Nesta terceira entrevista, o Scrum Master apresentou novamente as telas do sistema para a administradora da biblioteca, agora com as melhorias sugeridas na entrevista anterior já implementadas. O objetivo foi validar as alterações realizadas e coletar novos feedbacks sobre funcionalidades, layout e experiência do usuário. Além disso, foram discutidas novas ideias para melhorar a gestão da biblioteca e tornar o sistema mais interativo e dinâmico para os usuários.

O administrador testou todas as telas e funcionalidades e sugeriram as seguintes melhorias e adições:

Ajustes de Interface e Layout

- Tela de Login: Centralizar os textos para melhorar a organização visual e aumentar o tamanho da opção “Membros IMUBS” para melhor destaque.
- Tela do Usuário: Ajusta o tamanho da fonte para melhor legibilidade.
- Padronização da Exibição dos Livros: Tamanho uniforme para nome, autor e avaliação por estrelas, garantindo um layout mais organizado e agradável.
- Remova informações desnecessárias, como endereço da editora, para evitar excesso de dados irrelevantes na interface.

- Descrição dos Livros: Limitar o tamanho da descrição para não comprometer o design da tela.

Novas Funcionalidades Solicitadas

- Indicação de Livros da Semana: Crie uma funcionalidade onde o administrador pode destacar 5 livros semanalmente.
- Seção de Doações na Página Inicial: Inclui uma área na tela principal onde os usuários podem visualizar e realizar doações de livros.
- Sugestões de Livros Relacionados: Implementar na tela de consulta de um livro a opção “Você também pode gostar?”, fornecendo livros semelhantes ao pesquisado.
- Histórico e Solicitações: Incluir dados na solicitação de empréstimos e no histórico de leitura. Após uma semana, desligue automaticamente se as sugestões não forem atendidas.

Melhorias na Experiência do Usuário

- Notificações Personalizadas: Adicione notificações para usuários informando sobre a aprovação ou reprovação de empréstimos, com uma linguagem mais informal e amigável.
- Categorias e Subcategorias: Melhorar a navegação permitindo que o sistema puxe automaticamente as categorias e subcategorias dos livros cadastrados.
- Avatar Personalizado para Objetivos: Crie um sistema onde o usuário possa personalizar seu avatar, de acordo com os objetivos alcançados na plataforma.
- Objetivo: Implementar um elemento visual de progresso, onde um personagem carrega uma cruz e avança conforme os objetivos de leitura são cumpridos.

Relatórios e Análises para Administração

Para facilitar a gestão da biblioteca, foram solicitados novos relatórios que permitirão monitorar considerações importantes, como:

- Faixa etária dos leitores ativos.
- Lista de livros em atraso.
- Categorias de livros mais lidas.

- Média de leitura por usuário.
- Tempo estimado de leitura para cada livro.

Personalização Visual

- Mudar as cores da *Navbar*

6.3.2. DOCUMENTO DE REQUISITOS

Com base nas entrevistas nas entrevistas realizadas com os administradores da biblioteca, foram identificados e coletados os seguintes requisitos para o software:

Quadro 03: Fazer Login

Requisito Funcional		
Nome:	Login	Código: RF1
Descrição:	O sistema deve permitir que o usuário faça login inserindo o nome de usuário e senha.	
	Estimativa de Esforço: 10h	Prioridade: Alta
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	As informações do usuário que deverão ser armazenadas pela função no banco de dados são: E-mail e senha	Especificação /Confiabilidade
1.2	A função deverá ter 2 textbox um para digitar o email e outro para digitar a senha	Especificação
1.3	Deverá ser usada uma máscara enquanto o usuário digita a senha.	Especificação /Segurança
1.4	A função deverá exibir os botão "Entrar" e, quando aplicável, "Entrando...".	Interface
1.5	A função deve exibir mensagens claras de erro em caso de falha na autenticação.	Usabilidade
1.6	A função deverá ter um link "esqueceu a senha?" oferecendo ao usuário uma navegação simples para a página de esqueceu senha.	Usabilidade
1.8	Caso o login seja bem-sucedido, o sistema deverá redirecionar o usuário para a página de home de forma rápida após a autenticação.	Desempenho
1.9	A função deverá ter uma mensagem "Ainda não possui conta? Cadastre-se no site." e um botão "cadastre-se", oferecendo ao usuário uma navegação simples para a página de criação de conta.	Usabilidade
1.10	O requisito funcional deve estar aberto 24h por dia para o usuário.	Desempenho
1.11	Deverá ter um botão com o símbolo de olho que quando o usuário clicar ele consegue ver a senha digitada.	Interface

Quadro 04: Cadastre-se

Requisito Funcional		
Nome:	Cadastre-se	Código: RF2
Descrição:	O sistema deve permitir que o usuário crie sua conta.	
Estimativa de Esforço:	10h	Prioridade: Alta
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	As informações do usuário que deverão ser armazenadas pela função no banco de dados são: nome, email, senha, telefone, igreja local, foto, tipo usuário e endereço	Especificação /Confiabilidade
1.2	A função deverá ter textbox um para digitar o nome, email, senha, confirmar senha, telefone, igreja local, foto, tipo usuário e endereço	Especificação
1.3	Deve ser separado a parte de dados do usuário e endereço, deve ter uma aba para digitar os dados do usuário e uma para digitar o endereço	Especificação /interface
1.4	Deverá ser usada uma máscara enquanto o usuário digita a senha.	Especificação /Segurança
1.5	A função deverá exibir os botão "Cadastrar" e, quando aplicável, "Cadastrando..." e fica não clicável para que o usuário não faça mais de uma requisição no banco de dados.	Interface /segurança
1.6	A função deve exibir mensagens claras de erro em caso de falha na autenticação.	Usabilidade
1.7	Caso o cadastro seja bem-sucedido, o sistema deverá redirecionar o usuário para a página de conta em 2 segundos após a autenticação.	Desempenho
1.8	A função deverá ter uma mensagem "Já possui conta?Faça Login." e um botão "Login", oferecendo ao usuário uma navegação simples para a página de login.	Usabilidade

Quadro 05: Esqueceu senha?

Requisito Funcional		
Nome:	Esqueceu senha?	Código: RF3
Descrição:	O sistema deve permitir que o usuário redefina sua senha inserindo o e-mail.	
Estimativa de Esforço:	5h	Prioridade: Média
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	A função deverá ter textbox um para digitar o e-mail para que envie uma mensagem de confirmação da alteração da senha	Especificação
1.2	A função deverá ter o botão "Enviar e-mail" e após de ter clicado fique "Reenviar e-mail", porém só fica clicável após 1 minuto	Interface
1.3	A função deverá ter um contador que após o usuário ter digita o e-mail e clicado no botão de enviar e-mail comece diminuir dos segundos	Especificação /interface

Quadro 06: Redefinir Senha

Requisito Funcional		
Nome:	Redefinir Senha	Código: RF4
Descrição:	O sistema deve permitir que o usuário redefina sua senha	

Estimativa de Esforço: 5h		Prioridade: Média
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	A função deverá ter textbox um para digitar a “Nova senha” e “Confirme a senha”	Especificação
1.2	A função deverá ter o botão “Salvar” e após de ter clicado fique “Salvando senha...”	Interface
1.3	Deverá ser usada uma máscara enquanto o usuário digita a senha.	Especificação /Segurança
1.4	A função deve exibir mensagens claras de erro em caso de senhas incorretas.	Usabilidade

Quadro 07: Home

Requisito Funcional		
Nome:	Home	Código: RF5
Descrição:	O sistema deve permitir que o usuário navegue pelas funcionalidades	
Estimativa de Esforço:	12h	Prioridade: Alta
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	A função deverá ter um carrossel dos livros separados por categorias	Especificação /interface
1.2	A função deverá ter cards de cada livro, cada card deverá ter uma imagem, título e autor, avaliação do livro	Especificação /Interface
1.3	A função deverá ter uma navbar com o ícone para o menu lateral, um campo para pesquisa, foto do usuário	Especificação /Interface
1.4	A função deverá ter o cursor pointer para o usuário saber que o card é clicável	Especificação
1.5	A função deverá redirecionar o usuário para a página de “consultar livro” quando o usuário clicar sobre o livro	Especificação

Quadro 08: Cadastrar Livro

Requisito Funcional		
Nome:	Cadastrar Livro	Código: RF6
Descrição:	O sistema deve permitir que o usuário cadastre os livros	
Estimativa de Esforço:	12h	Prioridade: Alta
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	As informações do usuário que deverão ser armazenadas pela função no banco de dados são: nome do livro, autor, categoria(s), descrição, ano, publicação, quantidade de páginas, quantidade de livros	Especificação /Interface
1.2	A função deverá ter textbox um para digitar: nome do livro, autor, categoria(s), descrição, ano, publicação, quantidade de páginas, quantidade de livros	Especificação /Interface
1.3	A função deverá exibir os botões "Salvar" e, quando aplicável, "salvando..." e fica não clicável para que o usuário não faça mais de uma requisição no banco de dados.	Interface /segurança

1.4	A função deve exibir mensagens claras de erro em caso de falha na salvção dos dados.	Usabilidade
1.5	Caso o cadastro seja bem-sucedido, deverá aparecer uma mensagem de confirmação.	Desempenho
1.6	O requisito funcional deve estar aberto 24h por dia para o usuário.	Desempenho
1.7	A função deverá ter inputs de “cor topo” e “cor base” para o usuário escolher a cor daquela categoria.	Especificação
1.8	A função deverá incluir botões associados a campos que podem se repetir, como Categoria, Autor e Editora. Ao serem clicados pelo usuário, esses botões devem abrir um menu suspenso que liste os valores já existentes no banco de dados.	Especificação

Quadro 09: Consultar Livro

Requisito Funcional		
Nome:	Consultar Livro	Código: RF7
Descrição:	O sistema deve permitir que o usuário consulte o livro desejado e o reserve	
Estimativa de Esforço:	10h	Prioridade: Alta
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	A função deverá mostrar as seguintes informações: foto de capa, título, autor, categoria, descrição e um botão de reservar	Especificação
1.2	A função deverá exibir os botões "Reservar" que deverá ser da cor verde se o livro estiver disponível e caso não ele deverá ser da cor vermelho e não clicável e aparecerá "Indisponível"	Interface /segurança
1.3	A função deverá mostrar uma mensagem de confirmação quando o usuário clicar em “Reservar” deverá aparecer esta mensagem “Você reservou este livro, vá até a biblioteca da igreja nos finais de semana para retirar este livro, caso você demore mais de 7 para retirar o livro esta reserva será cancelada”	Especificação /interface
1.4	A função deverá ter uma tabnavigador com “Comentários” e “Avaliações” permitindo que o usuário comente ou avalie sobre o livro	Especificação
1.5	A tab de comentários deverá ter um input para o usuário comentar e exibir todos os comentários.	Especificação
1.6	A função deverá ter a opção de o usuário curtir ou responder o comentário.	Especificação /interface
1.7	A tab de avaliações deverá ter um input para o usuário digitar um comentário sobre a avaliação e 5 estrelas para ele avaliar de 0 a 5 estrelas.	Especificação
1.8	O requisito funcional deve estar aberto 24h por dia para o usuário.	Desempenho

Quadro 10: Lista de Empréstimos

Requisito Funcional		
Nome:	Lista de empréstimos	Código: RF8
Descrição:	O sistema deve permitir que o administrador veja a lista de empréstimo e requerimentos	
Estimativa de Esforço:	10h	Prioridade: Alta
Requisitos Não funcionais		
ID NF	Descrição	Categoria

1.1	Apenas o administrador terá acesso a essa função	Segurança
1.2	A função deverá mostrar as seguintes informações: foto de capa, nome do usuário, título, autor, data de vencimento e dias restantes.	Especificação
1.3	A função deverá ter uma borda interativas para o administrador saber como anda o empréstimo: “borda verde: o empréstimo ainda falta mais de 10 dias para vencer”, “borda amarela: o empréstimo ainda falta menos de 10 dias para vencer”, “borda vermelha: o empréstimo está vencido”	Interface /segurança
1.4	A função deverá ter um cursor pointer para o usuário saber que é clicável	Interface /confiabilidade
1.5	A função deverá ter o botão devolver livro e quando o usuário clicar aparecer a mensagem de “livro devolvido com sucesso”.	Especificação /Interface

Quadro 11: Aprovar Empréstimos

Requisito Funcional		
Nome:	Aprovar Empréstimos	Código: RF9
Descrição:	O sistema deve permitir que o administrador aprove ou negue a solicitação do empréstimo	
Estimativa de Esforço:	10h	Prioridade: Alta
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	Apenas o administrador terá acesso a essa função	Segurança
1.2	A função deverá mostrar as seguintes informações: foto de capa, nome do usuário, título e botões de confirmação ou não confirmação do empréstimo.	Especificação
1.3	A função deverá ter um cursor pointer para o usuário saber que é clicável	Interface /confiabilidade
1.4	A função deverá exibir a mensagem “Empréstimo aprovado com sucesso!” quando o usuário clicar no botão de confirmação.	Especificação /interface
1.5	A função deverá exibir a mensagem “Empréstimo rejeitado com sucesso!” quando o usuário clicar no botão de não confirmação.	Especificação /interface

Quadro 12: Sugere/doe um livro

Requisito Funcional		
Nome:	sugere/doe um livro	Código: RF10
Descrição:		
Estimativa de Esforço:	10h	Prioridade: Alta
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	A função deverá ter uma mensagem, um qr code, pix, e tabnavigador com a opção de “livro” ou “autor” e uma lista com as sugestões dos outros usuários	Especificação
1.2	A tab de livro deverá ter os inputs de “nome do livro”, “descrição do livro” e “Por que você quer este livro na biblioteca?”	Especificação
1.3	A tab de autor deverá ter os inputs de “nome do autor” e “Por que você quer este autor na biblioteca?”	Especificação

1.4	A sugestão do usuário deverá ser um card com “nome do usuário ou autor”, “motivo”, “descrição (se for livro)” e “data da sugestão”	Especificação /interface
-----	------------------------------------------------------------------------------------------------------------------------------------	--------------------------

Quadro 13: Criar Comunidade

Requisito Funcional		
Nome:	Criar Comunidade	Código: RF11
Descrição:	O sistema deve permitir que o usuário crie ou entre em uma comunidade	
Estimativa de Esforço:	10h	Prioridade: Média
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	A função deverá ter os inputs de “nome da comunidade” e “Descrição” e um select com as opções “pública” ou “privada” e uma lista das comunidades já criadas	Especificação
1.2	A função deverá ter um card para cada comunidade criada com o nome e descrição da comunidade, se a comunidade for pública deverá ter o botão “acessar” e se for privada o botão “solicitar”.	Especificação /interface
1.3	A função deverá permitir que o usuário solicite apenas uma vez a entrada na comunidade e quando tiver solicitado o botão deverá aparecer “solicitado”.	Confiabilidade /Segurança

Quadro 14: Comunidade

Requisito Funcional		
Nome:	Comunidade	Código: RF12
Descrição:	O sistema deve permitir que o usuário veja e interaja com os amigos na sua comunidade	
Estimativa de Esforço:	10h	Prioridade: Alta
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	A função deverá ter uma parte de gerenciar as comunidades privadas para aceitar ou rejeitar os usuários, apenas o administrador da comunidade tem acesso a essa parte.	Especificação /Segurança
1.2	A função deverá ter gráficos e ranking para ver quais usuários estão mais lendo ou mais próximo do objetivo.	Especificação /Interface
1.3	A função deverá ter um input para o usuário inserir o progresso do dia.	Especificação
1.4	A função deverá ter uma aba lateral para que os usuários troquem mensagens, deverá ter um input e um botão para enviar a mensagem.	Especificação /interface
1.5	O requisito funcional deve estar aberto 24h por dia para o usuário.	Desempenho

Quadro 15: Usuários

Requisito Funcional		
Nome:	Usuários	Código: RF13
Descrição:	O sistema deve permitir que o administrador veja todos os usuários do sistema	
Estimativa de Esforço:	10h	Prioridade: Alta
Requisitos Não funcionais		

ID NF	Descrição	Categoria
1.1	Apenas o administrador terá acesso a essa função	Segurança
1.2	A função deverá ter card para cada usuário e o card deverá conter “foto de usuário”, “email” e “nível de prioridade”.	Especificação
1.3	A função deverá ter o cursor pointer para o usuário saber que o card é clicável	Especificação
1.4	A função deverá redirecionar o usuário para a página de “consultar usuário” quando clicar no card.	Especificação

Quadro 16: Consultar Usuário

Requisito Funcional		
Nome:	Consultar Usuário	Código: RF14
Descrição:	O sistema deve permitir que o veja os dados do usuário.	
Estimativa de Esforço:	8h	Prioridade: Média
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	Apenas o administrador terá acesso a essa função	Segurança
1.2	A função deverá ter a foto do usuário, nome, e-mail, tipo de usuário e o histórico do usuário.	Especificação
1.3	A função deverá conter o botão de “editar” para o administrador editar o tipo do usuário e dar permissões	Especificação

Quadro 17: Conta do Usuário

Requisito Funcional		
Nome:	Conta do Usuário	Código: RF15
Descrição:	O sistema deve permitir que o usuário veja e edite seus dados.	
Estimativa de Esforço:	8h	Prioridade: Alta
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	A função deverá ter todos os dados do usuário e o histórico de empréstimo dele.	Especificação
1.2	A função deverá ter o botão de editar para o usuário edite seus dados se necessário	Especificação

Quadro 18: Relatórios

Requisito Funcional		
Nome:	Relatórios	Código: RF16
Descrição:	O sistema deve permitir que o administrador veja os relatórios.	
Estimativa de Esforço:	12h	Prioridade: Alta
Requisitos Não funcionais		

ID NF	Descrição	Categoria
1.1	Apenas o administrador terá acesso a essa função	Segurança
1.2	A função deverá ter dois inputs de data para o usuário selecionar a data de início e data de fim.	Especificação
1.3	A função deverá ter um select com as opções “30 dias”, “60 dias”, “90 dias” e “1 ano” para facilitar o usuário a selecionar o período do relatório.	Especificação
1.4	A função deverá ter gráficos e tabelas para facilitar a visualização dos relatórios	Especificação /Interface

Quadro 19: Notificações

Requisito Funcional		
Nome:	Notificações	Código: RF17
Descrição:	O sistema deve permitir que o usuário veja as notificações	
Estimativa de Esforço: 8h	Prioridade: Média	
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1	A função deverá mandar notificações para o usuário quando o livro estiver perto de vencer ou vencido, quando ele fizer uma solicitação de empréstimo notificar quando for aprovado ou rejeitado	Especificação
1.2	A função deverá ter o botão de “marcar como lida”, assim o contador de notificações diminui.	Especificação
1.3	O requisito funcional deve estar aberto 24h por dia para o usuário.	Desempenho
1.4	A função deverá mandar notificações para o usuário quando o livro estiver perto de vencer ou vencido, quando ele fizer uma solicitação de empréstimo notificar quando for aprovado ou rejeitado	Especificação
1.5	A função deverá ter o botão de “marcar como lida”, assim o contador de notificações diminui.	Especificação

6.4. CASO DE USO

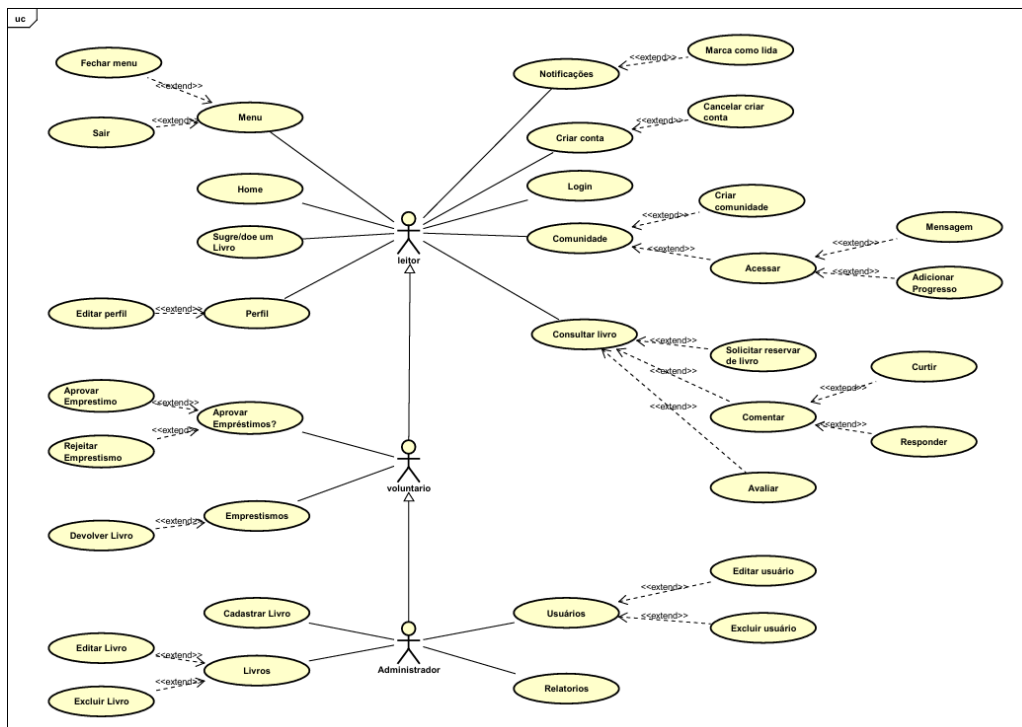
Um caso de uso representa a interação do usuário com um sistema em condições específicas e pode ser documentado de diversas maneiras, como texto narrativo, enumeração de tarefas ou passos, e diagrama, de acordo com Pressman (2010).

Os diagramas de caso de uso é uma maneira eficiente de simplificar em passos a futura interações do usuário com o sistema, tornando assim, uma clareza no momento em que for implementado a aplicação web.

6.4.1. Diagrama de caso uso

A figura abaixo representa o diagrama de casos de uso da aplicação pesquisada neste trabalho:

Figura 15: Diagrama de caso uso



Fonte: Próprio Autor

6.4.2. Diagrama de caso uso expandido

Aprofundando no diagrama da figura, temos as descrições detalhadas de cada caso de uso, contendo os atores, as pré e pós condições, o importante fluxo principal que cita o passo a passo que o usuário fará na função e as exceções que podem ocorrer no decorrer das ações operadas pelo usuário.

6.4.2.1. Caso de uso - Fazer Login

Caso de Uso: Fazer Login (CSU01)

Descrição: O usuário deve fazer seu login no sistema.

Ator Primário: Administrador

Atores Secundários: Leitor, Voluntários

Pré-condições: O usuário deve realizar o *login* no sistema

Fluxo Principal:

1. O usuário acessa a página de login.
2. O sistema exibe os campos de entrada para e-mail e senha.
3. O usuário insere suas credenciais.
4. O sistema valida as informações.
5. Caso as credenciais sejam corretas, o sistema de redirecionamento para a página inicial.

Fluxos Alternativos:

- **FA1 - Esqueceu a Senha:**
 1. O usuário clicou no link "Esqueceu a senha?".
 2. O sistema exibe uma tela para redefinir a senha solicitando o e-mail cadastrado.
 3. O usuário inseriu o e-mail e clicou em "Enviar e-mail".
 4. O sistema envia um e-mail com instruções para redefinir a senha e exibe uma mensagem de confirmação.
 5. O usuário clicou no link no e-mail recebido, inseriu uma nova senha e confirmou.
 6. O sistema salva a nova senha e exibe uma mensagem de sucesso.
- **FA2 - Criar Nova Conta:**
 1. O usuário clicou no link "Cadastre-se" na tela de login.
 2. O sistema exibe uma tela de cadastro com campos para informações pessoais e endereço.
 3. O usuário preenche as informações e clicar em "Cadastrar".
 4. O sistema valida os dados e cria a conta, redirecionando o usuário para a página de login com uma mensagem de sucesso.

Pós-condições:

1. Se o login for bem-sucedido, o usuário é autenticado e redirecionado para a tela de home.
2. Se o usuário redefinir a senha, o sistema atualiza as credenciais com a nova senha.
3. Se o usuário criar uma conta, os dados são armazenados no banco de dados, e o usuário pode efetuar o login.

6.4.2.2. Caso de uso - Home

Caso de Uso: Home (CSU02)

Descrição: O sistema deve mostrar os livros em cards separados por categorias.

Ator Primário: Leitor

Atores Secundários: Administrador, Voluntários

Pré-condições:

- O usuário deve realizar login no sistema.
- O sistema deve estar conectado à base de dados com informações sobre livros e categorias.
- O usuário deve ter acesso à página inicial do sistema.

Fluxo Principal:

1. O usuário acessa a página inicial do sistema.
2. O sistema solicita as categorias e os livros do banco de dados.
3. O sistema exibe as categorias disponíveis como seções organizadas em carrosséis.
4. Para cada categoria, o sistema lista os livros separados em cards correspondentes com:
 - Foto da capa, título, autor e avaliação média.
5. O usuário pode navegar pelos livros em cada carrossel.

6. O usuário clica em um livro para visualizar seus detalhes.
7. O sistema redireciona o usuário para a página de consulta do livro selecionado.

Fluxos Alternativos:

- **FA1 - Nenhuma Categoria Encontrada:**
 1. Caso não existam categorias no banco de dados, o sistema exibe uma mensagem informando que não há dados disponíveis.
- **FA2 - Nenhum Livro Encontrado em uma Categoria:**
 1. Caso uma categoria não tenha livros associados, o sistema exibe a categoria sem o carrossel de livros, acompanhado de uma mensagem informando a ausência de dados.

Pós-condições:

1. Os dados de categorias e livros são carregados com sucesso para exibição ao usuário.
2. Se o usuário clicar em um livro, é exibida a página de detalhes do livro correspondente.

6.4.2.3. Caso de uso - Cadastrar Livro

Caso de Uso: Cadastrar Livro (CSU03)

Descrição: Este caso de uso descreve o processo para cadastrar um novo livro no sistema, preenchendo os dados básicos, informações da editora, categorias, autores, estoque e a capa do livro.

Ator Primário: Administrador

Pré-condições:

- O ator deve estar autenticado e ter permissão para cadastrar livros.
- A conexão com a API (ViaCEP e backend) deve estar disponível.

Fluxo Principal:

1. O ator acessa a página de **Cadastrar Livro** no sistema.
2. O sistema exibe um formulário com as abas:
 - **Dados do Livro:** Nome, descrição, ano de publicação, quantidade de páginas, quantidade em estoque, categoria e subcategorias.
 - **Editora:** Nome, endereço (CEP, rua, bairro, número, cidade, estado).
3. O ator preenche os campos:
4. O ator clica no botão Salvar.
5. O sistema valida os dados e salva o registro no banco de dados.
6. Caso o cadastro seja bem-sucedido, o sistema exibe uma mensagem de confirmação: "Livro cadastrado com sucesso!".

O ator retorna à tela inicial ou continua no formulário para cadastrar outro livro.

Fluxos Alternativos:

- **FA1 - CEP inválido ou não encontrado:**
 1. O ator insere um CEP que não retorna resultados.
 2. O sistema exibe uma mensagem: "CEP inválido ou não encontrado".
 3. O ator pode corrigir o CEP ou preencher manualmente os campos de endereço.

- **FA2 - Erro ao salvar o livro:**
 1. Durante o salvamento, ocorre uma falha de comunicação com o servidor ou validação.
 2. O sistema exibe uma mensagem: "**Erro ao cadastrar livro.**"
 3. O ator pode tentar novamente ou entrar em contato com o suporte.

- **FA3: Campos obrigatórios não preenchidos:**

1. O ator tenta salvar o formulário sem preencher campos obrigatórios.
2. O sistema exibe mensagens de erro indicando os campos obrigatórios a serem preenchidos.

Pós-condições:

3. O livro cadastrado fica disponível para consulta na biblioteca do sistema.

6.4.2.4. Caso de uso - Consultar Livro

Caso de Uso: Consultar Livro (CSU04)

Descrição: Este caso de uso descreve como um usuário pode consultar os detalhes de um livro, fazer uma reserva, comentar e avaliar.

Ator Primário: Leitor

Atores Secundários: Administrador, Voluntários

Pré-condições:

- O usuário deve estar autenticado para realizar uma reserva, comentar ou avaliar.
- O sistema deve estar conectado ao banco de dados para consultar os detalhes do livro e processar as ações do usuário.

Fluxo Principal:

1. O caso de uso começa quando o usuário seleciona um livro para consultar detalhes.
2. O sistema exibe as seguintes informações sobre o livro: Nome do Livro, Nome do Autor, Categoria Principal, Descrição, Nome da Editora, Foto de Capa, Quantidade em Estoque, Quantidade de Páginas.

3. O sistema carrega e exibe:

- Comentários relacionados ao livro, incluindo:
 - Usuário que comentou.
 - Texto do comentário.
 - Curtidas e respostas aos comentários.
 - Data do comentário.
- Avaliações feitas no livro, incluindo:
 - Nota dada pelo usuário.
 - Texto da avaliação.
 - Data da avaliação.
 - Foto e nome do avaliador.

4. O usuário pode executar as seguintes ações:

- **Reservar o livro:** Se a quantidade em estoque for maior que zero, o usuário clica em "Reservar".
 - O sistema registra a reserva e atualiza o estoque do livro.
- **Comentar sobre o livro:** O usuário insere um texto no campo de comentário e o envia.
 - O sistema registra o comentário e o exibe na lista de comentários.
- **Responder a um comentário:** O usuário seleciona um comentário e insere uma resposta no campo apropriado.
 - O sistema registra a resposta e a associa ao comentário original.
- **Avaliar o livro:** O usuário insere uma nota (de 1 a 5) e um comentário adicional opcional.
 - O sistema registra a avaliação e a exibe na lista de avaliações.

Fluxos Alternativos:

- **FA1 - Livro Indisponível:**
- Se o livro tiver quantidade em estoque igual a zero:
 1. O sistema exibe a mensagem: "Livro indisponível no momento."
 2. O fluxo principal é retomado a partir do passo 3.

- **FA2 - Comentário ou Avaliação Não Submetidos:**
- Se o usuário tentar enviar um comentário ou avaliação sem preencher os campos obrigatórios:
 1. O sistema exibe a mensagem: "Preencha todos os campos obrigatórios."
 2. O fluxo principal é retomado.

- **FA3 - Erro na Comunicação com o Servidor**
- Se ocorrer um erro ao tentar realizar uma reserva, enviar um comentário ou avaliação:
 1. O sistema exibe a mensagem: "Erro ao processar sua solicitação. Tente novamente mais tarde."
 2. O fluxo principal é retomado.

Pós-condições:

4. O sistema atualiza as informações do livro com a reserva realizada.
5. Comentários e avaliações ficam registrados e disponíveis para outros usuários consultarem.
6. A interação do usuário com curtidas, respostas ou avaliações é persistida no banco de dados.

6.4.2.5. Caso de uso - Aprovar Empréstimos

Caso de Uso: Aprovar Empréstimos (CSU05)

Descrição: Permite que o administrador visualize e aprove ou rejeite solicitações de empréstimos de livros realizadas por usuários.

Ator Primário: Administrador

Atores Secundários: Voluntários

Pré-condições:

- O usuário deve estar autenticado e ter permissões para aprovar ou rejeitar um empréstimos
- O sistema deve estar conectado ao banco de dados para consultar os detalhes do empréstimo e processar as ações do usuário.

Fluxo Principal:

1. O administrador acessa a tela "Aprovar Empréstimos".
2. O sistema exibe uma lista de solicitações de empréstimos pendentes, incluindo informações como: Foto da capa do livro, Nome do usuário solicitante, Nome do livro solicitado, Data prevista de devolução.
3. O administrador pode clicar no botão "Aprovar" ou "Rejeitar".
4. Se a requisição for bem-sucedida, o empréstimo é removido da lista.
5. O administrador recebe feedback visual (alerta) sobre a ação realizada (sucesso ou erro).

Fluxos Alternativos:

● **FA1 - Fluxo de erro:**

1. Caso ocorra um erro na API ao aprovar/rejeitar um empréstimo:
 - O sistema exibe uma mensagem de erro informando o problema.
 - O empréstimo permanece na lista até que a operação seja concluída com sucesso.

Pós-condições:

1. Em caso de aprovação:
 - O status do empréstimo é atualizado no banco de dados para "Aprovado".
 - O registro do empréstimo é movido para a lista de empréstimos ativos.

2. Em caso de rejeição:
 - O status do empréstimo é atualizado no banco de dados para "Rejeitado".
 - O registro é removido da lista de solicitações pendentes.
3. A lista de solicitações pendentes é atualizada dinamicamente para refletir as mudanças.

6.4.2.6. Caso de uso - Empréstimos

Caso de Uso: Empréstimos (CSU06)

Descrição:

Ator Primário: Administrador

Atores Secundários: Voluntários

Pré-condições:

- O usuário deve estar autenticado para realizar uma devolução do livro.
- O sistema deve estar conectado ao banco de dados para consultar os detalhes do empréstimo e processar as ações do usuário.

Fluxo Principal:

1. O ator acessa a tela de gerenciamento de empréstimos.
2. O sistema exibe uma lista de empréstimos ativos, contendo: Foto da capa do livro, Nome do usuário, Nome do livro, Data de devolução prevista, Dias restantes para devolução com indicador de cor.
3. O ator confirmou a devolução do livro clicando no botão "Devolver Livro".
4. O sistema atualiza o status do empréstimo e do livro no banco de dados:
 - O status do empréstimo é marcado como "Concluído".
 - O status do livro é alterado para "Disponível".

5. O sistema exibe uma mensagem de confirmação de sucesso ou erro.
6. O Administrador retorna à tela inicial ou realiza novas ações.

Fluxos Alternativos:

- **FA1 - Fluxo de erro:**
2. Caso ocorra um erro na API ao devolver um livro:
 - O sistema exibe uma mensagem de erro informando o problema.
 - O empréstimo permanece na lista até que a operação seja concluída com sucesso.

Pós-condições:

1. O status do empréstimo é atualizado para "Concluído" no banco de dados.
2. O status do livro é atualizado para "Disponível" no banco de dados.
3. O sistema registra a data e o horário da devolução no histórico do empréstimo.

6.4.2.7. Caso de uso - Conta do Usuário

Caso de Uso: Conta do Usuário (CSU07)

Descrição: Este caso de uso descreve o gerenciamento das informações da conta de um usuário, permitindo a visualização de dados pessoais, endereço e histórico de atividades.

Ator Primário: Leitor

Atores Secundários: Administrador, Voluntários

Pré-condições:

- O usuário deve estar autenticado no sistema.

Fluxo Principal:

1. O sistema recebe o ID do usuário como parâmetro.
2. O sistema consulta os dados do usuário na base de dados, incluindo:
 - Dados pessoais (nome, email, telefone, data de nascimento).

- Endereço (CEP, rua, número, bairro, cidade, estado).
 - Histórico de atividades (ações realizadas, empréstimos etc.).
3. O sistema verifica se o usuário possui uma foto cadastrada.
 4. O sistema exibe todas as informações do usuário em uma interface gráfica organizada: Dados pessoais e foto de perfil, Endereço formatado, Histórico de atividades, recuperado dinamicamente.
 5. O usuário pode navegar entre as informações exibidas.

Fluxos Alternativos:

- **FA1 - Foto do Usuário Não Encontrada:**
 1. Se o usuário não possui uma foto cadastrada, o sistema exibe uma imagem padrão de perfil.
- **FA2 - Histórico Não Disponível:**
 1. Caso o histórico do usuário não esteja disponível, o sistema exibe uma mensagem informativa e oculta o bloco correspondente.

Pós-condições:

1. O usuário consegue visualizar todos os dados associados à sua conta, incluindo histórico de atividades.

6.4.2.8. Caso de uso - Usuários

Caso de Uso: Usuários (CSU08)

Descrição: Este caso de uso descreve a gestão de usuários no sistema, onde o administrador pode visualizar todos os usuários registrados e acessar detalhes sobre cada um deles.

Ator Primário: Administrador

Pré-condições:

- O administrador deve estar autenticado no sistema.

- O administrador tem permissão para acessar e gerenciar os dados dos usuários.
- O sistema deve conter usuários registrados.

Fluxo Principal:

1. O administrador acessa a interface de gestão de usuários.
2. O sistema exibe a lista de todos os usuários registrados, mostrando nome, email, tipo de usuário e foto.
3. O administrador pode clicar no nome de um usuário para visualizar os detalhes completos do usuário.
4. Ao clicar sobre o usuário é redirecionado a página de consultar usuários.
5. O administrador verifica os dados do usuário.

Pós-condições:

1. O administrador pode visualizar e gerenciar os usuários.
2. O sistema mantém as informações de cada usuário atualizadas.
3. O administrador pode acessar os detalhes completos do usuário e realizar ações necessárias.

6.4.2.9. Caso de uso - Consultar Usuário**Caso de Uso: Consultar Usuário (CSU9)**

Descrição: Este caso de uso permite que o administrador consulte os detalhes completos de um usuário registrado no sistema, incluindo nome, e-mail, tipo de usuário, e foto. Além disso, o administrador pode editar o tipo de usuário (por exemplo, para alterar a permissão de acesso) e visualizar o histórico de atividades do usuário.

Ator Primário: Administrador.

Pré-condições:

- O administrador deve estar autenticado no sistema.

- O administrador deve ter permissões de acesso para visualizar e editar as informações de qualquer usuário.
- O administrador está acessando uma página de detalhes de usuário com base no ID fornecido.

Fluxo Principal:

1. O administrador acessa a página de detalhes de um usuário, fornecendo o ID do usuário.
2. O sistema carrega as informações do usuário, como nome, email, foto e tipo de usuário.
3. O sistema exibe essas informações na interface do administrador.
4. Caso o administrador deseje editar o tipo de usuário, ele pode clicar em um botão de edição.
5. O administrador altera o tipo de usuário (por exemplo, de "Leitor" para "Administrador") e clica no botão "Salvar".
6. O sistema valida a alteração e a salva no banco de dados.
7. O sistema exibe a confirmação de que a alteração foi realizada com sucesso.
8. O sistema também exibe o histórico de atividades do usuário, através de um componente adicional.

Fluxos Alternativos:

- **FA1 - Erro ao Carregar Dados**

1. Caso ocorra um erro ao carregar os dados do usuário, o sistema exibe uma mensagem de erro informando que houve um problema ao buscar os dados.

- **FA2 - Erro ao Salvar Alteração**

1. Caso ocorra um erro ao tentar salvar a alteração no tipo de usuário, o sistema exibe uma mensagem de erro, mas mantém a interface acessível para tentar novamente.

Pós-condições:

1. O tipo de usuário é atualizado no banco de dados, caso tenha sido alterado.
2. O administrador pode visualizar as informações atualizadas do usuário.
3. O histórico de atividades do usuário é exibido corretamente.

6.4.2.10. Caso de uso - Sugere/Doe Um Livro**Caso de Uso: Sugere/Doe Um Livro (CSU10)**

Descrição: Este caso de uso descreve o processo de sugestão de livros ou autores, bem como a doação de livros físicos ou contribuições via Pix para enriquecer a biblioteca.

Ator Primário: Leitor

Atores Secundários: Administrador, Voluntários

Pré-condições:

- O Leitor deve estar logado na plataforma para sugerir um livro ou autor.

Fluxo Principal:

1. O Leitor acessa a página de sugestão e doação de livros.
2. O Leitor escolhe entre sugerir um livro ou sugerir um autor, alternando entre as opções por meio de abas.
3. O Leitor preenche os campos obrigatórios:
 - Para sugerir um livro: nome do livro, descrição do livro e motivo da sugestão.
 - Para sugerir um autor: nome do autor e motivo da sugestão.
4. O Leitor submete o formulário de sugestão.
5. O sistema salva a sugestão no banco de dados e exibe a confirmação de sucesso.

6. O Leitor visualiza sugestões de outros usuários, incluindo o nome do livro/autor sugerido, descrição e motivo da sugestão.
7. O Leitor também é informado sobre como pode realizar a doação do livro (via Pix ou física) e visualiza um QR code para doações via Pix.

Fluxos Alternativos:

- **FA1 - Campos obrigatórios vazios**

1. Caso o Leitor não preencha todos os campos obrigatórios, o sistema exibe uma mensagem de erro solicitando o preenchimento dos campos.

Pós-condições:

1. A sugestão do livro ou autor é registrada no sistema.
2. O usuário pode visualizar sua sugestão junto com as sugestões de outros usuários.
3. O usuário é informado sobre a doação via Pix e tem acesso ao QR code.

6.4.2.11. Caso de uso - Criar Comunidades

Caso de Uso: Criar Comunidades (CSU11)

Descrição:

Ator Primário: Leitor

Atores Secundários: Administrador, Voluntários

Pré-condições:

- O usuário deve estar autenticado no sistema.

Fluxo Principal:

1. O leitor acessa a página de criação de comunidades.

2. O sistema exibe um formulário onde o usuário pode preencher os seguintes campos: Nome da Comunidade, Descrição, Objetivo, Tipo (Pública ou Privada).
3. O leitor preenche o formulário com as informações desejadas.
4. O leitor clica no botão "Criar Comunidade".
5. O sistema valida as informações inseridas e, se tudo estiver correto, cria a comunidade no banco de dados.
6. O administrador (usuário que criou a comunidade) é automaticamente adicionado à comunidade com status "aceito".
7. O sistema exibe uma mensagem de sucesso informando que a comunidade foi criada com sucesso.
8. O sistema atualiza a lista de comunidades disponíveis para o usuário.

Fluxos Alternativos:

- **FA1 - Falha na Criação da Comunidade:**

1. Caso ocorra algum erro ao criar a comunidade (por exemplo, se os dados não forem válidos ou o servidor estiver inacessível), o sistema exibe uma mensagem de erro informando que houve um problema na criação da comunidade.

Pós-condições:

1. A comunidade é criada com sucesso.

6.4.2.12. Caso de uso - Acessar Comunidade

Caso de Uso: Acessar Comunidade (CSU12)

Descrição:

Ator Primário: Leitor

Atores Secundários: Administrador, Voluntários

Pré-condições:

- O usuário deve estar autenticado.
- O status do usuário em relação à comunidade deve estar atualizado.

Fluxo Principal:

1. O usuário visualiza uma lista de comunidades na página "Comunidades".
2. Cada comunidade possui um botão para "Entrar", "Solicitar Entrada" ou "Acessar", dependendo do status do usuário em relação a essa comunidade.
3. O status do usuário é determinado pela interação anterior (se ele já for solicitado ou se já foi aceito).
4. O usuário clicou no botão correspondente à ação desejada:
 - **Entrar:** Se o status do usuário for "não inscrito" em uma comunidade pública, ele pode entrar diretamente.
 - **Solicitar entrada:** Se o status de "não inscrito" em uma comunidade privada, ele envia uma solicitação para entrar.
 - **Solicitado:** Caso o status seja "pendente", o botão exibe "Solicitado" e não permite mais ações até que a solicitação seja respondida.
 - **Acessar:** Se o status for "aceito", o usuário é redirecionado para a página da comunidade.
5. O sistema verifica o status do usuário e executa a ação a seguir:
 - Se o status for "não inscrito", ele executa a ação "Entrar" ou "Solicitar Entrada".
 - Se o status for "pendente", ele notificará o usuário de que a solicitação está aguardando aprovação.
 - Se o status for "aceito", ele redireciona o usuário para a página da comunidade.

Fluxos Alternativos:

- **FA1 - Comunidade Pública:**

1. Se for uma comunidade pública, o botão "Entrar" estará disponível para os usuários com o status "não inscrito".
2. O usuário clica em "Entrar" e é adicionado à comunidade diretamente, com o status atualizado para "aceito".
3. O usuário é redirecionado para a página da comunidade.

- **FA2 - Comunidade Privada:**

1. Se a comunidade for privada e o status do usuário para "não inscrito", o botão exibirá "Solicitar Entrada".
2. O usuário clicou em "Solicitar Entrada" e a solicitação é enviada para aprovação.
3. O status é atualizado para "pendente", e o usuário aguarda a aprovação do administrador.

- **FA3 - Solicitação Pendente:**

1. Se o status for "pendente", o botão exibe "Solicitado".
2. O usuário não pode realizar mais ações até que uma solicitação seja aprovada ou rejeitada.

- **FA4 - Status Aceito:**

1. Se o status for "aceito", o botão exibe "Acessar".
2. O usuário clica em "Acessar" e é redirecionado para a página da comunidade.

Pós-condições:

1. O status do usuário é atualizado de acordo com a ação realizada (aceito, pendente ou não inscrito).
2. O usuário é redirecionado para a página da comunidade se a entrada for bem-sucedida ou se ele já for aceito.

6.4.2.13. Caso de uso - Comunidade

Caso de Uso: Comunidade (CSU13)

Descrição: Este caso de uso permite ao usuário visualizar os detalhes de uma comunidade, interagir com o progresso de leitura dos membros, visualizar gráficos sobre a participação na comunidade, comentar nas discussões e, se for administrador, gerenciar os usuários da comunidade.

Ator Primário: Leitor

Atores Secundários: Administrador, Voluntários

Pré-condições:

- O usuário deve estar autenticado.
- O usuário deve ter acesso à comunidade (pode ser selecionado se ele for membro ou administrador).
- O usuário deve ter permissões permitidas para acessar ou interagir com a comunidade.

Fluxo Principal:

1. O usuário acessou a página de detalhes de uma comunidade, com base no seu ID.
2. O sistema verifica o status do usuário, ele é administrador, e define uma visualização conforme as instruções.
3. O sistema exibe os detalhes da comunidade, como nome, descrição e objetivo.
4. O sistema carrega e exibe informações sobre o progresso da leitura dos membros.
5. O sistema exibe gráficos sobre estatísticas de idade e participação dos membros da comunidade.
6. O usuário pode interagir com a seção de comentários, adicionando novos comentários ou visualizando os existentes.
7. Caso o usuário seja administrador, ele poderá gerenciar os usuários da comunidade, como aceitar ou rejeitar solicitações de membros.
8. O sistema permite que o usuário adicione progresso de leitura ou interaja com a comunidade.

9. O sistema atualiza os dados em tempo real, como progresso e comentários, conforme a interação do usuário.

Fluxos Alternativos:

- **FA1 - Usuário sem permissão de administrador:**
 1. Se o usuário não for um administrador, ele não verá a seção "Gerenciar Comunidade" e suas interações serão limitadas à visualização de dados e comentários.
- **FA2 - Erro no carregamento de dados:**
 1. Caso haja um erro ao carregar os dados da comunidade (progresso, comentários ou estatísticas), o sistema exibirá uma mensagem de erro informando o usuário e tentará realizar nova tentativa automaticamente.
- **FA3 - Dados de progresso ou estatísticas ausentes:**
 1. Caso os dados de progresso ou estatísticas de idade não sejam encontrados, o sistema exibirá uma mensagem informando que não há dados disponíveis para exibição, mas permitirá que o usuário continue interagindo nas outras áreas da página.

Pós-condições:

1. O sistema exibe os dados atualizados da comunidade (mensagem, progresso de leitura, gráficos e estatísticas).
2. O status de progresso do usuário ou de qualquer outra interação é atualizado em tempo real.
3. Se o usuário for administrador, ele pode gerenciar as interferências dos usuários.

6.4.2.14. Caso de uso - Relatórios

Caso de Uso: Relatórios (CSU014)

Descrição: Este caso de uso permite que o Administrador visualize relatórios detalhados sobre a biblioteca, como o número de livros emprestados, faixa etária dos usuários, e empréstimos atrasados. O Administrador pode configurar o período para

a geração dos relatórios, sendo possível visualizar gráficos e tabelas com informações relevante.

Ator Primário: Administrador

Pré-condições:

- O Administrador deve estar autenticado e autorizado a visualizar relatórios.

Fluxo Principal:

1. O administrador acessou a página de relatórios.
2. O sistema exibe um formulário de filtro de dados, com campos para selecionar os dados de início e fim e um período predefinido (30 dias, 60 dias, 90 dias, ou 1 ano).
3. O administrador seleciona um período ou define um intervalo de dados personalizado.
4. O sistema envia uma requisição para a API, solicitando os dados dos relatórios de acordo com o período selecionado.
5. O sistema recebe os dados da API e exibe os relatórios.
6. O administrador pode visualizar os dados de forma interativa, filtrando os resultados por dados ou período.

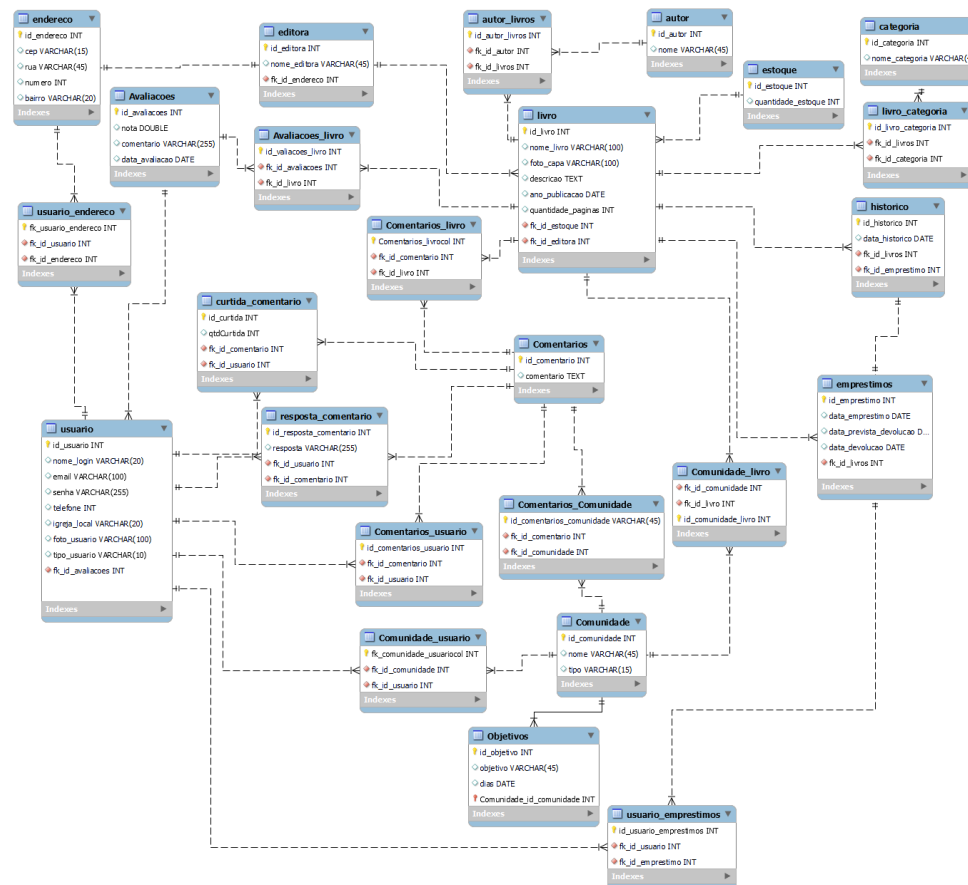
Pós-condições:

1. O Administrador visualiza os relatórios solicitados.

6.4.3. Diagrama de Banco de Dados

A figura abaixo representa o diagrama de banco de dados da aplicação pesquisada neste trabalho.:

Figura 16: Diagrama de Banco de Dados



Fonte: Próprio autor

6.4.4. Diagrama de classes

A figura abaixo representa o diagrama de casos de classe da aplicação pesquisada neste trabalho.

7. RESULTADO E DISCUSSÕES

O software foi desenvolvido e encontra-se em funcionamento, atendendo aos requisitos definidos na fase de planejamento

7.1. Aplicação *front-end*

A aplicação front-end foi desenvolvida utilizando Next.js, um framework baseado em React.js, que possibilita a construção de interfaces dinâmicas, eficientes e escaláveis.

Durante o processo de implementação, alguns desafios técnicos foram encontrados, como a necessidade de garantir responsividade em múltiplos dispositivos. Esses pontos foram solucionados por meio do uso de CSS Flexbox/Grid e outras classes do CSS.

É importante destacar que as funcionalidades implementadas respondem diretamente à problemática levantada. O controle de empréstimos, antes realizado de forma manual e suscetível a erros, agora é automatizado, impedindo duplicidade e fornecendo relatórios em tempo real. Isso contribui para a diminuição de falhas humanas e aumento da eficiência no gerenciamento da biblioteca.

Quanto às perspectivas futuras, o sistema poderá incorporar novas funcionalidades, como envio de notificações automáticas para devoluções, dashboards de acompanhamento de leitura por comunidade e integração com serviços de nuvem para maior escalabilidade. Essas evoluções pretendem fortalecer ainda mais a experiência do usuário e ampliar o impacto da solução no contexto da gestão bibliotecária

7.1.1. Página de login

A página de login é em essência a porta de entrada para o sistema, pois 99% das funcionalidades do sistema dependem de se estar logado no mesmo para ver o nível de usuário e seus respectivos acessos. Essa página contém dois inputs sendo um para inserir o usuário e o outro a senha, como pode ser visto na figura 18.

Figura 18: Página de login

HELENA
RAMAZZOTTE
BIBLIOTECA

Login
Bem-vindo de volta!

Informe os dados para Login

Usuário

Senha

Entrar Perdeu a senha?

Cadastre-se

Ainda não possui conta? Cadastre-se no site.

Cadastre-se

Fonte: Próprio Autor

Se o usuário tentar entrar sem preencher os campos obrigatórios, o sistema exibe um alerta avisando que os campos não podem ficar vazios. Isso evita tentativas inválidas de acesso, como pode ser visto na figura 19.

Figura 19: Página de login validação de campos vazios

HELENA
RAMAZZOTTE
BIBLIOTECA

Login
Bem-vindo de volta!

Informe os dados para Login

Usuário

Senha

Os campos não podem ser vazios

Entrar Perdeu a senha?

Cadastre-se

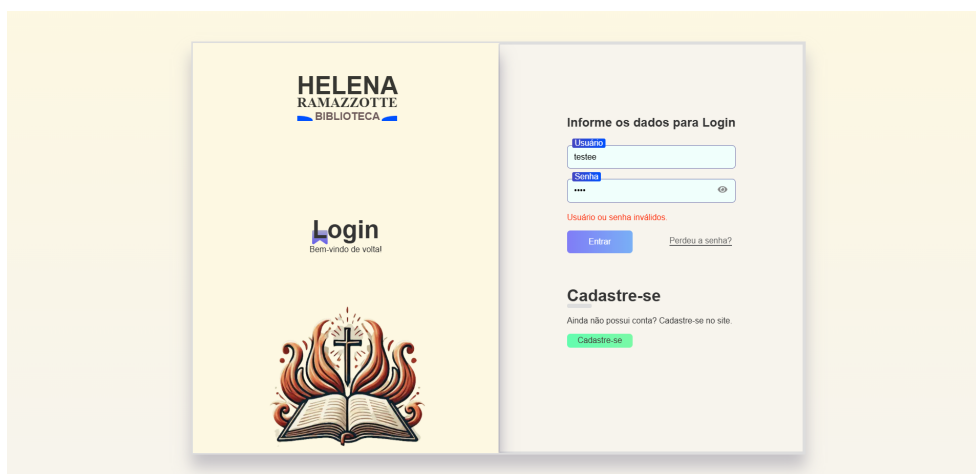
Ainda não possui conta? Cadastre-se no site.

Cadastre-se

Fonte: Próprio Autor

Caso o usuário digite um login ou senha incorretos, o sistema exibe uma mensagem de erro informando que as credenciais estão erradas, como pode ser visto na figura 20.

Figura 20: Página de login validação de usuário e senha



Fonte: Próprio Autor

Se ele não tiver uma conta, pode clicar em “cadastre-se”. Ao clicar em cadastre-se o usuário é levado a página de cadastre-se.

Se o usuário não lembrar a sua senha, ele pode pedir uma nova senha clicando em “Perdeu a senha?”. Ao clicar em esqueceu a senha o usuário é levado a página de perdeu a senha.

7.1.2. Página de cadastre-se

Essa tela permite que novos usuários criem suas contas no sistema. Aqui, o usuário deve preencher informações como nome, e-mail, senha, telefone, sexo e se faz parte dos membros da igreja para criar sua conta, como pode ser visto na figura 21.

Figura 21: Página de cadastre-se aba dados de usuário

The image shows a web form titled "Informe os dados" with two tabs: "Usuário" (selected) and "Endereço". The "Usuário" tab contains the following fields: "Usuário", "Email", "Senha", "Confirma a Senha", "Telefone", "Sexo" (with a dropdown menu), and a checkbox "Membro IMUB?". Below these fields are buttons for "Cadastar" and "Faça Login". A link "Já possui conta? Faça Login." and a "Login" button are also present. The right side of the page features the logo for "HELENA RAMAZZOTTE BIBLIOTECA", the slogan "Cadastre-se Junte-se a nós!", and a logo of an open book with a cross.

Fonte: Próprio Autor

Na segunda etapa do cadastro, o usuário precisa informar seu endereço, ao preencher o cep os dados de rua, bairro, cidade e estado são preenchidos automaticamente para facilitar o usuário, ele também deve informar a data de nascimento e uma foto de perfil. Essa aba separa os dados pessoais dos dados de localização para facilitar o preenchimento, como pode ser visto na figura 22.

Figura 22: Página de cadastre-se aba de endereço do usuário

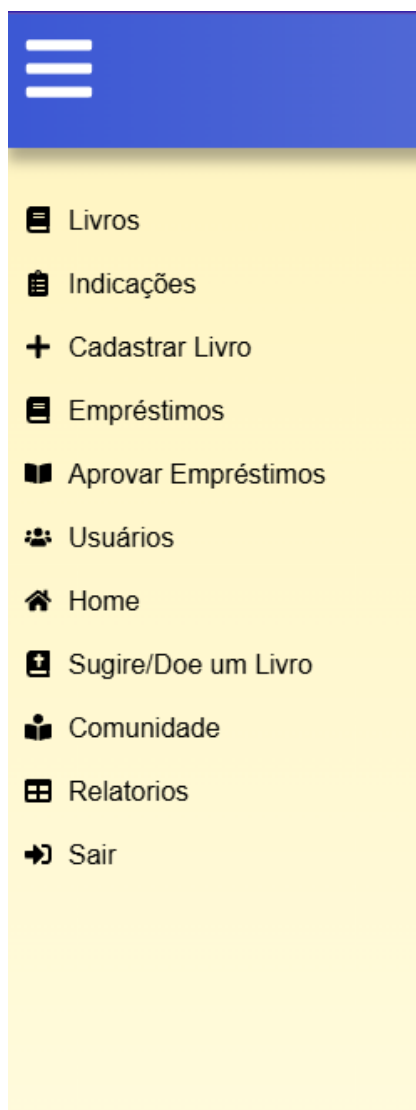
The image shows the same web form as in Figure 21, but with the "Endereço" tab selected. The "Usuário" tab is now disabled. The "Endereço" tab contains the following fields: "CEP", "Rua", "Bairro", "Cidade", "Estado", "Número", "Data Nascimento" (with a date picker), and "Foto Perfil" (with a file upload button). Below these fields are buttons for "Cadastar" and "Faça Login". A link "Já possui conta? Faça Login." and a "Login" button are also present. The right side of the page features the logo for "HELENA RAMAZZOTTE BIBLIOTECA", the slogan "Cadastre-se Junte-se a nós!", and a logo of an open book with a cross.

Fonte: Próprio Autor

7.1.3. Menu Lateral

O menu lateral do administrador contém todas as opções do sistema, incluindo o gerenciamento de usuários, livros e relatórios. Ele tem acesso total às funcionalidades, como pode ser vista na figura 23.

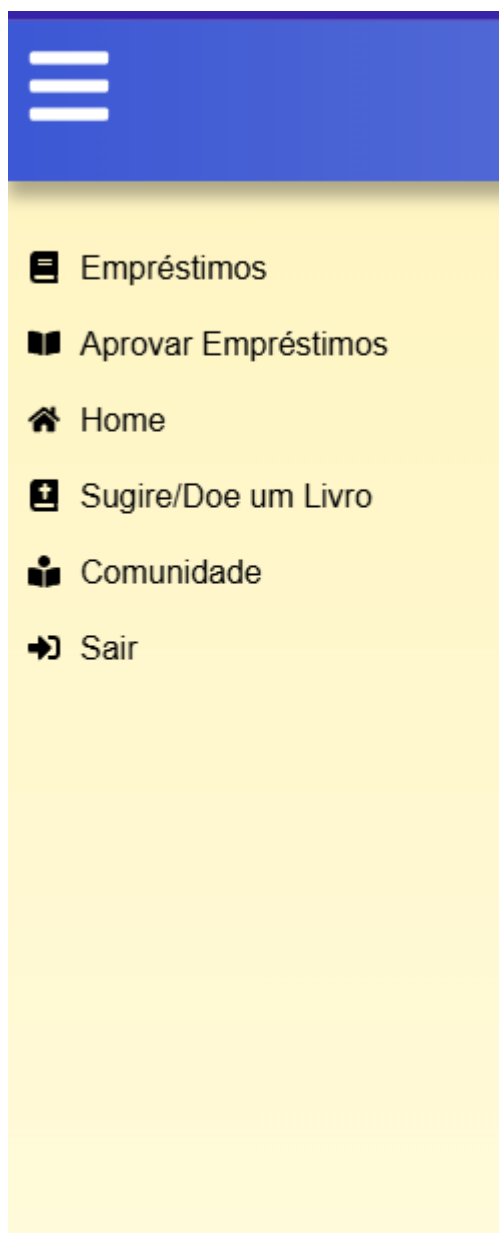
Figura 23: Página de menu lateral usuário: Administrador



Fonte: Próprio Autor

O menu lateral para voluntários tem opções reduzidas em relação ao administrador. Ele pode ajudar no controle de empréstimos, mas não tem acesso a opções mais avançadas, como pode ser visto na figura 24.

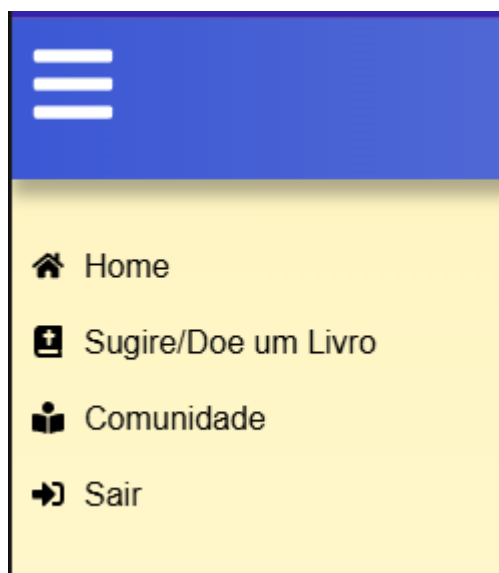
Figura 24: Página de menu lateral usuário: Voluntário



Fonte: Próprio Autor

O menu lateral do leitor é mais simples, focado na navegação pelos livros, comunidades, doação ou sugestão de livros e histórico de empréstimos, como pode ser visto na figura 25.

Figura 25: Página de menu lateral usuário: Leitor

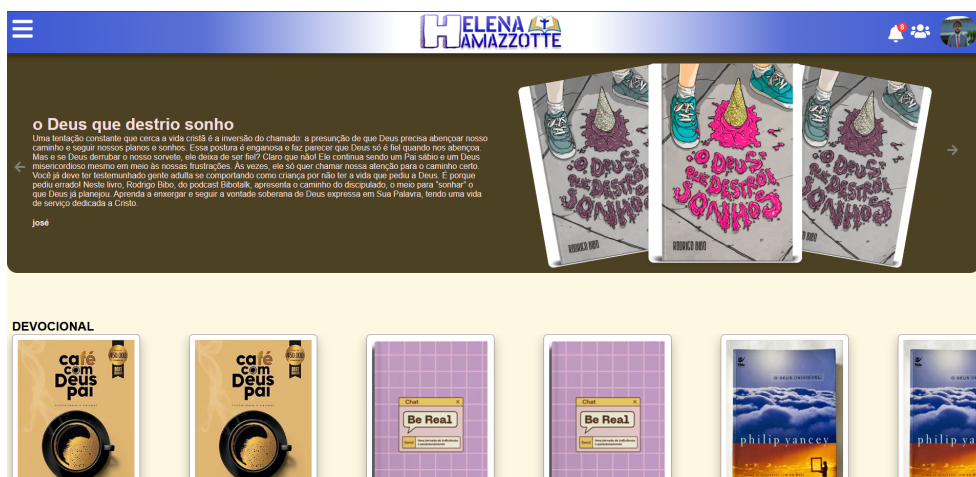


Fonte: Próprio Autor

7.1.4. Página de home

Essa é a tela principal do sistema. Aqui, os usuários podem visualizar os livros disponíveis, separados por categorias, facilitando a navegação e a busca por novas leituras, como pode ser visto na figura 26. Ao clicar sobre um livro o usuário é redirecionado para a página de detalhes do livro

Figura 26: Página de home



Fonte: Próprio Autor

7.1.5. Página de detalhes do livro

Ao selecionar um livro, o usuário pode ver informações completas sobre ele, como título, autor, descrição, comentários e avaliações de outros leitores, como pode ser visto na figura 27.

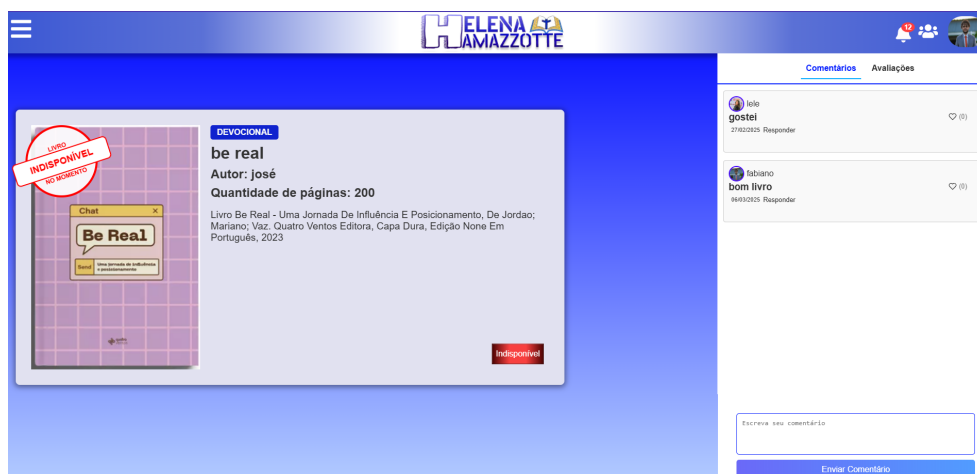
Figura 27: Página de detalhes do livro



Fonte: Próprio Autor

Se um livro não estiver disponível para empréstimo, a tela exibe um aviso indicando que ele está indisponível no momento e o botão de reserva fica indisponível e desabilitado, como pode ser visto na figura 28.

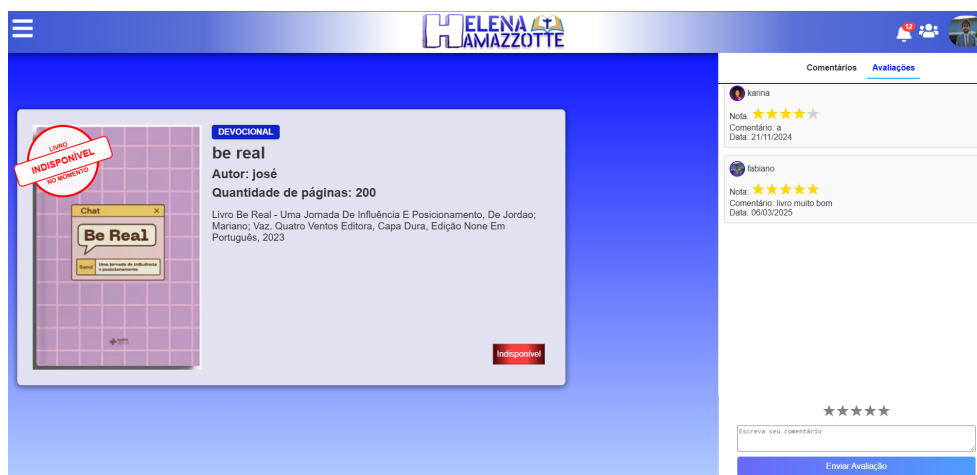
Figura 28: Página de detalhes do livro se o livro estiver indisponível



Fonte: Próprio Autor

Os usuários podem avaliar os livros e deixar comentários sobre suas experiências de leitura nesta aba de avaliações, como pode ser visto na figura 29.

Figura 29: Página de detalhes do livro aba avaliações



Fonte: Próprio Autor

7.1.6. Página de cadastrar livro

Nesta tela o usuário poderá cadastrar um novo livro na biblioteca, como pode ser visto nas figuras 30 e 31

Figura 30: Página de cadastrar livro parte1



A captura de tela mostra a interface de usuário para o cadastro de um livro. No topo, há o logotipo 'HELENA AMAZZOTTE' e o título 'Cadastrar Livro'. Abaixo, há uma aba 'Dados do Livro' e um campo 'Editora'. O formulário principal contém os seguintes campos: 'Nome do Livro', 'Descrição', 'Ano de Publicação', 'Quantidade de Páginas', 'Quantidade em Estoque' (com uma seta para baixo), e 'Categoria' (menu suspenso). Abaixo desses campos, há dois botões de upload de imagem: 'Car. Logo' e 'Car. Capa', cada um com uma caixa de seleção vazia. O fundo da página é amarelo claro.

Fonte: Próprio Autor

Figura 31: Página de cadastrar livro parte2



A segunda captura de tela mostra a continuação do formulário de cadastro. O campo 'Categoria' é um menu suspenso. Abaixo dele, há os botões de upload de imagem 'Car. Logo' e 'Car. Capa'. Um campo de texto para 'Sub-Categorias' é seguido por um botão verde 'Adicionar mais Subcategorias'. Abaixo, há um campo 'Autor' com um menu suspenso, um botão verde 'Adicionar Autor', e um campo 'Capa do Livro' com o texto 'Escolher arquivo | Nenhum arquivo selecionado'. No final, há um botão azul 'Salvar'.

Fonte: Próprio Autor

7.1.7. Página de aprovar empréstimo

Os administradores ou voluntários podem aprovar ou recusar solicitações de empréstimos nesta tela, garantindo um controle eficiente da circulação dos livros, como pode ser visto na figura 32.

Figura 32: Página de aprovar empréstimo



Fonte: Próprio Autor

7.1.8. Página de empréstimos

Essa página exibe todos os livros que estão emprestados no momento, mostrando quem pegou cada um e a data prevista para devolução, e se a data prevista de devolução tiver em dia a borda fica verde, se faltar menos de 10 dias a borda fica amarela e se tiver vencida fica vermelha assim facilitando o usuário ver o status de entrega do livro, como pode ser visto na figura 33.

Figura 33: Página empréstimos



Fonte: Próprio Autor

7.1.9. Página de usuário

Essa tela mostra todos os usuários cadastrados no sistema e ao clicar sobre o usuário o administrador é redirecionado para a página de detalhes do usuário.

Figura 34: Página de usuário

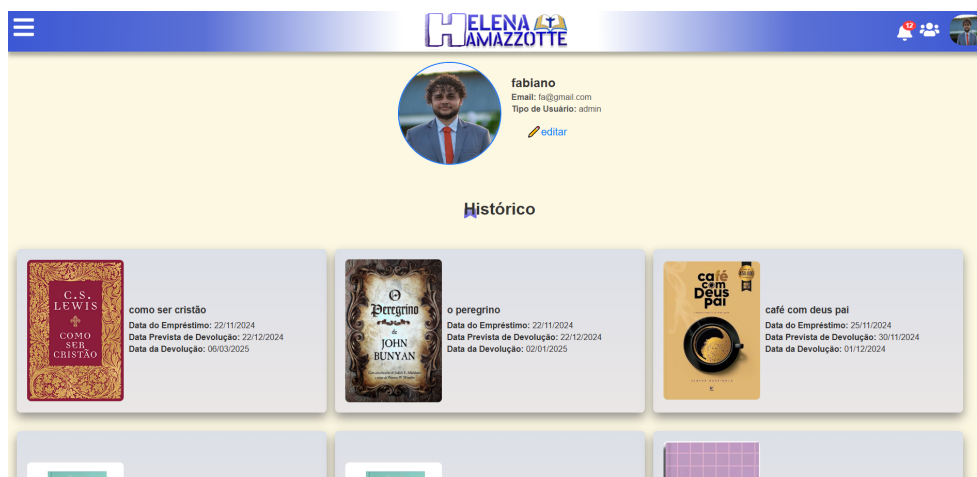


Fonte: Próprio Autor

7.1.10. Página de detalhes do usuário

O administrador pode acessar os detalhes de um usuário específico, verificando o histórico, os dados e o nível de acesso dentro do sistema, como pode ser visto na figura 35.

Figura 35: Página de detalhes do usuário



Fonte: Próprio Autor

O administrador poderá alterar o nível de acesso de um usuário, como pode ser visto na figura 36.

Figura 36: Página de detalhes do usuário mudar nível de usuário

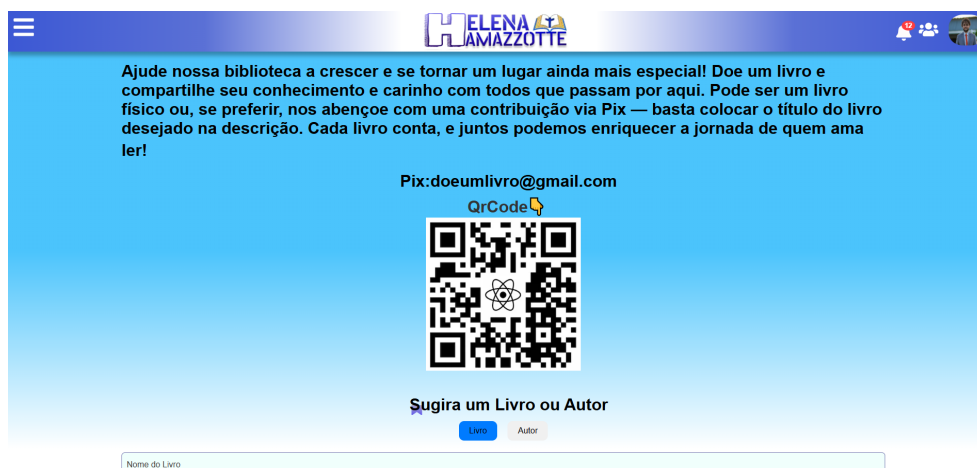


Fonte: Próprio Autor

7.1.11. Página de de sugira ou doe um livro

O usuário verá uma mensagem de explicação sobre doar um livro e terá um qr code com pix para caso ele queira fazer uma doação.

Figura 37: Página de sugira ou doe um livro parte 1



Fonte: Próprio Autor

O usuário poderá sugerir um livro ou autor, deixando o nome do livro, descrição e o porquê daquela indicação deveria estar na biblioteca.

Figura 38: Página de sugira ou doe um livro parte 2



Fonte: Próprio Autor

7.1.12. Página de criar comunidade

Os usuários podem criar comunidades de leitura nesta página, definindo o nome, a descrição e o tipo de comunidade (pública ou privada). E abaixo ver a lista de comunidades já criadas, se for pública o botão ficará como “entrar” e o usuário poderá entrar diretamente na comunidade e se for privada ele tem que solicitar acesso e o administrador da comunidade aceitar o convite.

Figura 39: Página de criar comunidade

A imagem mostra a interface de usuário para criar uma comunidade de leitura. No topo, há o logotipo 'HELENA AMAZZONE' e ícones de notificação e perfil. O formulário 'Criar Comunidade' possui os seguintes campos: 'Nome da Comunidade', 'Descrição', 'Objetivo', 'Tipo (Pública)' e um botão 'Criar Comunidade'. Abaixo, a seção 'Comunidades' exibe quatro cartões:

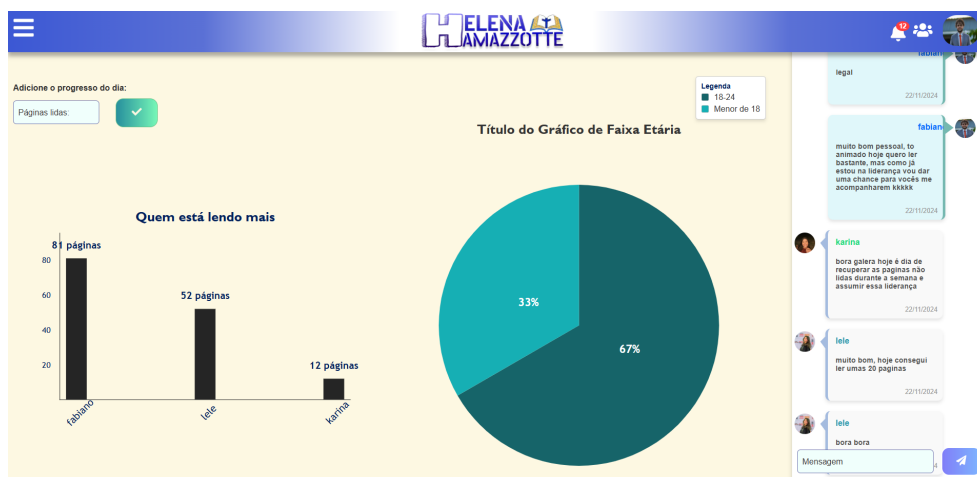
Nome da Comunidade	Descrição	Ação
leitura biblica	ler a biblia em 75 dias	Acessar
leitura do livro tal	ler o livro	Entrar
ler teste	ler a biblia em 75 dias	Solicitar Entrada
IIIIIIII	ainda tá em teste	Acessar

Fonte: Próprio Autor

7.1.13. Página de comunidade

Os usuários diariamente deverá adicionar o seu progresso de leitura e isso cria um ranking conforme quem está lendo mais, os usuários podem se comunicar por meio de mensagens, como podem ser visto na figura 40.

Figura 40: Página de comunidade

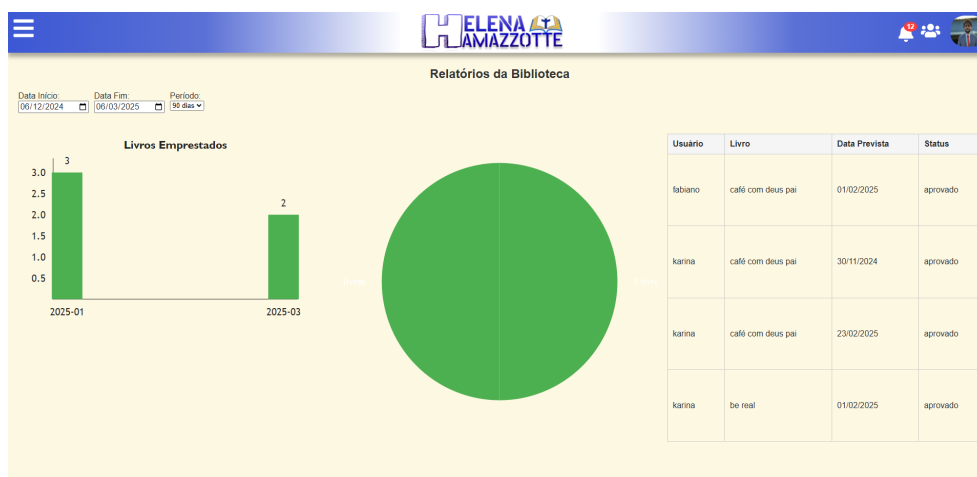


Fonte: Próprio Autor

7.1.14. Página de relatórios

Os administradores podem acessar relatórios detalhados sobre os empréstimos, isso ajudará os administradores a tomarem decisões sobre novos livros, estratégias de leituras para os usuários e muito mais como pode ser visto na figura 41.

Figura 41: Página de relatórios

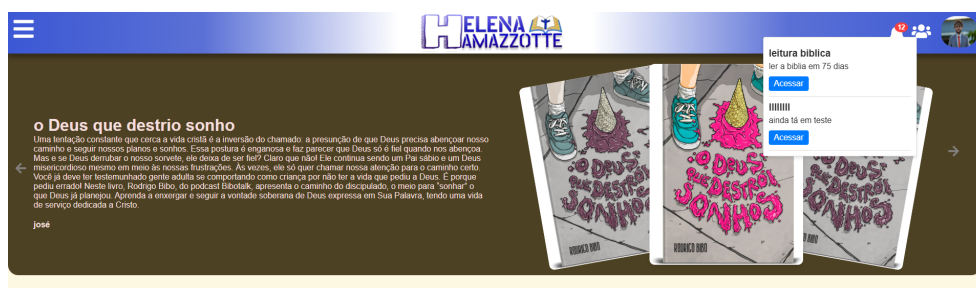


Fonte: Próprio Autor

7.1.15. Página de acesso rápido à comunidade

Os usuários poderão acessar as comunidades que ele faz parte de maneira rápida diretamente da navbar, como pode ser visto na figura 42.

Figura 42: Página de acesso rápido à comunidade



Fonte: Próprio Autor

7.1.16. Página de perfil do usuário

O usuário verá seus dados pessoais cadastrados e todo o seu histórico de livros que ele pegou emprestado, como pode ser visto na figura 43.

Figura 43: Página de perfil do usuário



Fonte: Próprio Autor

7.1.17. Página de notificação

Os usuários poderão acessar as notificações diretamente da navbar, ele verá os seus avisos e notificações importantes, como prazos de devolução e aprovações de empréstimos, como pode ser visto na figura 44

Figura 44: Página de notificações

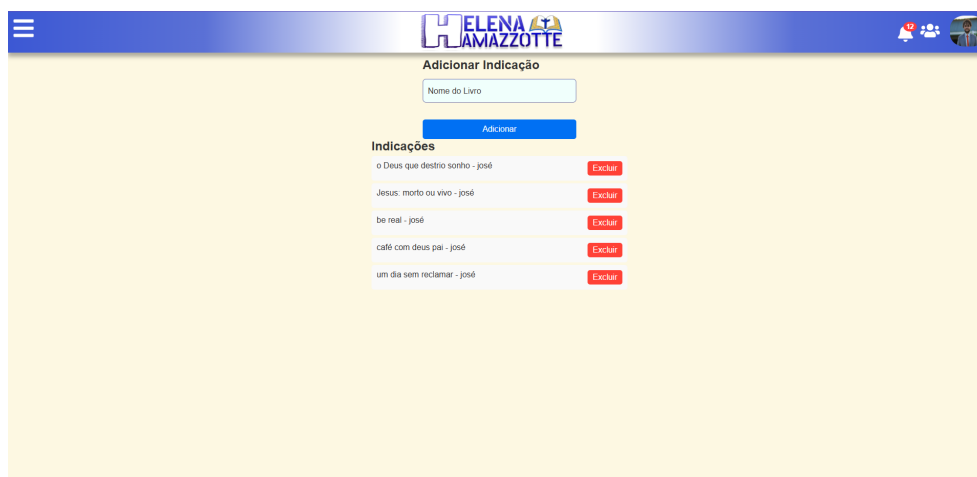


Fonte: Próprio Autor

7.1.18. Página de indicações semanais

Toda semana, os administradores farão a recomendação de cinco livros para os leitores, essas recomendações ficarão disponíveis para os leitores na tela de home.

Figura 45: Página de indicações semanais

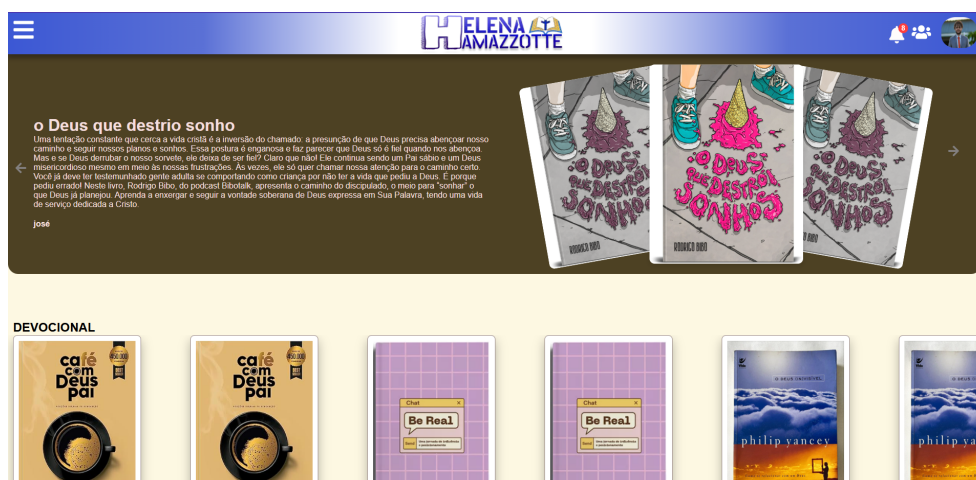


Fonte: Próprio Autor

7.1.19. Página de indicações semanais(home)

Na tela inicial de home os leitores verão as indicações da semana, essas indicações ficam passando a cada 10 segundos e ele pode passar as indicações para o lado ou voltar uma indicação caso ele queira saber mais sobre a indicação, como pode ser visto na figura 46.

Figura 46: Página de indicações semanais



Fonte: Próprio Autor

8. CRONOGRAMA

Atividade	Período
Estudo bibliográfico e fundamentação teórica	Outubro/2024 - novembro/2024
Modelagem de casos de uso	Novembro/2024 - dezembro/2024
Modelagem do banco de dados (conceitual e lógico)	Novembro/2024 - dezembro/2024
Implementação do banco de dados (físico, tabelas, relacionamentos e índices)	Dezembro/2024-janeiro/2025
Desenvolvimento da API (endpoints de usuários, livros, empréstimos, autenticação e segurança)	Janeiro/2025 - março/2025
Configuração do servidor de arquivos e imagens (upload, armazenamento e integração com API)	Março/2025
Desenvolvimento do Front-End – Estrutura inicial e navegação	Março/2025 - abril/2025
Desenvolvimento do Front-End – Interfaces de cadastro e consulta	Abril/2025 - maio/2025
Desenvolvimento do Front-End – Integração com a API e design responsivo	Maió/2025 - junho/2025

configuração do PWA	Junho/2025
Testes unitários	Julho/2025
Validação com usuários e ajustes finais	Julho/2025
Deploy e Finalização do Projeto	Agosto/2025

REFERÊNCIAS

AQUINO, Tainá. Front-End: o que é, linguagens e frameworks? UDS, 2024. Disponível em: <https://uds.com.br/blog/front-end-o-que-e-linguagens-frameworks/>. Acesso em: 02 fev. 2025.

AWS (SERVIÇOS WEB AMAZON). O que é uma API RESTful? 2024. Disponível em: <https://aws.amazon.com/what-is/restful-api/>. Acesso em: 01 fev. 2025.

BASTOS, Athena. Metodologias Ágeis: o que são e como aplicá-las na sua empresa? Alura, 2023. Disponível em: <https://www.alura.com.br/empresas/artigos/metodologias-ageis>. Acesso em: 04 fev. 2025.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: guia do usuário. 2ª ed. Rio de Janeiro: Campus, 2005.

CLEMENTE, Paulo. Dominando Node.js: um guia prático para iniciantes. Rocketseat, 2024. Disponível em: <https://blog.rocketseat.com.br/dominando-node-js-um-guia-pratico-para-iniciantes/>. Acesso em: 01 fev. 2025.

CLEMENTE, Paulo. Introdução ao Next.js 13. Rocketseat, 2024. Disponível em: <https://blog-rocketseat.rocketseat.com.br/blog/artigos/post/introduction-nextjs-13>. Acesso em: 03 fev. 2025.

CNN BRASIL. Scrum: entenda o que é e como essa metodologia pode gerar produtividade. 2024. Disponível em: <https://www.cnnbrasil.com.br/tecnologia/scrum/>. Acesso em: 04 fev. 2025.

FAMÍLIA CRISTÃ. Nossa História. Disponível em: <https://www.livrariasfiliacrista.com.br/>. Acesso em: 29 jan. 2025.

FARROCO, Leonardo de Oliveira; STEFANO, Ercilia de; EBECKEN, Nelson Francisco Fávilla; PAULO, Fernanda Santana de; NOSOLINE, Sumaya Mário. O paradigma funcional no desenvolvimento Front-End: oportunidades e desafios. Brazilian Journal of Development, v. 6, n. 2, 2020. DOI: <10.34117/bjdv6n2-083>.

HILMAN, M. Conhecendo TypeScript no Front-End. Alura, 2024. Disponível em: <https://www.alura.com.br/artigos/conhecendo-typescript-front-end>. Acesso em: 30 jan. 2025.

JAISON. Introdução ao JavaScript. DevMedia, 2012. Disponível em: <https://www.devmedia.com.br/introducao-ao-javascript/25548>. Acesso em: 29 jan. 2025.

JESUSCOPY. Loja JesusCopy - Livros Cristãos e Materiais para Crescimento Espiritual. Disponível em: <https://loja.jesuscop.com/>. Acesso em: 29 jan. 2025.

JUNGTHON, S.; GOULART, C. Paradigmas de programação e suas aplicações em linguagens modernas. Brazilian Journal of Development, v. 6, n. 2, p. 6464-6475, 2020. Disponível em:

<https://ojs.brazilianjournals.com.br/ojs/index.php/BRJD/article/view/6753/5954>. Acesso em: 27 jan. 2025.

JWT.IO. Introduction to JSON Web Tokens. 2024. Disponível em: <https://jwt.io/introduction> . Acesso em: 02 fev. 2025.

LUZ DA SABEDÓRIA. Biblioteca Cristã: Recursos e Importância . 2024. Disponível em: <https://luzdasabedoria.com.br/glossario/biblioteca-crista-recursos-e-importancia/> . Acesso em: 29 jan. 2025.

MENEZES, Elizabeth. O que é Desenvolvimento Web? Estratégia Concursos, 2024. Disponível em: <https://www.estrategiaconcursos.com.br/blog/desenvolvimento-web/> . Acesso em: 30 jan. 2025.

MICROSOFT. Documentação TypeScript. Disponível em : <https://www.typescriptlang.org/> . Acesso em: 30 jan. 2025.

MOZILLA DEVELOPER NETWORK (MDN). Documentação de APIs REST. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Glossary/REST> . Acesso em: 01 fev. 2025.

MOZILLA DEVELOPER NETWORK (MDN). CSS Básico. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Getting_started/Your_first_website/Styling_the_content . Acesso em: 03 fev. 2025.

MOZILLA DEVELOPER NETWORK (MDN). HTML: HyperText Markup Language. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML> . Acesso em: 03 fev. 2025.

MOZILLA DEVELOPER NETWORK (MDN). Documentação JavaScript. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript> . Acesso em: 29 jan. 2025.

MOZILLA DEVELOPER NETWORK (MDN). Introdução ao Node.js e Express. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Extensions/Server-side/Express_Nodejs/Introduction . Acesso em: 01 fev. 2025.

MOURA, Beatriz; HENRIQUE, Matheus. O que é HTML e suas tags (Parte 1): Estrutura Básica. Alura, 2024. Disponível em: <https://www.alura.com.br/artigos/o-que-e-html-suas-tags-parte-1-estrutura-basica> . Acesso em: 03 fev. 2025.

NASCIMENTO, Felipe. Next.js: vantagens e como melhorar seu projeto. Alura, 2022. Disponível em: <https://www.alura.com.br/artigos/next-js-vantagens> . Acesso em: 03 fev. 2025.

NEVES, V. O que é TypeScript? Alura, 2024. Disponível em: <https://www.alura.com.br/artigos/o-que-e-typescript> . Acesso em: 29 jan. 2025.

NODE.JS. Documentação Oficial. Disponível em: <https://nodejs.org/pt> . Acesso em: 01 fev. 2025.

ORACLE. O que é um banco de dados? . Disponível em: <https://www.oracle.com/br/database/what-is-database/> . Acesso em: 03 fev. 2025.

PEREIRA, J.; SILVÉRIO, R. O que é JavaScript? 2024. Disponível em : <https://www.alura.com.br/artigos/javascript> . Acesso em: 30 jan. 2025.

PRESSMAN, Roger. Engenharia de Software. 6ª ed. São Paulo: Mc Graw Hill, 2006.

REACT. Documentação Oficial do React. Disponível em: <https://pt-br.legacy.reactjs.org/> . Acesso em: 03 fev. 2025.

SANTANA, A. O que é linguagem de programação? 2024. Disponível em: <https://www.alura.com.br/artigos/linguagem-programacao> . Acesso em: 27 jan. 2025.

SANTANA, Eduardo. O que é Back-End? Alura, 2024. Disponível em: <https://www.alura.com.br/artigos/backend> . Acesso em: 31 jan. 2025.

SEBESTA, R. W. Conceitos de Linguagens de Programação - 11.ed. [s.l.] Bookman Editora, 2018.

SITES GOOGLE. Biblioteca Cristã Digital . Disponível em: <https://sites.google.com/view/bibliotecacrist/home> . Acesso em: 29 jan. 2025.

STACK OVERFLOW. *Stack Overflow Developer Survey 2022*. Disponível em: <https://survey.stackoverflow.co/2022/#most-popular-technologies-language-learn> . Acesso em: 29 jan. 2025.

SOUZA, Robertson Douglas Castro; FLORIAN, Fabiana; DALLILO, Felipe Diniz. Uma análise de performance das linguagens de programação no processamento paralelo. Revista Científica Multidisciplinar Núcleo do Conhecimento. Ano 08, Ed. 12, Vol. 04, pp. 05-37, dez. 2023. ISSN: 2448-0959. Disponível em: <https://www.nucleodoconhecimento.com.br/engenharia-da-computacao/linguagens-de-programacao> . DOI: <10.32749/[nucleodoconhecimento.com.br/engenharia-da-computacao/linguagens-de-programacao](https://www.nucleodoconhecimento.com.br/engenharia-da-computacao/linguagens-de-programacao) >. Acesso em: 27 jan. 2025.

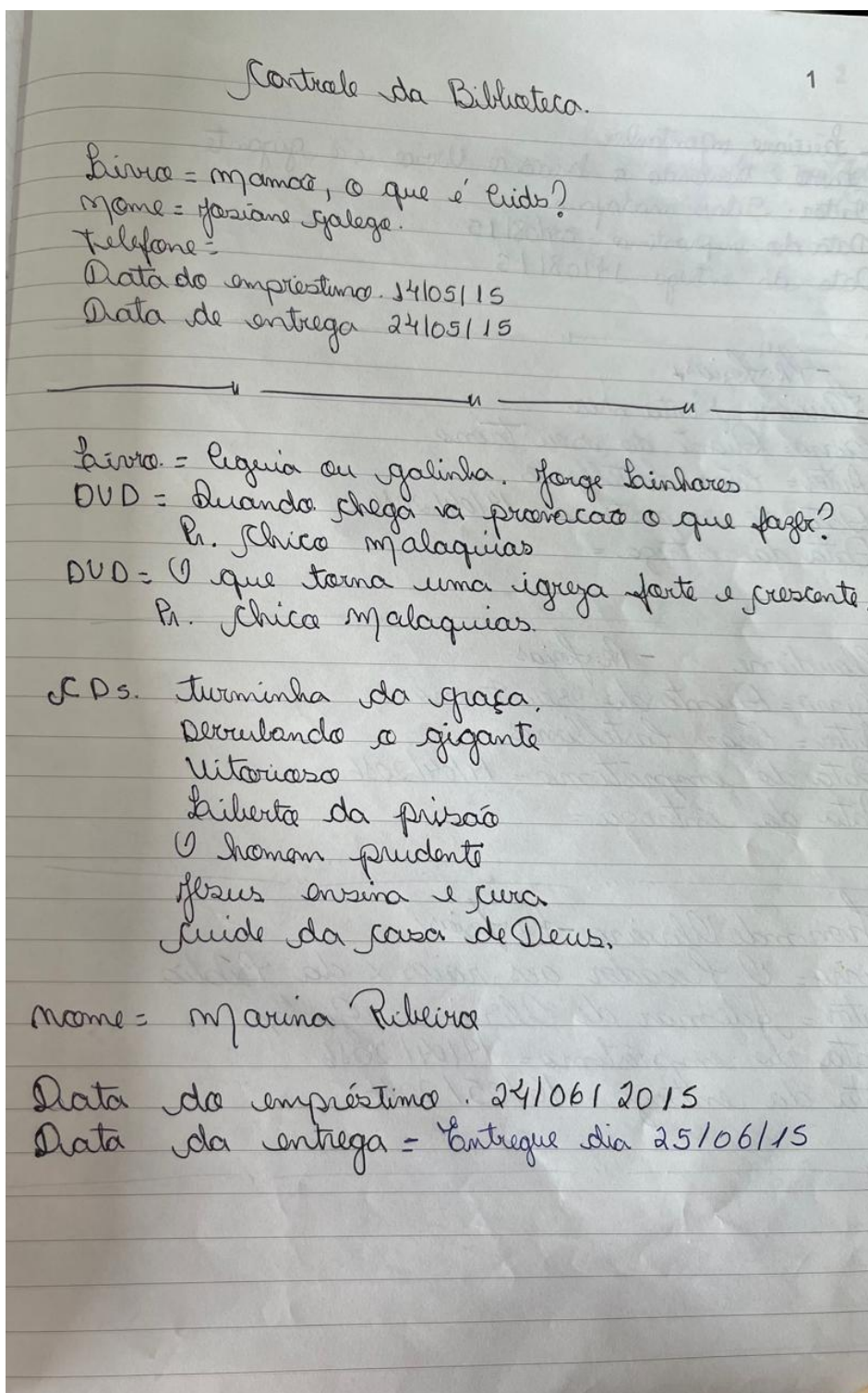
UNIVERSIDADE FEDERAL DE ALAGOAS (UFAL). O impacto potencial da biblioteca cristã na comunidade . 2023. Disponível em: https://www.repositorio.ufal.br/bitstream/123456789/14083/1/O%20impacto%20potencial%20da%20biblioteca%20crist%C3%A3%20na%20comunidade_uma%20an%C3%A1lise%20da%20comunidade%20crist%C3%A3%20da%20Igreja%20Evang%C3%A9lica%20Assembleia%20de%20Deus%20em%20Macei%C3%B3.pdf . Acesso em: 29 jan. 2025.

VALENÇA, Ranieri. Fundamentos do Desenvolvimento Web: uma revisão rápida. Dev.to, 2023. Disponível em: <https://dev.to/ranierivalenca/fundamentos-do-desenvolvimento-web-uma-rapida-revisao-15cj> . Acesso em: 30 jan. 2025.

VERCEL. Documentação Oficial. Disponível em : <https://nextjs.org/docs> . Acesso em: 03 fev. 2025

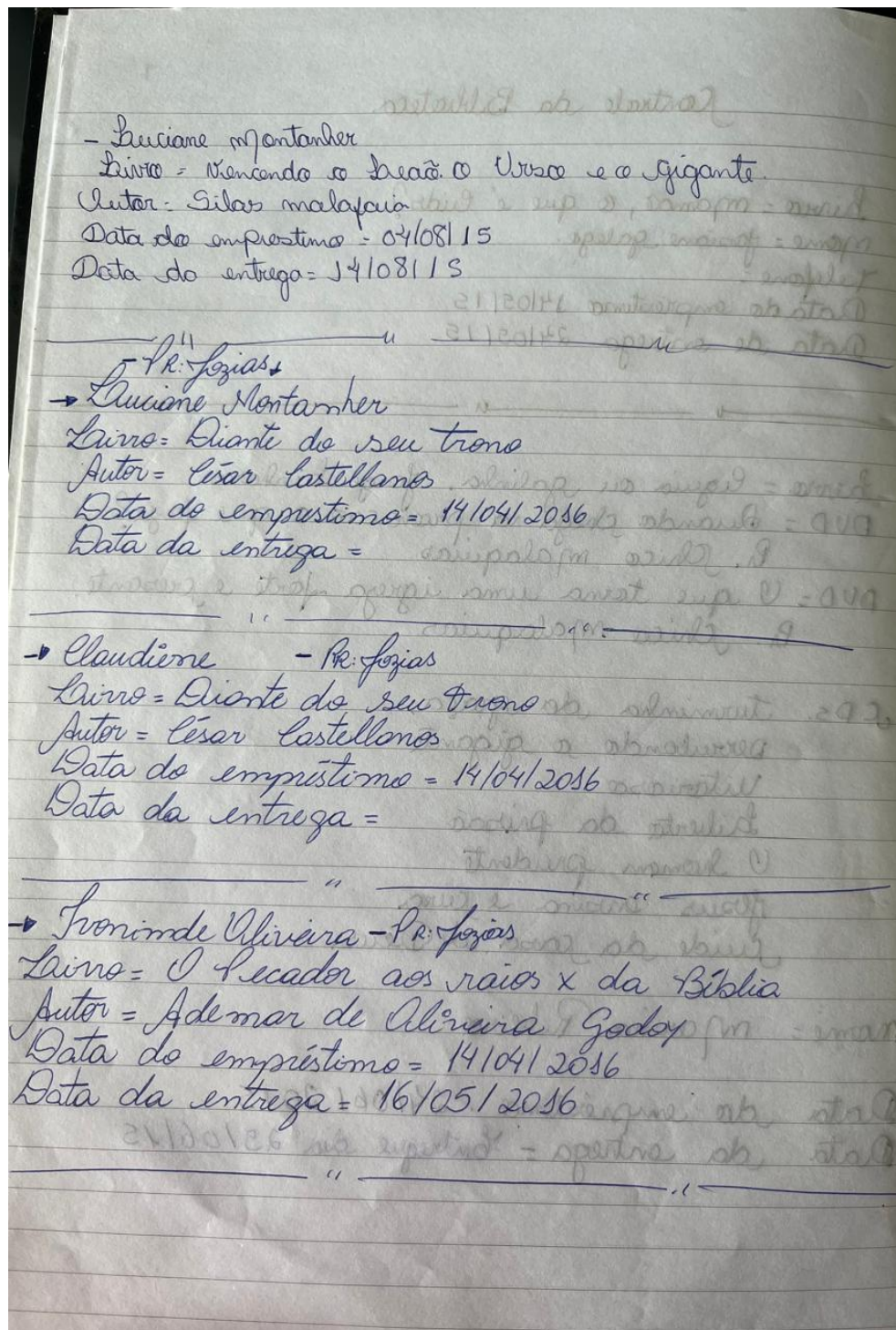
9. ANEXOS

Anexo 1 - Controle da Biblioteca



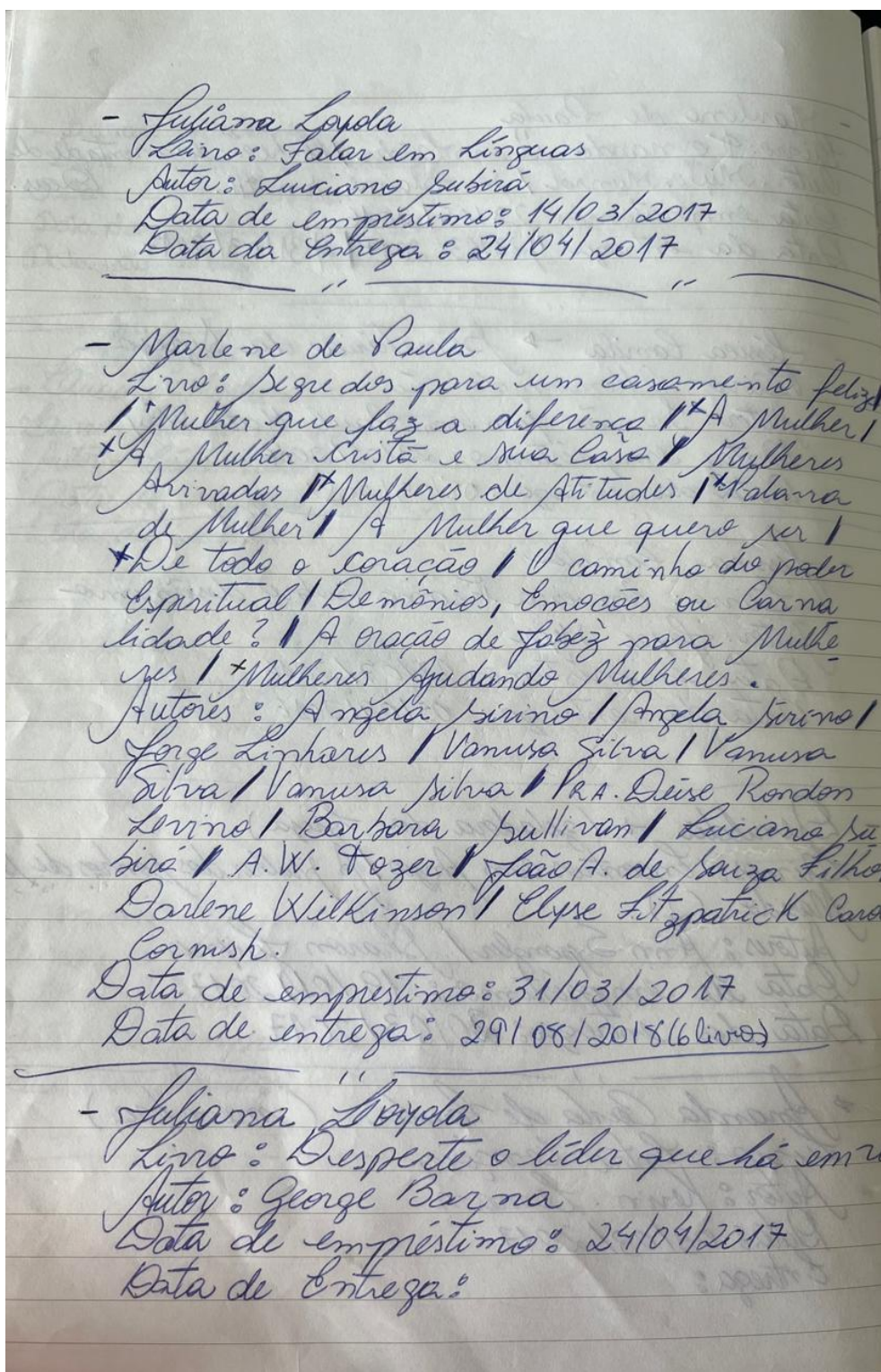
Fonte: Administrador da Biblioteca

Anexo 2 - Controle da Biblioteca 2



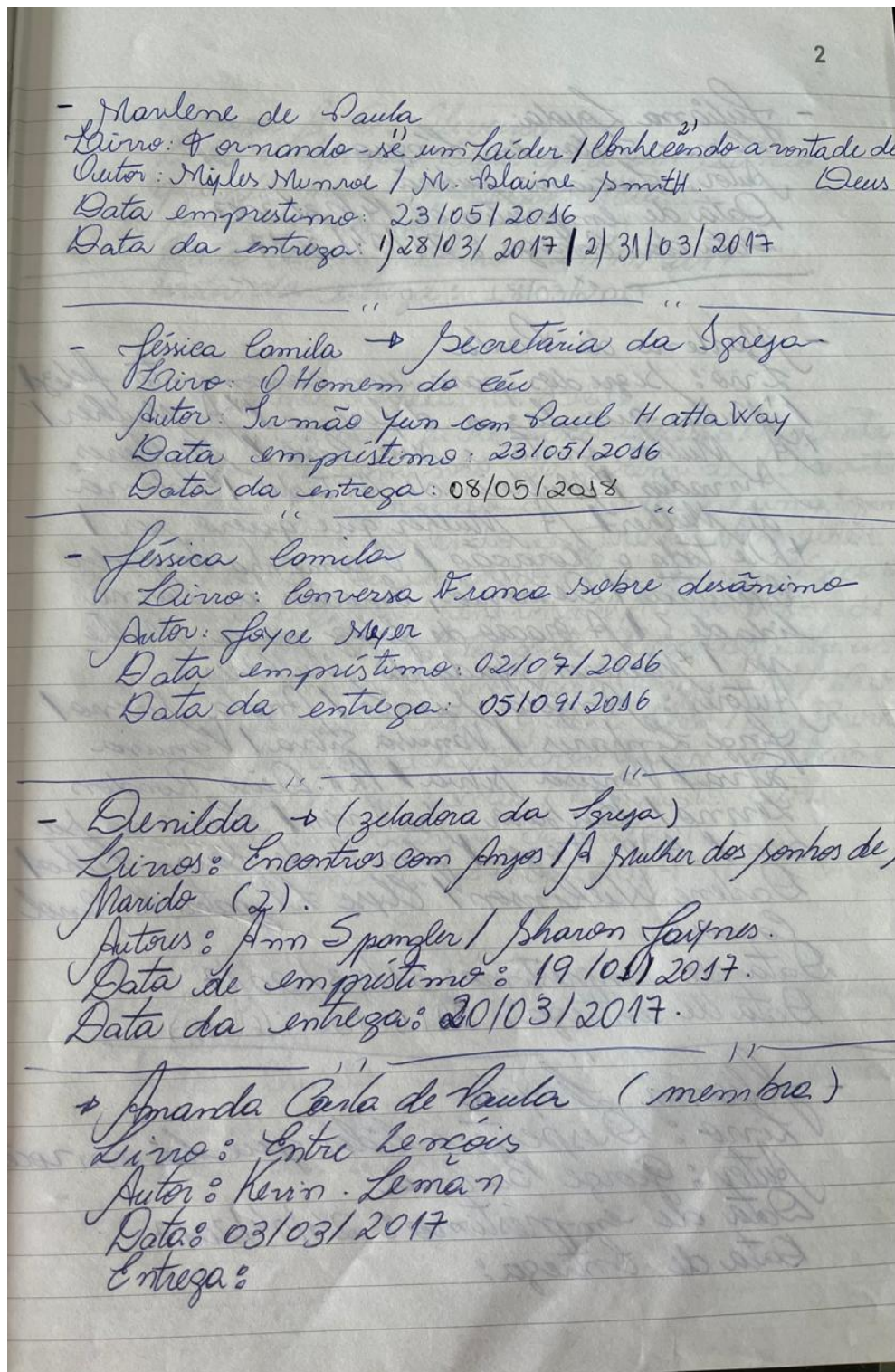
Fonte: Administrador da Biblioteca

Anexo 3 - Controle da Biblioteca 3



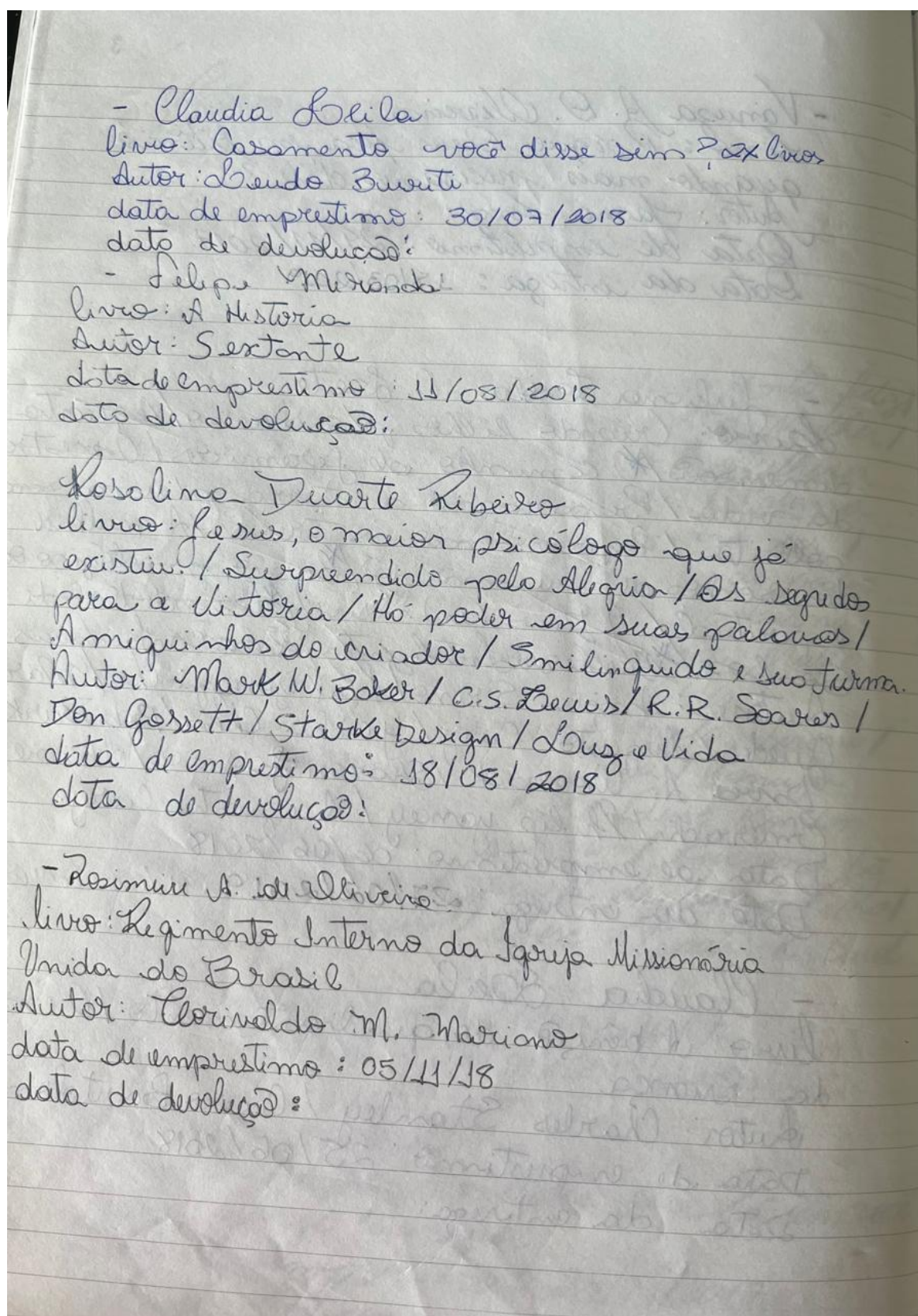
Fonte: Administrador da Biblioteca

Anexo 4 - Controle da Biblioteca 4



Fonte: Administrador da Biblioteca

Anexo 5 - Controle da Biblioteca 5



Fonte: Administrador da Biblioteca

Anexo 6 - Armazenamento dos livros

<u>Livro</u>	<u>Autor</u>	1
1 A História da Bíblia	-	
2 A Receita de Neemias para uma vida vitoriosa. II	Gilas Malafaia	
3 A Mulher	Jorge Linhares	
4 A Força do peixe	Jorge Linhares	
5 A Teologia do Cachorro e do Gato.	Bob Payne	
6 A Mulher Cristã e sua Casa. III	Mrs. Vanuza	
7 A Pirâmide Vermelha	Rick Riordan	
8 A Medida de um Homem espiritual	Abba Pires. / Gene A.	
9 A Igreja, o País e o Mundo.	Robson Cabralente	
10 Autoridade Espiritual	Gilas Malafaia	
11 A Lei da Atração	Michael J. Joseph	
12 A Verdade sobre Repro- dução Hormonal	-	
13 A Fara na Balança	Glenis Fonseca	
14 Avaliações de Deus para na Igreja.	-	
15 A Estratégia	Antônio Vieira / Louis P.	
16 A imagem e semelhança de Deus	Dr. Paul Brand.	
17 Ai que dor / Ai que aflição	Jorge Linhares	
18 A Família Cristã nos dias atuais	Clinaldo Renovato	
19 A mulher dos sonhos de seu marido	Pharon Jaynes	
20 A educação dos nossos Filhos	Neil T. Anderson e Steve Russo.	

Fonte: Administrador da Biblioteca

Anexo 7 - Armazenamento dos livros 2

21	Arrepender-se para não cair	R.R. Jeares
22	Alguns ensinamentos acerca do Salmo 23 e 46	Pr. Márcio Valadão
23	A quarta dimensão	Paul. Yonngi Cho
24	A isca de Botomás	John Bevere
25	Adão e Eva	Gordon Lindsay
26	A Mulher que quero ser.	Barbara Fullerton
27	Atos comentários	Myer Pearlman
28	As portas do inferno não se avaliam.	D. James Kennedy
29	Até o Lugar da Vitória	-
30	A outra face dos Milagres.	Luciano Subirã
31	A escolha suprema	Nilson Di-marzio
32	A Receita para Cumprir seus objetivos.	Julia Malafaia
33	A importância de termos experiências com Deus.	Julia Malafaia
34	A autoridade do Crente	Kenneth C. Hagin
35	A morte do grão de trigo.	Glenis Fonseca P.
36	A Porta Mágica	Haroldo Maranhão
37	A Noiva Restaurada	Neuza Hioza
38	A bruxa próspera e pestor e os recursos do reino.	Donald. C.
39	A Cruz e a Coroa	Paulo Robinson
40	As doenças da Família	George Robello
41	A arte de virar o jogo no segundo tempo da	Peter Drucken

Fonte: Administrador da Biblioteca

Anexo 8 - Armazenamento dos livros 2

64	A Jornada de União	Helena Ramazzo	-
	Fliz		-
65	Abandagem e Prevenção		-
	em dependência Química		-
	Missões Nacionais		-
66	A predestinação conforme		-
	a Bíblia		-
67	A educação de Deus		-
	para os problemas		-
	da Vida		-
68	A Vida de Cristo		-
69	A França Antártica		-
70	A Demografia Grega		-
71	A Inquisição Italiana		-
72	A Guerra do Paraguai		-
73	Antigo Egito e novo Império		-
74	fiéis e inimigo crônico		-
75	A Grécia Mites e Lendas		-
76	Alimento da Fé		-
77	A nós com Deus	Campbell Mcalpine	
78	Bênção e Maldição	George Linhares	
79	Bem-vindo, Espírito	Benny Hinn	
	Fonte		-
80	Bendito Fruto do Ventre	George Linhares	
81	Bons Pais Filhos Me	Julia Malafaia	
	Mors		-
82	Bebendo da Fonte	Antônio Renato Guss	
83	Bem aventurado	Kenneth Haggin Jr	
84	Bênção ou Maldição	Nide Volnei Innocente	
85	Bênções de Deus para		-
	Você		-
86	Bíblia da Criança		-
87	Bem vindo a família		-

Fonte: Administrador da Biblioteca

Anexo 9 - Armazenamento dos livros 3

5	
174 Emoções saudáveis	Josadak Lima
175 Escute aqui, botanais	Carlos Amacordia
176 Esse maravilhoso Pentecostes	João Paulo Lachini
177 Em sua presença	Charles Stonley
178 Eu não consigo emagrecer.	Dr. Piene Duker
179 Equilíbrio firmando sua saúde espiritual	Josadak Lima
180 Ele veio para libertar os latinos	Rebeca Brown
181 Ensinando para trans- formar vidas	Howard Hendricks
182 Evangelismo Explosivo	D. James Kennedy
183 Entendas o que lês?	Gordon e Douglas
184 Edificando uma Igreja Contextualizada.	Colin W. Dye
185 Em diálogo com a Bíblia Filipenses.	L. W. Jungartner
186 Estatuto da criança e do Adolescente.	Anselmo d. Jesus
187 Evangelização Dinâmica	Luisa J. Walker
188 Esse maravilhoso Pentecostes.	João Paulo Lachini
189 Escalando o abismo	Vilma Laudelino
190 Ei, maravilhosos!	Adriana Helen Berger
191 Esperança pra viver	Ellen G. W.
192 Educando o coração da criança	Cicero Bente
193 Ensina-me sobre o Amor	John e Sandra
194 Experimente uma nova experiência de fé.	Beth Moore

Fonte: Administrador da Biblioteca

Anexo 10 - Separação de Livros Por Etiqueta de Cores (Categoria Verde)



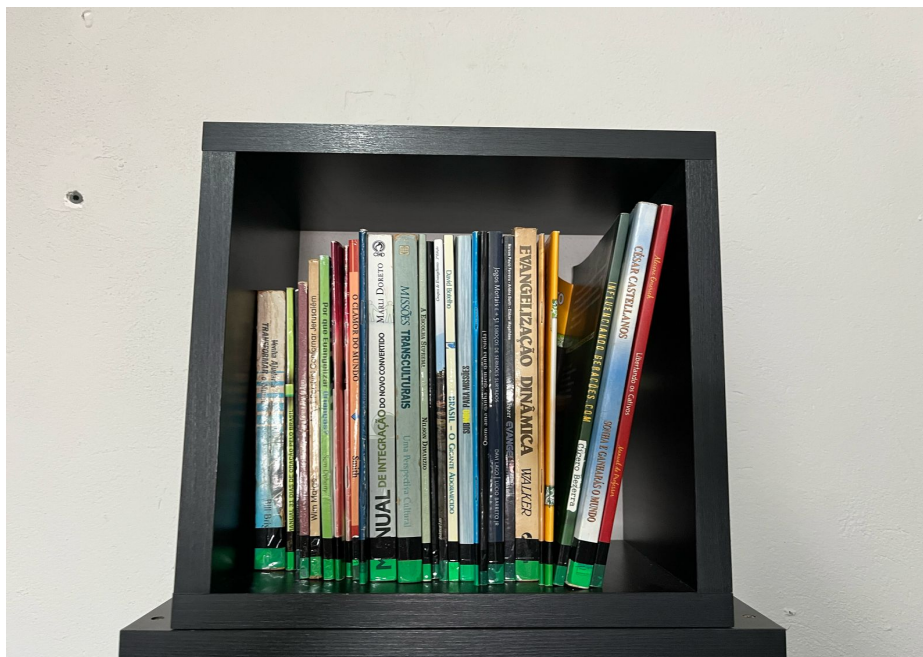
Fonte: Administrador da Biblioteca

Anexo 11 - Separação de Livros Por Etiqueta de Cores (Categoria Preto)



Fonte: Administrador da Biblioteca

Anexo 12 - Separação de Livros Por Etiqueta de Cores(Categoria Verde-Preto)



Fonte: Administrador da Biblioteca

Anexo 13 - Separação de Livros Por Etiqueta de Cores(Categoria Azul-Vermelho)



Fonte: Administrador da Biblioteca

Anexo 14 - Separação de Livros Por Etiqueta de Cores (Apenas algumas Categorias de Cores)



Fonte: Administrador da Biblioteca