

GABRIEL BROGOTA LOBO DURANTE
RAFAEL DOS SANTOS AFONSO
VINÍCIUS FREIRE DE OLIVEIRA BORGES

Projeto Desapega

São Paulo
2021

GABRIEL BROGOTA LOBO DURANTE
RAFAEL DOS SANTOS AFONSO
VINÍCIUS FREIRE DE OLIVEIRA BORGES

Projeto Desapega

Versão Original

Monografia apresentada ao PCS – Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de Engenharia Elétrica.

Área de Concentração: Engenharia de Computação

Orientador: Prof. Dr. Kechi Hirama

São Paulo
2021

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Catálogo-na-publicação

Durante, Gabriel

Projeto Desapega / G. Durante, R. Afonso, V. Borges -- São Paulo, 2021.
111 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Aplicativo 2.Android 3.Doação de livros I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t. III.Afonso, Rafael IV.Borges, Vinicius

DEDICATÓRIA

Esta monografia é dedicada aos familiares, amigos e professores que nos apoiaram e estiveram conosco durante todos esses anos de curso. Sem vocês ela não teria sido possível.

AGRADECIMENTOS

À Universidade de São Paulo – USP que sempre ofereceu um espaço rico de conhecimento e aprendizado.

À Escola Politécnica da Universidade de São Paulo – EPUSP que nos ensinou a superar desafios antes pensados impossíveis, e por todos os ensinamentos ao longo dos anos, não apenas técnicos, mas sociais e éticos.

Ao PCS – Departamento de Engenharia de Computação e Sistemas Digitais que nos proporcionou o estudo da nossa paixão: computadores e sistemas digitais.

Aos nossos pais que sempre nos apoiaram e estavam lá nos momentos de preocupação e de conquista. Nada teria sido possível sem o seu apoio.

Ao Prof. Dr. Kechi Hirama, que nos orientou com toda a sua sabedoria e paciência. Nos ajudou a chegar muito mais longe do que chegaríamos sem a sua valiosa ajuda.

RESUMO

DURANTE, Gabriel Brogota Lobo; AFONSO, Rafael dos Santos; BORGES, Vinícius Freire De Oliveira. **Projeto Desapega**. 2021. 111 páginas. Monografia. Departamento de Engenharia de Computação e Sistemas Digitais Politécnica da Universidade de São Paulo. São Paulo. 2021.

A motivação principal do projeto foi a ideia de poder desenvolver algo de utilidade para a sociedade por meio das necessidades de uma organização social. Assim, em parceria com o Enactus USP, uma organização estudantil sem fins lucrativos vinculada à USP, acordou-se desenvolver um sistema de cunho social, cujo objetivo é facilitar a troca e a doação de livros. Assim, o objetivo do projeto, denominado Projeto “Desapega”, é desenvolver um sistema com *back-end* e *front-end* resultante em um aplicativo para celulares com sistema operacional Android. O sistema foi desenvolvido utilizando os *frameworks* Django e React Native. O trabalho adotou um ciclo de desenvolvimento de projeto baseado no modelo Cascata seguindo as fases tradicionais, ou seja, Especificação, Análise, Projeto, Implementação, Testes e Instalação. Foi obtido como resultado uma aplicação que atende aos requisitos principais do projeto, possibilitando a troca e doação de livros entre usuários, com uma navegação adequada para diversos perfis de usuários. O projeto foi desenvolvido de maneira estruturada e documentada, o que permite ao Enactus USP, no futuro, evoluir a aplicação para transformá-la em produto.

Palavras-chave: React Native, Django, aplicativo, Android, doação de livros, troca de livros.

ABSTRACT

DURANTE, Gabriel Brogota Lobo; AFONSO, Rafael dos Santos; BORGES, Vinícius Freire De Oliveira. **Projeto Desapega**. 2021. 111 páginas. Monografia. Departamento de Engenharia de Computação e Sistemas Digitais Politécnica da Universidade de São Paulo. São Paulo. 2021.

The main project motivation was the idea to develop something of use to the society through the needs of a social organization. Thus, in partnership with Enactus USP, a student organization associated with USP, it was agreed to develop a system of social nature, whose objective is to ease book exchange and donation. Therefore, the project objective, named "Projeto Desapega", is to develop a system with back-end and front-end that results in an app for Android operating system smartphones. The system was developed with the Django and React Native frameworks. The project adopted a development cycle based on the Waterfall model following its traditional steps, that is, Requirements, Analysis, Design, Development, Testing and Deployment. The result achieved is an application that satisfies the main project requirements, allowing exchanging and donating books between users, with an adequate navigation to different user profiles. The project was developed in a structured and documented manner, allowing Enactus USP, in the future, to improve the application in order to turn it into a product.

Keywords: React Native, Django, application, Android, books donations, books exchanges.

LISTA DE ILUSTRAÇÕES

	Pág.
Figura 1 - Arquitetura do sistema	20
Figura 2 - Diagrama de casos de uso	38
Figura 3 - Diagrama simplificado do fluxo de transações.....	39
Figura 4 - Diagrama de classes em alto nível - Cliente	40
Figura 5 - Diagrama de classes - Servidor	41
Figura 6 – O Padrão MVC	70
Figura 7 - Diagrama de classes - API de conexão entre cliente e servidor	71
Figura 8 - Autenticação por Cookie	77
Figura 9 - Autenticação por Token	78
Figura B.1 - Diagrama de classes completo detalhado	101
Figura B.2 - Diagrama de classes completo detalhado - Parte 1 Cliente	102
Figura B.3 - Diagrama de classes completo detalhado - Parte 2 Cliente	103
Figura B.4 - Diagrama de classes completo detalhado - Parte 3 Cliente	104
Figura B.5 - Diagrama de classes completo detalhado - Parte 4 Cliente	105
Figura B.6 - Diagrama de classes completo detalhado - Parte 1 Servidor	106
Figura B.7 - Diagrama de classes completo detalhado - Parte 2 Servidor	107
Figura B.8 - Diagrama de classes completo detalhado - Parte 3 Servidor	108
Figuras C.1 à C.3 - Telas do onboarding, menu principal e informações de livro...109	
Figuras C.4 à C.6 - Telas de informações de livro, escaneamento e confirmação..110	
Figuras C.7 à C.9 - Telas de oferta, solicitação e cadastro de livro.....110	
Figuras C.10 - Tela da interface do administrador	111

LISTA DE TABELAS

	Pág.
Tabela 1 - Resumo do estudo de stack de tecnologias	23
Tabela 2 - Recursos necessários para testes	41
Tabela 3 - Plano de testes de validação	42
Tabela 4 - Resultados dos testes de validação	85
Tabela A - Relação entre casos de uso e requisitos funcionais	100

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
MVC	<i>Model-View-Controller</i>
SQL	<i>Structured Query Language</i>
USP	Universidade de São Paulo

SUMÁRIO

	Pág.
1. INTRODUÇÃO	13
1.1 Motivações	13
1.2 Objetivo e Escopo	14
1.3 Justificativas	14
1.4 Método de Trabalho	15
1.5 Estrutura do Trabalho	16
2. ESPECIFICAÇÃO	18
2.1 Requisitos Funcionais e Não-Funcionais	18
2.2 Arquitetura do Sistema	19
2.3 Análise de Tecnologias	20
3. ANÁLISE	24
3.1 Casos de Uso	24
3.2 Diagrama de Classes	39
3.3 Plano de Testes de Validação	41
4. PROJETO	70
4.1 Descrição de Padrões e APIs do Projeto	70
4.2 Especificação das Classes	71
4.2.1 Cliente	71
4.2.2 Servidor	75
4.3 Estrutura de Segurança	76
5. IMPLEMENTAÇÃO	80
5.1 Organização do código	80
5.1.1 Mobile	80
5.1.2 Servidor	82
5.2 Repositórios	84
6. TESTES	85
6.1 Relatório de Testes de Validação	85
6.2 Discussões sobre o Processo de Validação	90
7. INSTALAÇÃO	92
7.1 Cliente	92
7.1.1 Pré-requisitos do Dispositivo	92
7.1.2 Instalação do Software	92
7.2 Servidor	93

7.2.1 Pré-requisitos do Servidor	93
7.2.2 Instalação do Software	93
8. CONSIDERAÇÕES FINAIS	94
8.1 Conclusões	94
8.2 Contribuições do Trabalho	95
8.3 Trabalhos Futuros	96
REFERÊNCIAS	99
APÊNDICE A - Tabela de relação entre casos de uso e requisitos funcionais	100
APÊNDICE B - Diagrama de classes detalhado	101
APÊNDICE C - Seleção de telas da aplicação final	109

1. INTRODUÇÃO

Este capítulo apresenta as motivações, o objetivo, o escopo, as justificativas, o método e a estrutura deste trabalho.

1.1 Motivações

A motivação principal para a escolha do projeto foi a de poder construir algo que seria de utilidade para a sociedade, ajudando as necessidades de alguma organização social. A parceria com o Enactus USP, uma organização estudantil sem fins lucrativos vinculada à USP, se mostrou adequada para o projeto. Eles apresentaram uma ideia de projeto de cunho social focado na doação e troca de livros, algo tão relevante no cotidiano, onde o acesso a livros pela população está cada vez mais difícil.

Segundo o Instituto Pró-Livro (2020, p. 20), em um estudo realizado em parceria com o Itaú Cultural, o Brasil perdeu 4,6 milhões de leitores no período de 2015 a 2019. Esse resultado apresenta uma tendência de diminuição de leitura pela população brasileira.

Dentre as razões apresentadas pelos participantes da pesquisa como justificativa de por que não leram mais, cerca de 19% apontaram problemas relacionados ao acesso aos livros (INSTITUTO PRÓ-LIVRO, 2020, p. 58). Assim, um sistema que apoie esse acesso se mostraria extremamente relevante na atualidade, visto a importância e relevância que a leitura tem social e culturalmente.

Ao analisar as necessidades do Enactus USP, foi possível perceber uma oportunidade de aplicar diversos conceitos de desenvolvimento de software que foram fragmentados ao longo do curso de Engenharia de Computação. Além disso, é uma ótima oportunidade para trabalhar com um cliente em um processo similar ao mundo profissional fora da USP.

1.2 Objetivo e Escopo

Este trabalho tem como objetivo o desenvolvimento de um sistema para celular, denominado Projeto “Desapega”, que visa dar acesso às obras literárias das pessoas trocando e doando livros. O desenvolvimento dessa aplicação serve como base de estudo para diversos fatores relacionados a desenvolvimento de software, tais como sistemas de recomendação, bancos de dados, desenvolvimento mobile, comunicação textual simultânea (sistema de *chat*), conceitos de experiência de usuário, qualidade de software e métodos ágeis. Além disso, esta é uma experiência valiosa no contexto do mercado de trabalho, pois envolve a criação de uma aplicação do começo ao fim para um cliente do mundo real.

O escopo do projeto engloba o desenvolvimento de um sistema de celular com as principais funções do sistema descritas neste trabalho. Esse além de ser funcional, também servirá como base para evoluir o sistema em novas melhorias que possam ser implementadas pelo cliente no futuro, caso assim deseje para inserir novas funções no sistema.

Não fazem parte deste trabalho o desenvolvimento da aplicação para sistemas iOS e a função de *marketplace* dos livros para compras e vendas internas no sistema. Contudo, isso é levado em consideração na arquitetura do projeto, de forma a facilitar implementações futuras caso o cliente assim deseje.

1.3 Justificativas

O desenvolvimento deste projeto engloba alguns pontos importantes. Em primeiro lugar, a temática social do tema do projeto, juntamente à relevância de promover o acesso à leitura é algo que poderá ter um impacto extremamente positivo na sociedade.

Em segundo lugar, esta experiência é de grande valia para os autores do projeto no contexto do mercado de trabalho e desenvolvimento de software com aplicações fora da USP e com um cliente real. Esta é uma oportunidade de trabalhar com aspectos de projeto (negociação, desenvolvimento e aceitação do projeto) que o curso não pode oferecer dentro do contexto de aulas.

Em terceiro lugar, o desenvolvimento deste sistema é um pretexto para trabalhar e estudar com mais profundidade uma série de conceitos e técnicas aprendidas ao longo do curso de forma fragmentada, agora de maneira uniforme e conjunta. Esse último ponto não engloba apenas a parte prática da programação, mas também todo o processo de um projeto e organização prévia do desenvolvimento do software.

1.4 Método de Trabalho

Este trabalho segue um ciclo de desenvolvimento de projeto baseado no Modelo Cascata seguindo as fases tradicionais, ou seja, Especificação, Análise, Projeto, Implementação, Testes e Implantação.

Uma vez conhecidas as necessidades do cliente, a especificação de requisitos funcionais e não-funcionais e a arquitetura do sistema são elaboradas e são estudadas as tecnologias aplicáveis a um projeto do tipo Cliente-Servidor envolvendo *front-end* e *back-end*, considerando interfaces de celular para interação dos usuários com o sistema e bancos de dados para armazenamento de dados. Além de protocolos de comunicação para Internet.

Em seguida, a especificação de requisitos do sistema é traduzida em funções agrupadas em Casos de Uso. O sistema é então estruturado em Classes.

A implementação das classes e os seus testes são realizados até os testes de validação do sistema para verificar se todos os requisitos funcionais e não-funcionais estão presentes.

Uma vez passado nos testes, o Manual de Instalação do sistema é elaborado.

1.5 Estrutura do Trabalho

Este trabalho é organizado de acordo com os seguintes capítulos.

1. INTRODUÇÃO

Este capítulo descreve as motivações, o objetivo e escopo, as justificativas, o método e a estrutura do trabalho.

2. ESPECIFICAÇÃO

Este capítulo descreve as necessidades levantadas junto ao cliente, tendo como resultado a especificação de requisitos funcionais e não-funcionais do sistema. Descreve também o padrão arquitetural aplicável ao sistema. Além disso, apresenta um estudo das principais tecnologias e a sua aplicação no sistema.

3. ANÁLISE

Este capítulo descreve os Casos de Uso elencados com base nos requisitos funcionais, o Diagrama de Classes simplificado obtido a partir da análise dos Casos de Uso e o Plano de Testes de Validação do sistema.

4. PROJETO

Este capítulo apresenta as classes em detalhes, os relacionamentos entre as classes e suas descrições. Também são descritos os padrões de projeto usados.

5. IMPLEMENTAÇÃO

Este capítulo apresenta a descrição da estrutura do código-fonte do sistema.

6. TESTES

Este capítulo apresenta o Relatório de Testes de Validação do sistema de acordo com os testes realizados.

7. INSTALAÇÃO

Este capítulo descreve um procedimento para a instalação do aplicativo em um dispositivo móvel (*smartphone*) e no servidor.

8. CONSIDERAÇÕES FINAIS

Este capítulo apresenta as conclusões, as contribuições do trabalho e os trabalhos futuros.

REFERÊNCIAS

Este capítulo apresenta uma lista de fontes de pesquisa usadas na elaboração deste trabalho.

APÊNDICES

Os apêndices apresentam o detalhamento do projeto.

2. ESPECIFICAÇÃO

Este capítulo apresenta a lista de requisitos funcionais e não-funcionais levantados com o cliente, a arquitetura básica do sistema, a especificação de Casos de Uso e o Diagrama de Classes em alto nível.

2.1 Requisitos Funcionais e Não-Funcionais

A partir de sucessivas conversas com a equipe do Enactus USP, chegou-se à lista de requisitos funcionais abaixo. Esses requisitos definem o escopo do projeto.

O sistema deve:

1. Cadastrar os usuários
2. Autenticar os usuários
3. Oferecer funções limitadas para usuários não cadastrados
4. Apresentar telas de *onboarding* para usuários não logados
5. Oferecer opções de títulos de livros (banco de dados padronizado para os nomes)
6. Gerenciar históricos de livros lidos
7. Gerenciar livros desejados
8. Pesquisar livros por palavras chaves
9. Recomendar livros de acordo com o perfil do usuário
10. Recomendar livros recentes e populares
11. Permitir que usuários adicionem livros para doação ou troca
12. Identificar qual livro está sendo adicionado
13. Registrar fotos e estado do livro
14. Listar ofertas de livros para doação ou troca em ordem de proximidade
15. Oferecer opções de compra em lojas parceiras (*links* patrocinados)
16. Dar suporte a um sistema de avaliação de usuários

17. Ter telas de perfil para cada usuário
18. Ter um sistema de *chat* para conversa entre usuários
19. Oferecer dicas de segurança no início de uma conversa entre usuários
20. Gerenciar notificações com o andamento de transações
21. Permitir que usuários aceitem ou recusem ofertas de troca ou doação
22. Oferecer sistema de leitura e escrita de resenhas para livros
23. Pesquisar perfis de outros usuários
24. Ter um *feed* de pessoas que o usuário segue contendo resenhas postadas.
25. Dar suporte a um sistema básico de moderação de banco de dados.
26. Permitir que usuários realizem denúncias.

Como requisito não-funcional foi definida a Segurança. O sistema deve ser capaz de impedir acessos não autorizados à contas de usuários com a combinação incorreta de *e-mail* e senha. Além disso, o banco de dados deve ser resistente a vazamentos de informações privadas do usuário, como por exemplo a senha de acesso ao aplicativo. Sendo assim, a autenticação dos usuários deve ser protegida.

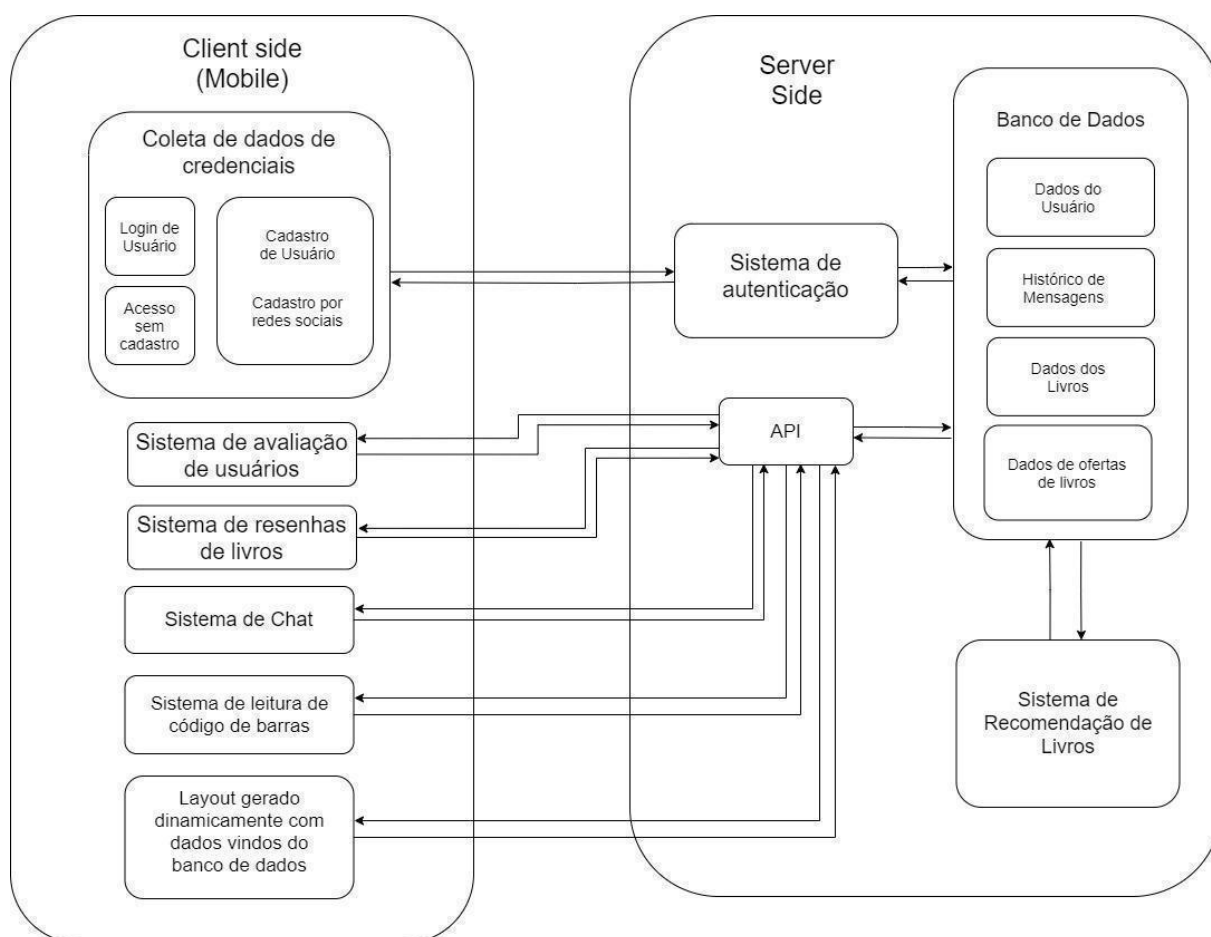
2.2 Arquitetura do Sistema

Baseado nos requisitos listados na seção 2.1, a Arquitetura do Sistema é apresentada na Figura 1 representada por um diagrama de blocos do sistema. O primeiro nível representa a arquitetura Cliente-Servidor que é a base do projeto, onde o Cliente é representado pelos dispositivos móveis (*Mobile*) e o Servidor é representado por servidor na nuvem (*Cloud*).

O segundo nível representa os blocos funcionais básicos do sistema. O Cliente representa o sistema *mobile* e contém blocos que irão acessar os serviços fornecidos pelo *back-end* para gerar telas com informações dinâmicas. Nestes blocos, também existe a coleta de dados dos usuários para as mais diversas

funções. No Servidor, há a implementação da API que permite a requisição dos dados pela parte *mobile* e realiza a lógica envolvida na autenticação dos usuários.

Figura 1 - Arquitetura do sistema



Fonte: autores.

2.3 Análise de Tecnologias

Com base nos Requisitos Funcionais e Não-funcionais e Casos de Uso especificados, foi definido um *stack* de tecnologias utilizadas no desenvolvimento do sistema. Esta seção apresenta as tecnologias escolhidas e as justificativas para a sua utilização.

Em primeiro lugar foram levantadas possibilidades de tecnologias para compreender melhor suas características e entender qual atenderia melhor as características do projeto. Após esse levantamento, duas alternativas foram levadas em consideração: o *React Native* (REACT NATIVE, 2021) e o *Flutter* (FLUTTER, 2021).

O *React Native* é um *framework* em *JavaScript* desenvolvido e mantido pela empresa Facebook dos EUA. Sua grande vantagem em comparação ao desenvolvimento nativo na plataforma *Android* é o de permitir o desenvolvimento para *Android* e *iOS* simultaneamente. O código em *JavaScript* é convertido para código nativo para cada plataforma. Embora o escopo do projeto se limite em desenvolver a aplicação para *Android* apenas, o uso de um *framework* multiplataforma é definido para facilitar um possível porte futuro para plataforma *iOS* caso se deseje evoluir o sistema no futuro. Outro ponto positivo neste *framework* é a ampla gama de bibliotecas para adicionar recursos de forma eficiente, geralmente através do próprio gerenciador de pacotes do projeto.

O *Flutter* é um concorrente direto do *React Native*. Desenvolvido pela empresa Google dos EUA, trata-se de um *framework* em *Dart* que também permite desenvolvimento multiplataforma. Além de permitir desenvolvimento para *Android* e *iOS*, é possível utilizar o mesmo código para aplicações *Web* e *Desktop*.

Um ponto negativo de ambos os *frameworks* é que, por não serem nativos específicos para *Android* ou *iOS* e sim multiplataforma, possuem algumas limitações de funções que são específicas de cada sistema operacional e não estão presentes no outro. Contudo, isto não é um limitante para o projeto, já que é possível fazer ajustes no código nativo.

Ambos oferecem todas as ferramentas necessárias para o desenvolvimento do projeto, mas optou-se por utilizar o *React Native*. A linguagem de programação utilizada (*JavaScript*) já é mais bem estabelecida, além do fato do *React Native* ser um *framework* um pouco mais maduro, com maior estabilidade e suporte. Além disso, os membros do grupo possuem um pouco de familiaridade com

desenvolvimento em *React/React-Native*, o que permite dedicar menos tempo estudando a ferramenta e mais tempo desenvolvendo a aplicação.

Para o servidor, o *framework* escolhido foi o *Django* (DJANGO, 2005). Trata-se de uma solução muito completa, pois possui em seu próprio conjunto de ferramentas com recursos de segurança para evitar ataques de *SQL Injection*, *XSS*, *CSRF*, e outros. Tendo em vista que a aplicação contém uma base de dados de usuários, é importantíssimo garantir a segurança dos dados. Ainda nesse tópico, o *Django* possui o *Django Admin* que proporciona uma estrutura para implementar a função de moderação do sistema sem precisar implementá-la desde o início, permitindo uma maior eficiência. Além disso, ele possui uma ampla documentação e uma comunidade bem ativa, o que facilita encontrar soluções para eventuais dificuldades. Finalmente, o *Django* foi escolhido também por ser um *framework* de da linguagem *Python* e trabalhar muito bem com as bibliotecas desta linguagem.

Com relação a possíveis desvantagens do *Django*, seria a curva de aprendizado do *framework* em si, pois apesar do grupo possuir familiaridade com *Python*, ainda não havia uma experiência significativa com *Django*. Outras opções de *frameworks* para serem aplicados no lado do servidor seriam o Flask (FLASK, 2010) também para a linguagem *Python*, porém este se trata de um *microframework* que não possui um conjunto de ferramentas tão completo quanto o *Django*, fato que consequentemente atrasaria o desenvolvimento do sistema. Também poderia ser utilizado o *Express.js* (EXPRESS, 2017) um *framework* de *Node.js*, também muito utilizado na indústria de software, porém que ofereceria ao grupo uma maior barreira de entrada e curva de aprendizado, além de não apresentar também muitas ferramentas de segurança inclusas.

Um resumo do estudo das tecnologias é mostrado na Tabela 1.

Tabela 1 - Resumo do estudo de *stack* de tecnologias

Parte	Tecnologias	Tecnologia Definida	Justificativa
Mobile	<i>Flutter</i> <i>React Native</i>	<i>React Native</i>	<i>Framework</i> de desenvolvimento mobile ágil e com portabilidade para diversos OS, documentação completa, linguagem (<i>JavaScript</i>) mais bem estabelecida e o grupo já possui relativa experiência.
Servidor	<i>Flask</i> <i>Django</i> <i>Express.js</i>	<i>Django</i>	Framework completo de <i>back-end</i> para <i>Python</i> , com ferramentas de segurança incluídas, documentação muito bem feita e menor curva de aprendizado para o grupo.

Fonte: autores.

3. ANÁLISE

Este capítulo descreve o detalhamento funcional do sistema por meio dos Casos de Uso, apresenta um Diagrama de Classes simplificado baseado na análise dos Casos de Uso e um Plano de Testes de Validação.

3.1 Casos de Uso

Os casos de uso abaixo levam em conta todos os requisitos funcionais do sistema. No Apêndice A encontram-se uma tabela de correlação entre os casos de uso e os requisitos funcionais para evidenciar a representação de todos os requisitos.

Caso de Uso 1: Acessar aplicativo

Descrição: O usuário abre o aplicativo que oferece as opções de login, cadastro ou entrar sem cadastro.

Ator: Usuário cadastrado, Usuário não cadastrado

Pré-condição: Não há.

Fluxo principal:

1. Usuário abre o aplicativo
2. Sistema fornece informações gerais sobre o aplicativo (“informações de *onboarding*”). Existem as opções de “pular” ou prosseguir uma a uma até o final das informações.
3. Usuário segue por cada uma das informações.
4. O sistema fornece opções para seguir os fluxos de cadastro ou seguir utilizando a aplicação sem estar logado.
5. O usuário seleciona o fluxo que deseja seguir.
6. Fim do caso de uso.

Pós-condição: Usuário é redirecionado para o caso de uso adequado.

Fluxos de extensão:

(a) Usuário pula as telas com mensagens introdutórias. (passo 3)

1. Usuário aciona a opção de pular.
2. Sistema redireciona o usuário para a tela de Login, no Caso de Uso 3.
3. Fim do caso de uso.

(b) Usuário aciona opção de acessar o aplicativo sem cadastro. (passo 5)

1. Usuário seleciona opção de usar sem cadastro.
2. Sistema redireciona para o menu principal do aplicativo.
3. Fim do Caso de Uso.

Caso de Uso 2: Cadastrar usuários

Descrição: Cadastra um novo usuário no sistema, coletando nome, email e senha, ou por assistência do Facebook ou Twitter. Coleta, opcionalmente, os gêneros de livro favoritos do usuário. Também abre uma seleção inicial opcional de livros recomendados para o usuário indicar se tem interesse ou já leu.

Ator: Usuário cadastrado, Usuário não cadastrado

Pré-condição: N/A

Fluxo principal:

1. O usuário deseja fazer cadastro.
2. Sistema redireciona para criação de conta.
3. O usuário insere nos campos respectivos: nome, email e senha.
4. Sistema exibe Termos e Condições e solicita “De acordo”.
5. Usuário aceita Termos e Condições de uso.
6. O usuário aciona “Criar uma conta”.
7. Sistema verifica entrada de dados e pede por permissão para detectar localização do usuário por meio do GPS do dispositivo.
8. Usuário aceita permissão.
9. Sistema redireciona usuário para uma lista de gêneros literários. O nome do usuário aparece no topo.
10. O usuário seleciona os gêneros pelos quais tem interesse. Aciona “Próximo”.
11. Sistema redireciona para uma lista de livros recomendados de acordo com os gêneros escolhidos na tela anterior. Ao acionar um livro, surgem as opções de “Já li” e “Quero ler”.
12. Usuário seleciona os livros de forma adequada com o que deseja. Aciona a opção “Próximo”.
13. Sistema exibe confirmação de cadastro. Depois, redireciona automaticamente o usuário para o menu principal.
14. Fim do Caso de Uso.

Pós-condição: Usuário está cadastrado.

Fluxos de extensão:**(a) Cadastro por redes sociais. (passo 3)**

1. Usuário aciona opções do Facebook ou Twitter.
2. Sistema faz a chamada do serviço de assistência de login da rede escolhida.
3. Usuário confirma permissão.
4. Sistema retorna ao passo 5 do fluxo principal.

(a1) Permissão de rede social não concedida. (passo a)

1. Sistema apresenta mensagem de erro de associação com rede social.
2. Sistema retorna ao passo 2 do fluxo principal.

(b) Erro na inserção de dados do cadastro. (passo 5)

1. Sistema exibe uma mensagem de erro adequada para o campo de entrada incorreto.
2. Sistema retorna ao passo 3 do fluxo principal.

(c) Usuário pula seleção de gêneros favoritos. (passo 10)

1. Usuário toca no botão “Pular”.
2. Sistema retorna ao passo 11 do fluxo principal.

(d) Usuário pula seleção de livros recomendados. (passo 11)

1. Usuário toca no botão “Pular”.
2. Sistema retorna ao passo 12 do fluxo principal.

(e) Usuário não concorda com os termos de uso (passo 3)

1. Usuário não concorda com os termos de uso.
2. Sistema informa que o cadastro não pode seguir sem a aceitação dos termos.
3. Fim do Caso de Uso.

(f) Usuário já cadastrado. (passo 7)

1. Sistema identifica que aquele email já foi previamente cadastrado.
2. Sistema exibe mensagem de erro.
3. Fim do Caso de Uso.

(g) Usuário não aceita permissão de localização. (passo 8)

1. Usuário não aceita permissão de localização
2. Sistema exibe mensagem de erro.
3. Fim do Caso de Uso.

Caso de Uso 3: Permitir acesso de usuário

Descrição: Usuário entra com email e senha ou por meio da autenticação de redes sociais. O sistema realiza login do usuário caso as credenciais estejam adequadas e o redireciona para o menu principal.

Ator: Usuário cadastrado, Usuário não cadastrado

Pré-condição: N/A

Fluxo principal:

1. Usuário solicita acesso ao serviço de doação e troca de livros.
2. Sistema exibe a tela de login, com campos de email e senha, e botões para autenticação opcional por redes sociais.
3. Usuário insere email e senha e aciona “Entrar”.
4. Sistema valida os dados e redireciona para a tela principal após a validação.
5. Fim do Caso de Uso.

Pós-condição: Usuário logado.

Fluxos de extensão:

(a) Login autenticado por redes sociais. (Passo 2)

1. Usuário opta por rede social já previamente cadastrada.
2. Sistema faz a autenticação dos dados do usuário.
3. Sistema retorna ao passo 3 do fluxo principal.

(b) Autenticação de usuário falhou. (Passo 3)

1. Sistema detecta erro na autenticação de usuário e exibe uma mensagem de erro.
2. Fim do Caso de Uso.

Caso de Uso 4: Visualizar recomendações personalizadas de livros

Descrição: Usuário visualiza recomendações de livros. Usuários não logados têm acesso apenas a seções “Novos” e “Populares”, insensíveis às preferências do usuário.

Ator: Usuário cadastrado.

Pré-condição: Usuário está logado.

Fluxo principal:

1. Usuário aciona uma opção do serviço de doação ou troca de livros.
2. Sistema exibe subseções de livros: uma lista de livros recomendados para o usuário (se logado), uma lista de livros novos, e uma lista de livros populares.
3. Usuário pode navegar pelas recomendações listadas separadas de acordo com suas subseções ("populares", "novos" e "recomendados").
4. Usuário aciona opção para expandir uma subseção e visualizar mais opções de cada lista.
5. Usuário tem a opção de selecionar um dos livros listados para ver mais informações.
6. Fim do Caso de Uso.

Pós-condição: N/A

Fluxos de extensão: N/A

Caso de Uso 5: Sistema de busca

Descrição: Usuário pesquisa por um livro específico ou por um usuário. Sistema responde com livros ou perfis relacionados ao que foi digitado.

Ator: Usuário cadastrado, Usuário não cadastrado

Pré-condição: N/A

Fluxo principal:

1. Usuário busca um termo relacionado ao livro que deseja pesquisar, como parte do nome ou autor. Ele possui opções de filtragem do livro (e.g. gênero, condição do livro, etc). É necessário apertar um botão para realizar a busca.

2. Sistema realiza busca por livros catalogados que possuam referência aos termos digitados. Então, exibe em uma lista em ícones com a capa dos livros os resultados da pesquisa no banco de dados.
3. Fim do Caso de Uso.

Pós-condição: Lista de livros relacionados à busca encontrados são exibidos com suas respectivas capas.

Fluxos de extensão:

(a) Nenhum resultado encontrado. (passo 2)

1. Sistema não conseguiu encontrar nenhum resultado para a busca. Exibe na tela uma mensagem avisando ao usuário a falta de resultados para o termo digitado.
2. Fim do Caso de Uso.

(b) Busca por perfis de outros usuários. (passo 1)

1. Usuário busca pelo nome do perfil que deseja encontrar.
2. Sistema realiza busca por usuários cadastrados na plataforma e exibe resultados.
3. Fim do Caso de Uso.

(c) Pesquisa apenas por filtros. (passo 1)

1. Usuário não entra com nenhum termo na busca, apenas seleciona filtros.
2. Sistema realiza busca por livros catalogados que tem relação com os filtros selecionados. Então, exibe em uma lista em ícones com a capa dos livros os resultados da pesquisa no banco de dados.

Caso de Uso 6: Ler e escrever resenhas

Descrição: Usuário é capaz de ler e escrever resenhas de um livro específico.

Ator: Usuário cadastrado, Usuário não cadastrado

Pré-condição: Usuário está logado (apenas para escrever)

Fluxo principal:

1. Usuário acessa resenhas na seção de detalhe do livro sob a aba “Sobre”.
2. Sistema exibe resenhas do livro feitas por outros usuários do aplicativo.
3. Usuário navega na lista de resenhas e lê quantas quiser.
4. Fim do Caso de Uso

Pós-condição: Usuário leu resenhas.

Fluxos de extensão:

(a) Usuário quer escrever resenhas. (passo 1)

1. Usuário seleciona a opção de escrever uma resenha no botão “+” no menu principal.
2. Sistema verifica se usuário tem permissão de escrita.
3. Sistema redireciona para que o usuário escreva a resenha sobre o livro escolhido.
4. Usuário escreve sua resenha.

5. Fim do Caso de Uso.

(b) Usuário não está logado. (passo a2)

1. Sistema identifica que usuário não está logado.
2. Sistema exibe mensagem de erro.
3. Fim do Caso de Uso.

(c) Usuário já escreveu uma resenha para aquele livro. (passo a2)

1. Sistema exibe opção de apagar ou editar a resenha escrita.
2. Usuário aciona opção de edição e edita o texto.
3. Fim do Caso de Uso.

(c1) Usuário quer apagar a resenha escrita. (passo c1)

1. Usuário aciona opção de apagar a resenha escrita.
2. Sistema remove resenha do usuário.
3. Fim do Caso de Uso.

Caso de Uso 7: Marcar/Desmarcar livros que tem interesse ou já leu

Descrição: Usuário marca se já leu ou quer um livro, e associa essa informação ao seu perfil.

Ator: Usuário cadastrado

Pré-condição: Usuário logado.

Fluxo principal:

1. O usuário aciona a página correspondente ao livro em questão.
2. Sistema exibe livro em questão.
3. Usuário aciona um campo indicando que deseja marcar este livro.
4. O sistema exibe opções “Quero ler” e “Já li”.
5. O usuário seleciona a opção que deseja.
6. O sistema vincula informação ao perfil do usuário e fecha a caixa de opções. Tag da opção escolhida agora aparece sob o nome do autor, junto a outras tags.
7. Fim do Caso de Uso.

Pós-condição: O livro escolhido é vinculado ao usuário. Tag da opção escolhida agora aparece sob o nome do autor, junto a outras tags.

Fluxos de extensão:

(a) Usuário cancela operação. (passo 3)

1. Usuário não seleciona nenhuma das duas opções, aciona cancelar.
2. Sistema retorna à tela de informações do livro sem nenhuma alteração.
3. Fim do Caso de Uso.

(b) Usuário deseja excluir livro que já leu ou quer ler. (passo 2)

1. Usuário aciona marcador do livro salvo.
2. Sistema retira marcação do livro do perfil do usuário e a tag.
3. Fim de Caso de Uso.

Caso de Uso 8: Visualizar feed de atividade de usuários seguidos

Descrição: Usuário abre o menu de feed, onde é possível ver resenhas postadas por usuários que segue.

Ator: Usuário cadastrado

Pré-condição: Usuário está logado

Fluxo principal:

1. Usuário aciona opção “Feed” no menu de navegação.
2. Sistema exhibe em ordem de mais recente para mais antiga resenhas de usuários seguidos.
3. É possível interagir com as resenhas por meio do botão “Gostei” ou similar.
4. Usuário navega o quanto quiser na página e aciona um livro.
5. Sistema exhibe informações daquele livro.
6. Fim do Caso de Uso.

Pós-condição: Livro escolhido exibido.

Fluxos de extensão: N/A

Caso de Uso 9: Solicitar livros

Descrição: Usuário faz pedido de doação ou troca de um livro pelo qual tenha interesse. O pedido é enviado ao usuário escolhido que possui o livro, e pode estar acompanhado de uma mensagem e uma oferta de troca por outro livro.

Ator: Usuário cadastrado

Pré-condição: Usuário está logado

Fluxo principal:

1. Usuário seleciona um livro.
2. Sistema exhibe o livro escolhido, contendo uma imagem da capa, nome da obra, autor, gênero, e uma pequena descrição em uma seção “Sobre”. Há uma seção alternativa à descrição do livro chamada “Doadores”.
3. Usuário seleciona doadores.
4. O sistema lista alguns doadores disponíveis. Cada doador na lista mostra nome, localização e um botão “Ver livro”. Lista ordenada por proximidade. Há tags para indicar se o usuário está doando ou trocando.
5. Usuário aciona “Ver livro” do usuário escolhido.
6. Sistema apresenta detalhes daquela oferta. Há fotos exibidas, descrição do estado do livro, uma tag indicando se o usuário que possui o livro quer doar ou trocar a obra, quadros indicando condição, língua do livro, e avaliação e quantidade de transações do doador. Há também a foto, nome e localização do doador, juntamente a um link que redireciona para seu perfil.
7. Usuário aciona “Solicitar esse livro”.
8. Sistema exhibe nome do livro solicitado, doador selecionado, e as opções de adicionar uma mensagem e oferecer um livro em troca.
9. Usuário aciona “Adicionar mensagem”.
10. Sistema exhibe espaço para mensagem no celular do usuário.
11. Usuário digita mensagem.

12. Sistema exibe pré-visualização da mensagem.
13. Usuário aciona “Adicionar livro em troca”.
14. Sistema exibe lista de livros que o usuário possui, cadastrados previamente.
15. Usuário seleciona um livro.
16. Sistema exibe pré-visualização do livro oferecido como troca.
17. Usuário aciona "Enviar solicitação".
18. Sistema envia solicitação para o doador. Redireciona o usuário para confirmação de envio.
19. Sistema exibe “Continuar procurando livros”.
20. Usuário aciona essa opção e é levado à tela inicial.
21. Fim do Caso de Uso.

Pós-condição: Solicitação de doação ou troca por um livro de algum doador registrado, acompanhado de uma mensagem ou não.

Fluxos de extensão:

(a) Usuário não adiciona mensagem e/ou não oferece livro para troca. (passo 9)

1. Sistema segue a partir do passo 16 do fluxo principal.

Caso de Uso 10: Acessar link patrocinado

Descrição: Usuário tem acesso a uma aba de “Compre” nas informações do livro, onde pode acessar links de lojas parceiras.

Ator: Usuário cadastrado, Usuário não cadastrado

Pré-condição: N/A

Fluxo principal:

1. Usuário seleciona um livro.
2. Sistema exibe tela de informações detalhadas do livro escolhido. Ao lado dos submenus “Sobre” e “Doadores” há um chamado “Compre”.
3. Usuário seleciona o submenu “Compre”.
4. O sistema lista links de lojas parceiras.
5. Usuário aciona link à sua escolha.
6. Sistema exibe mensagem explicando que usuário será redirecionado para o site parceiro e para verificar a edição do livro antes de comprar. Sistema abre link no navegador padrão do celular do usuário.
7. Fim do Caso de Uso.

Pós-condição: Usuário acessou por meio de seu navegador padrão o site parceiro.

Fluxos de extensão: N/A

Caso de Uso 11: Adicionar livros na plataforma

Descrição: Usuário adiciona livro que deseja doar ou trocar na plataforma. Especifica condição em que o livro se encontra, se deseja doar ou trocar, pode adicionar descrição e fotos, opcionalmente. Sistema deve vincular o livro adicionado a uma obra já existente no catálogo, ou criar uma nova entrada.

Ator: Usuário cadastrado

Pré-condição: Usuário está logado. Usuário já é cadastrado há algum tempo.

Fluxo principal:

1. Usuário aciona opção de adicionar livro.
2. Sistema pede permissão para acessar a câmera do dispositivo.
3. Usuário permite.
4. Sistema abre a câmera para poder ler código de barras do livro. Exibe também uma opção para permitir que o usuário entre manualmente com o código de barras.
5. Usuário escaneia o código de barras do livro.
6. Sistema identifica o livro correspondente no banco de dados. Exibe dados do livro como capa, nome, autor e resenha do livro.
7. Usuário confirma o livro.
8. Sistema exibe uma página mostrando categorias para preenchimento do cadastro da obra.
9. O usuário deve selecionar o estado em que o livro se encontra e se tem intenção de doar ou trocar o livro. Opcionalmente, o usuário pode também adicionar uma descrição e fotos.
10. Usuário aciona “Registrar”.
11. Sistema salva os dados no banco de dados e exibe confirmação de cadastro do livro.
12. Fim do Caso de Uso.

Pós-condição: O livro está cadastrado no banco de dados e vinculado à conta do usuário, disponível para que outros enviem solicitações.

Fluxos de extensão:

(a) Usuário não permite uso da câmera. (passo 3)

1. Sistema solicita cadastro manual do livro.
2. Sistema retorna ao passo 6 do fluxo principal.

(b) Usuário digita código de barras manualmente. (passo 5)

1. Sistema solicita que usuário entre com o código de barras via texto.
2. Usuário escreve código de barras.
3. Sistema retorna ao passo 6 do banco de dados.

(c) Sistema não consegue identificar a obra. (passo 6)

1. Sistema exibe uma mensagem avisando que não conseguiu encontrar um livro correspondente no banco de dados. Sistema exibe formulário de cadastro da obra, que será enviado à moderação do aplicativo para aprovação.
2. Fim do Caso de Uso.

(d) Usuário não adiciona mensagem e/ou não adiciona fotos. (passo 9)

1. Sistema retorna ao passo 10 do fluxo principal.

Caso de Uso 12: Responder solicitação

Descrição: Usuário analisa solicitação de outra pessoa para doação ou troca de livro, podendo aceitar ou recusar o pedido.

Ator: Usuário cadastrado

Pré-condição: Usuário está logado. Usuário recebe uma solicitação de doação ou troca para um livro que adicionou ao seu perfil.

Fluxo principal:

1. Usuário aciona notificações no menu de navegação.
2. Sistema exibe tela com todas as solicitações referentes ao usuário, seja como doador ou receptor na transação. Cada solicitação possui uma tag associada de acordo com o estado da transação, variando entre “novo”, “aceito” e “fechado”. O usuário tem como opções ver todas as mensagens, no menu superior, e para cada solicitação, pode pedir para ver mais detalhes da solicitação ou enviar uma mensagem para o outro usuário correspondente.
3. Usuário aciona “Ver solicitação” para um dos pedidos.
4. Sistema exibe resumo da solicitação, com foto, nome e local do usuário correspondente, bem como um link para seu perfil. Também são exibidos o livro solicitado, a mensagem de envio da solicitação, e opcionalmente livros que tenham sido oferecidos em troca. O usuário pode selecionar aceitar ou negar a solicitação.
5. Usuário escolhe aceitar solicitação.
6. O sistema exibe uma tela de confirmação e um botão para enviar uma mensagem para o usuário correspondente.
7. Fim do Caso de Uso.

Pós-condição: Usuário aceitou a solicitação de troca ou doação recebida.

Fluxos de extensão:

(a) Usuário rejeita solicitação. (passo 5)

1. Usuário aciona opção de negar solicitação.
2. Sistema atribui avaliação ao perfil do usuário correspondente.
3. Fim do Caso de Uso.

(b) Usuário não avalia experiência. (passo a)

1. Usuário não preenche campo de avaliação da experiência.
2. Fim do Caso de Uso.

Caso de Uso 13: Enviar ou ler mensagens

Descrição: Usuário acessa histórico de mensagens com outras pessoas, e pode ler e enviar mensagens.

Ator: Usuário cadastrado

Pré-condição: Usuário está logado. Deve haver uma transação ativa ou previamente fechada entre os dois usuários.

Fluxo principal:

1. Usuário aciona notificações no menu de navegação.

2. Sistema exibe todas as solicitações referentes ao usuário. O usuário pode escolher ver as notificações ou as mensagens.
3. Usuário aciona submenu mensagens.
4. Sistema redireciona o usuário para o histórico de mensagens dele, se houver alguma, separadas por destinatário.
5. Usuário seleciona uma conversa.
6. Sistema mostra o chat de mensagens com aquele destinatário, permite o envio de novas mensagens e leitura das antigas.
7. Fim do Caso de Uso.

Pós-condição: Chat de mensagens exibida.

Fluxos de extensão:

(a) Envio de mensagens e acordo com a solicitação. (passo 2)

1. Usuário, seleciona “Enviar mensagem” para o usuário referente àquela notificação.
2. Sistema retorna para o passo 6 do fluxo principal.

(b) Aviso de segurança na primeira mensagem. (passo 6)

1. Caso seja a primeira troca de mensagens com um usuário, o sistema exibe algumas mensagens de segurança e bem estar para a realização das trocas e envios de livros.
2. Sistema retorna ao passo 6 do fluxo principal.

Caso de Uso 14: Avaliar doador após transação

Descrição: Usuário tem a opção de dar uma nota e comentário sobre como o “doador” foi durante a transação do livro ou cancelamento.

Ator: Usuário cadastrado

Pré-condição: Usuário finalizou transação com “doador”, seja ela bem sucedida ou cancelada.

Fluxo principal:

1. Após a interação com o usuário, qualquer um dos dois participantes podem encerrar a transação, dada a conclusão da mesma ou não.
2. Usuário aciona “Finalizar transação”.
3. Sistema redireciona usuário para tela com campo de texto e espaço para inserir nota (seleciona nota de 0 a 5, sendo 0 ruim e 5 excelente), além de indicadores sobre estado do livro e se deu tudo certo com a transação.
4. Usuário preenche os campos e aciona “Finalizar transação”.
5. Sistema associa avaliação ao perfil do usuário avaliado.
6. Fim do Caso de Uso.

Pós-condição: Avaliação associada ao perfil do usuário avaliado.

Fluxos de extensão:

(a) Usuário opta por não avaliar. (passo 4)

1. Usuário escolhe por não avaliar a transação.

2. Transação é encerrada sem atribuição de avaliação.
3. Fim do Caso de Uso.

Caso de Uso 15: Visualizar perfil

Descrição: Usuário tem acesso a visualização de uma página de perfil.

Ator: Usuário cadastrado

Pré-condição: Usuário está logado.

Fluxo principal:

1. Usuário acessa página de perfil escolhida
2. Sistema redireciona para onde pode visualizar dados de perfil como: nome, dados de transações passadas (apenas no perfil próprio), avaliações da comunidade, doações, livros lidos, livros desejados e resenhas escritas.
3. Fim do Caso de Uso.

Pós-condição: Perfil é exibido.

Fluxo de extensão: N/A

Caso de Uso 16: Editar perfil próprio

Descrição: Usuário tem acesso a visualização da própria página de perfil.

Ator: Usuário cadastrado

Pré-condição: Usuário está logado.

Fluxo principal:

1. Usuário aciona “Meu perfil” no menu de navegação
2. Sistema redireciona para a tela correspondente à visualização de perfil
3. O usuário aciona a opção de edição de perfil
4. O sistema fornece ao usuário campos com suas informações e opções para alterá-las
5. Usuário modifica os campos desejados e confirma suas alterações
6. Fim do Caso de Uso.

Pós-condição: Perfil é alterado.

Fluxos de extensão:

(a) Usuário seleciona opção “Configurações”. (passo 5)

1. Usuário seleciona ícone de aba no canto superior direito.
2. Sistema exibe aba lateral contendo opções “Sobre”, “Configurações” e “Sair da conta”.
3. Usuário seleciona “Configurações”
4. Sistema exibe tela com diferentes opções de configuração do aplicativo
5. Usuário pode navegar e alterar opções livremente
6. Fim do Caso de Uso

(b) Usuário seleciona opção “Sair da conta”. (passo 5)

1. Usuário seleciona ícone de aba no canto superior direito.

2. Sistema exibe aba lateral contendo opções “Sobre”, “Configurações” e “Sair da conta”.
3. Usuário seleciona “Sair da conta”
4. Sistema exibe pop-up de confirmação
5. Usuário confirma logout
6. Sistema realiza logout do usuário do aparelho e o redireciona para as telas de onboarding.
7. Fim do Caso de Uso.

(c) Usuário nega confirmação de logout. (passo b)

1. Usuário aciona a opção de negar logout.
2. Sistema fecha pop-up.
3. Retorna ao passo 4 do fluxo principal.

(d) Usuário seleciona opção “Sobre”. (passo 5)

1. Usuário seleciona ícone de aba no canto superior direito.
2. Sistema exibe aba lateral contendo opções “Sobre”, “Configurações” e “Sair da conta”.
3. Usuário seleciona “Sobre”.
4. Sistema exibe tela com informações relevantes sobre o aplicativo, como versão e licenças.
5. Fim do Caso de Uso.

Caso de Uso 17: Realizar denúncia

Descrição: Usuário realiza denúncia de algum conteúdo no app, como resenhas, avaliações, imagens e perfis.

Ator: Usuário cadastrado

Pré-condição: Usuário está logado

Fluxo principal:

1. Usuário aciona botão de denúncia.
2. Sistema exibe mensagem de confirmação.
3. Usuário confirma denúncia.
4. Sistema notifica moderadores do conteúdo denunciado.

Pós-condição: Denúncia realizada e enviada aos moderadores.

Fluxos de extensão:

(a) Usuário não confirma denúncia. (passo 3)

1. Usuário cancela denúncia.
2. Fim do Caso de Uso.

Caso de Uso 18: Moderar banco de dados

Descrição: Moderador pode visualizar denúncias, apagar entradas do banco de dados e adicionar flags a usuários específicos.

Ator: Moderador

Pré-condição: N/A

Fluxo principal:

1. Moderador busca por entrada no banco de dados referente ao que deseja apagar ou marcar (*flag*).
2. Sistema exibe lista resultante da busca realizada.
3. Moderador opta por apagar entrada do banco de dados.
4. Sistema remove dados selecionados.
5. Fim do Caso de Uso

Pós-condição: Dados selecionados são apagados do banco de dados.

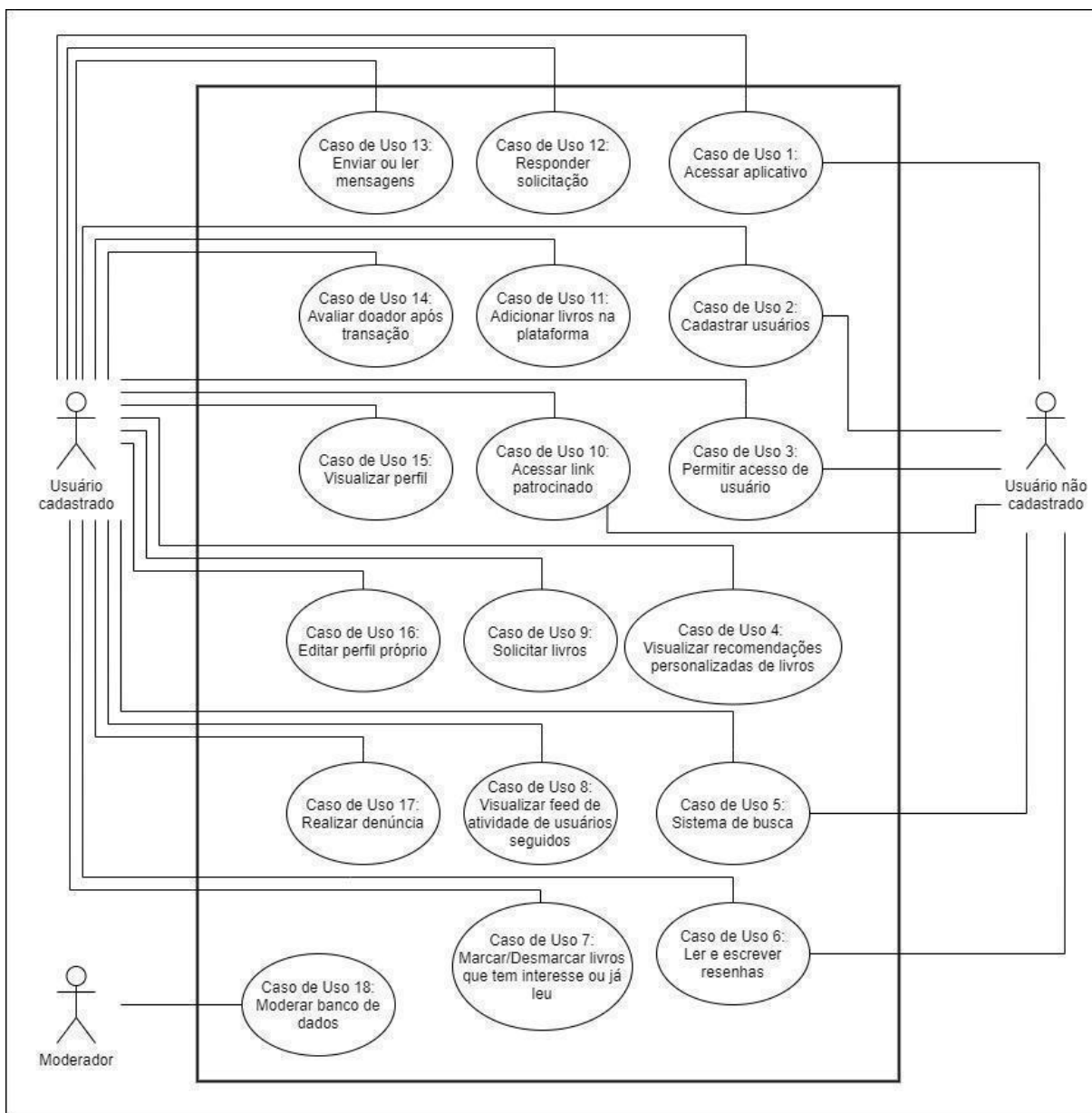
Fluxos de extensão:

(a) Moderador adiciona *flag* a um usuário. (passo 3)

1. Moderador acrescenta *flag* a um usuário para baní-lo.
2. Sistema proíbe acesso do usuário escolhido.
3. Fim do Caso de Uso.

Pode-se observar na Figura 2 o Diagrama de Casos de Uso, explicitando seus nomes e com quais atores se relacionam.

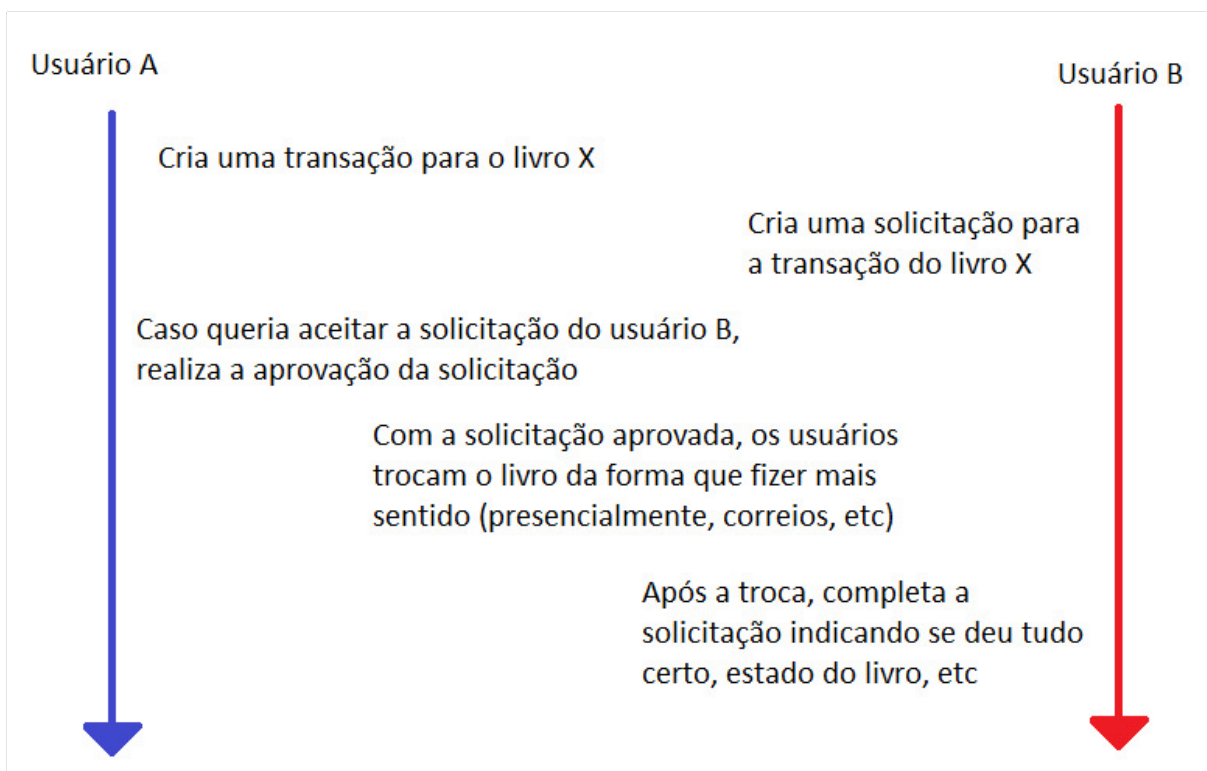
Figura 2 - Diagrama de casos de uso



Fonte: autores.

O fluxo principal de funcionamento do processo de solicitação e resposta das ofertas adicionadas pode ser resumido no diagrama visto na Figura 3, onde é possível acompanhar o funcionamento esperado do fluxo de transações da perspectiva de ambos os usuários participantes.

Figura 3 - Diagrama simplificado do fluxo de transações

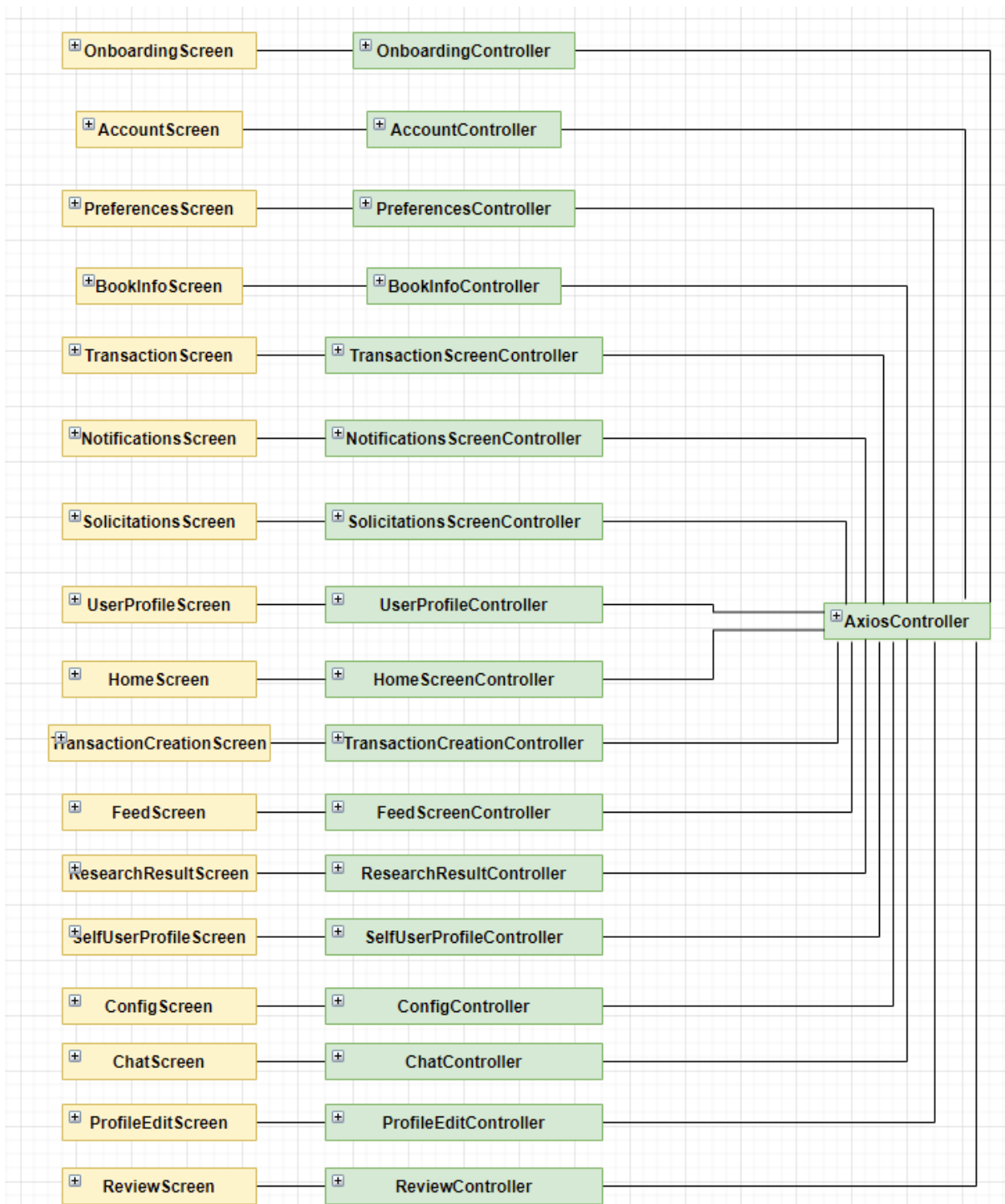


Fonte: autores.

3.2 Diagrama de Classes

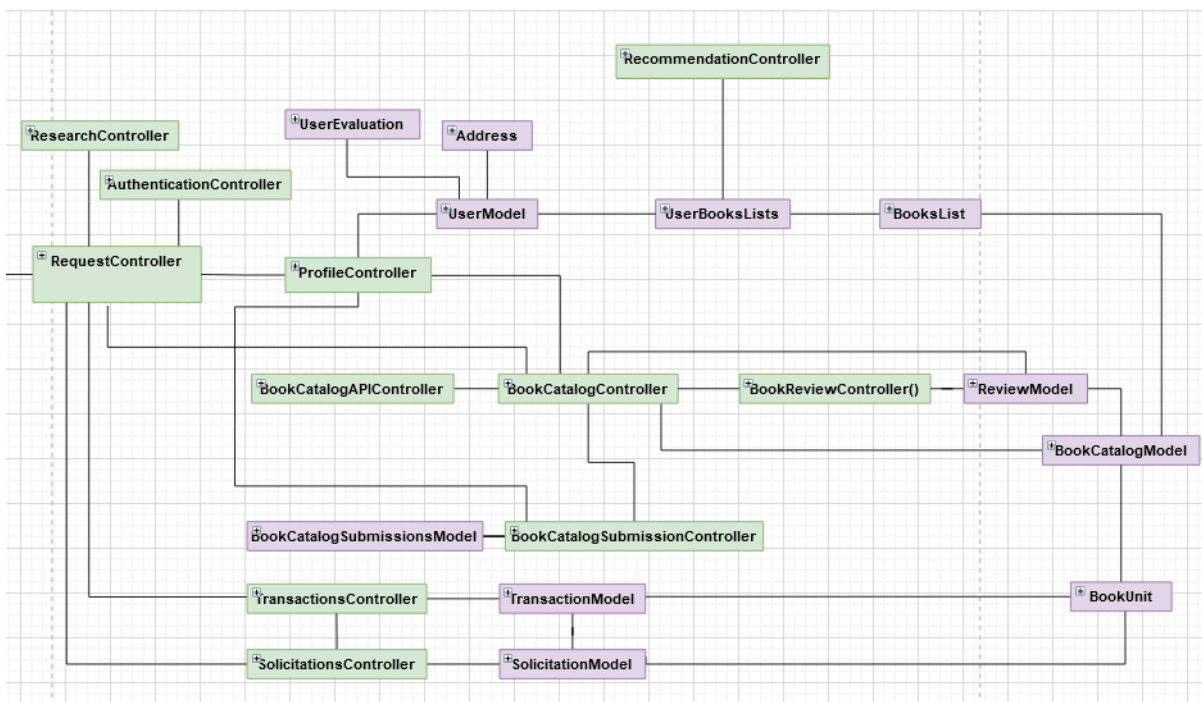
A partir dos Casos de Uso especificados na Seção 2.3, os Diagramas de Classes em alto nível são apresentados nas Figura 4 e 5, levando em consideração a utilização do padrão MVC (*Model-View-Controller*), utilizado no desenvolvimento do projeto. Estes diagramas contemplam divisão entre Cliente e Servidor.

Figura 4 - Diagrama de classes em alto nível - Cliente



Fonte: autores.

Figura 5 - Diagrama de classes - Servidor



Fonte: autores.

3.3 Plano de Testes de Validação

Nesta seção é apresentado o Plano de Testes de Validação do sistema desenvolvido. O teste de validação tem como objetivo assegurar o correto funcionamento da aplicação de acordo com todos os requisitos funcionais (mapeados em Casos de Uso) e não-funcionais do sistema. Os recursos necessários para realização dos testes são vistos na Tabela 2.

Tabela 2 - Recursos necessários para testes

Nome	Qtde.	Descrição
Servidor	1	Servidor com banco de dados desenvolvido que fornece informações ao aplicativo <i>mobile</i> durante seu uso.
<i>Smartphone</i> ou simulador	2	<i>Smartphone</i> ou simulador onde o aplicativo <i>mobile</i> será executado para realização dos testes.

Computador	1	Computador para realização dos testes de uso de moderadores.
Conexão à Internet	-	Acesso à Internet para os <i>smartphones</i> e computador para conexão ao servidor.
Analista de Testes	2	Responsável por realizar os testes propostos e registrar seus resultados.

Fonte: autores.

O Plano de Testes de Validação pode ser visto na Tabela 3, ordenado de acordo com os casos de uso elencados na seção 2.3.

Tabela 3 - Plano de testes de validação

Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
1: Acessar aplicativo	Valida o acesso ao aplicativo pelas tela de <i>onboarding</i> .	1. Abrir o aplicativo.	Sistema exibe tela de <i>onboarding</i> .
		2. Acionar opção de "próximo" em cada tela de <i>onboarding</i> .	Sistema exibe cada tela de <i>onboarding</i> disponível, e por fim uma tela com as opções de Login, Cadastro e Usar sem Cadastro.
		3. Selecionar opção de "Login" ou "Cadastro".	Sistema redireciona usuário para a tela adequada de acordo com sua opção.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
1: Acessar aplicativo (Extensão a)	Valida o acesso ao aplicativo pelas telas de <i>onboarding</i> , mas o usuário decide pular as telas introdutórias.	1. Abrir o aplicativo.	Sistema exibe tela de <i>onboarding</i> .
		2. Acionar opção de "pular".	Sistema exibe diretamente uma tela com as opções de Login, Cadastro e Usar sem Cadastro.

		3. Selecionar opção de "Login" ou "Cadastro".	Sistema redireciona usuário para a tela adequada de acordo com sua opção.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
1: Acessar aplicativo (Extensão b)	Valida entrada no aplicativo pela opção de acessar sem cadastro.	1. Abrir o aplicativo.	Sistema exibe tela de <i>onboarding</i> .
		2. Acionar opção de "próximo" em cada tela de <i>onboarding</i> .	Sistema exibe cada tela de <i>onboarding</i> disponível, e por fim uma tela com as opções de Login, Cadastro e Usar sem Cadastro.
		3. Selecionar opção de "Usar sem Cadastro".	Sistema redireciona usuário para o menu principal.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
2: Cadastrar usuários	Valida o cadastro correto do usuário no sistema, sem uso de redes sociais.	1. Selecionar opção de "Cadastro".	Tela de cadastro é exibida com os campos: nome, e-mail, senha, bem como a aceitação dos termos de uso e botões para realizar cadastro por meio de redes sociais.
		2. Preencher respectivamente os campos da tela de cadastro com "Teste 1", "teste1@teste.com", "!!TEsTe1!!" e aceitar termos de uso.	Campos preenchidos.

		3. Selecionar "Criar uma conta".	Tela de cadastro para informações adicionais é exibida com os campos: foto, idade e região de moradia.
		4. Preencher respectivamente os campos da tela de cadastro para informações adicionais com foto de teste, 20, São Paulo.	Campos preenchidos.
		5. Selecionar "Continuar".	Tela com uma lista de gêneros literários é exibida.
		6. Selecionar quaisquer gêneros literários e acionar "próximo".	Tela com lista de livros relacionados aos gêneros literários escolhidos é exibida.
		7. Selecionar quaisquer livros e escolher entre as opções "Quero ler" ou "Já li", e acionar "próximo".	Tela com confirmação de cadastro é exibida. Depois, usuário é redirecionado ao menu principal.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
2: Cadastrar usuários (Extensão a)	Valida o cadastro correto do usuário no sistema fazendo uso de redes sociais.	1. Selecionar opção de "Cadastro".	Tela de cadastro é exibida com os campos: nome, e-mail, senha, bem como a aceitação dos termos de uso e botões para realizar cadastro por meio de redes sociais.
		2. Selecionar um dos botões de redes sociais.	Mensagem de permissão da rede social é exibida.

		3. Confirmar permissão.	Tela de cadastro é novamente exibida para aceite dos termos de uso.
		4. Aceitar termos de uso e selecionar "Criar uma conta".	Tela de cadastro para informações adicionais é exibida com os campos: foto, idade e região de moradia.
		5. Preencher respectivamente os campos da tela de cadastro para informações adicionais com foto de teste, 20, São Paulo.	Campos preenchidos.
		6. Selecionar "Continuar".	Tela com uma lista de gêneros literários é exibida.
		7. Selecionar quaisquer gêneros literários e acionar "próximo".	Tela com lista de livros relacionados aos gêneros literários escolhidos é exibida.
		8. Selecionar quaisquer livros e escolher entre as opções "Quero ler" ou "Já li", e acionar "próximo".	Tela com confirmação de cadastro é exibida. Depois, usuário é redirecionado ao menu principal.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
2: Cadastrar usuários (Extensão a1)	Valida erro no processo de cadastro por rede sociais quando o usuário não confirma permissão de uso da rede social.	1. Selecionar opção de "Cadastro".	Tela de cadastro é exibida com os campos: nome, e-mail, senha, bem como a aceitação dos termos de uso e botões para realizar cadastro por meio de redes sociais.

		2. Selecionar um dos botões de redes sociais.	Mensagem de permissão da rede social é exibida.
		3. Negar permissão.	Mensagem de erro da associação com rede social é exibida.
		4. Selecionar "OK".	Exibe novamente tela de cadastro.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
2: Cadastrar usuários (Extensão b)	Valida erro no processo de cadastro ao entrar com um campo inválido.	1. Selecionar opção de "Cadastro".	Tela de cadastro é exibida com os campos: nome, e-mail, senha, bem como a aceitação dos termos de uso e botões para realizar cadastro por meio de redes sociais.
		2. Preencher respectivamente os campo da tela de cadastro em um formato incorreto com " /"~`", "@teste", "abc".	Mensagem de erro é exibida alertando que cada campo está em formato inválido.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
2: Cadastrar usuários (Extensão c)	Valida o processo de pular a escolha de gêneros literários favoritos.	1. Selecionar opção de "Cadastro".	Tela de cadastro é exibida com os campos: nome, e-mail, senha, bem como a aceitação dos termos de uso e botões para realizar cadastro por meio de redes sociais.

		2. Preencher respectivamente os campos da tela de cadastro com "Teste 1c", "teste1c@teste.com", "!!TEsTe1c!!" e aceitar termos de uso.	Campos preenchidos.
		3. Selecionar "Criar uma conta".	Tela de cadastro para informações adicionais é exibida com os campos: foto, idade e região de moradia.
		4. Preencher respectivamente os campos da tela de cadastro para informações adicionais com foto de teste, 20, São Paulo.	Campos preenchidos.
		5. Selecionar "Continuar".	Tela com uma lista de gêneros literários é exibida.
		6. Selecionar "Pular".	Tela com lista de livros é exibida.
		7. Selecionar quaisquer livros e escolher entre as opções "Quero ler" e "Já li", e acionar "próximo".	Tela com confirmação de cadastro é exibida. Depois, usuário é redirecionado ao menu principal.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
2: Cadastrar usuários (Extensão d)	Valida o processo de pular a seleção de livros recomendados.	1. Selecionar opção de "Cadastro".	Tela de cadastro é exibida com os campos: nome, e-mail, senha, bem como a aceitação dos termos de uso e botões para realizar

			cadastro por meio de redes sociais.
		2. Preencher respectivamente os campos da tela de cadastro com "Teste 1d", "teste1d@teste.com", "!!TEsTe1d!!" e aceitar termos de uso.	Campos preenchidos.
		3. Selecionar "Criar uma conta".	Tela de cadastro para informações adicionais é exibida com os campos: foto, idade e região de moradia.
		4. Preencher respectivamente os campos da tela de cadastro para informações adicionais com foto de teste, 20, São Paulo.	Campos preenchidos.
		5. Selecionar "Continuar".	Tela com uma lista de gêneros literários é exibida.
		6. Selecionar quaisquer gêneros literários e acionar "próximo".	Tela com lista de livros relacionados aos gêneros literários escolhidos é exibida.
		7. Selecionar "Pular".	Tela com confirmação de cadastro é exibida. Depois, usuário é redirecionado ao menu principal.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado

2: Cadastrar usuários (Extensão e)	Valida erro no processo de cadastro ao não aceitar os termos de uso.	1. Selecionar opção de "Cadastro".	Tela de cadastro é exibida com os campos: nome, e-mail, senha, bem como a aceitação dos termos de uso e botões para realizar cadastro por meio de redes sociais.
		2. Preencher respectivamente os campos da tela de cadastro com "Teste 1e", "teste1e@teste.com", "!!TEsTe1e!!" e não aceitar termos de uso.	Campos preenchidos.
		3. Selecionar "Criar uma conta".	Mensagem de erro é exibida alertando que é necessário aceitar os termos de uso.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
2: Cadastrar usuários (Extensão f)	Valida erro que aquele e-mail já foi cadastrado previamente.	1. Selecionar opção de "Cadastro".	Tela de cadastro é exibida com os campos: nome, e-mail, senha, bem como a aceitação dos termos de uso e botões para realizar cadastro por meio de redes sociais.
		2. Preencher respectivamente os campos da tela de cadastro com "Teste 1", "teste1@teste.com", "!!TEsTe1!!" e não aceitar termos de uso. E-mail já	Campos preenchidos.

		deve ter sido cadastrado previamente.	
		3. Selecionar "Criar uma conta".	Mensagem de erro é exibida informando que aquele e-mail já foi cadastrado.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
2: Cadastrar usuários (Extensão g)	Valida erro sobre a falta de entrada da região de moradia, obrigatória para o cadastro.	1. Selecionar opção de "Cadastro".	Tela de cadastro é exibida com os campos: nome, e-mail, senha, bem como a aceitação dos termos de uso e botões para realizar cadastro por meio de redes sociais.
		2. Preencher respectivamente os campos da tela de cadastro com "Teste 1g", "teste1g@teste.com", "!!TEsTe1g!!" e não aceitar termos de uso.	Campos preenchidos.
		3. Selecionar "Criar uma conta".	Tela de cadastro para informações adicionais é exibida com os campos: foto, idade e região de moradia.
		4. Preencher respectivamente os campos da tela de cadastro para informações adicionais com foto de teste, 20. Não preencher campo de região de moradia.	Campos preenchidos, exceto o de região de moradia.

		5. Selecionar "Continuar".	Mensagem de erro é exibida informando a obrigatoriedade de informar região de moradia.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
3: Permitir acesso de usuário	Valida processo de login do usuário no aplicativo, sem uso de redes sociais.	1. Selecionar opção de "Login".	Tela de login é exibida com os campos e-mail e senha, além de botões para realizar login por meio de redes sociais.
		2. Inserir e-mail e senha já cadastrados ("teste1@teste.com", "!!TEsTe1!!") e selecionar "Entrar".	Login realizado com sucesso, tela do menu principal é exibida.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
3: Permitir acesso de usuário (Extensão a)	Valida processo de login do usuário no aplicativo, com uso de redes sociais.	1. Selecionar opção de "Login".	Tela de login é exibida com os campos e-mail e senha, além de botões para realizar login por meio de redes sociais.
		2. Selecionar uma rede social previamente cadastrada.	Login realizado com sucesso, tela do menu principal é exibida.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado

3: Permitir acesso de usuário (Extensão b)	Valida erro na autenticação do usuário.	1. Selecionar opção de "Login".	Tela de login é exibida com os campos e-mail e senha, além de botões para realizar login por meio de redes sociais.
		2. Inserir e-mail e senha não cadastrados ("abc", "123") e selecionar "Entrar".	Mensagem de erro é exibida alertando da falha na autenticação e tela de login é novamente exibida.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
4: Visualizar recomendações personalizadas de livros	Validar exibição de sugestões de livros baseado nos gostos do usuário.	1. Acessar menu principal.	Exibe subseções de livros: uma lista de livros recomendados para o usuário (se logado), uma lista de livros novos, e uma lista de livros populares.
		2. Selecionar "Expandir" em uma das subseções. Retornar e repetir o teste com as outras.	Exibe tela com mais livros pertencentes àquele grupo de livros, repetido para cada grupo.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
5: Sistema de busca - livro	Validar uso adequado do sistema de busca do aplicativo ao buscar por livros.	1. Acessar barra de busca e escrever "Harry Potter".	Tela exibe lista de livros com resultados relacionados ao termo digitado e filtros aplicados.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado

5: Sistema de busca - autor	Validar uso adequado do sistema de busca do aplicativo ao buscar por autores.	1. Acessar barra de busca e escrever "Tolkien".	Tela exibe lista de livros com resultados relacionados ao termo digitado e filtros aplicados.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
5: Sistema de busca (Extensão a)	Validar uso adequado do sistema de busca do aplicativo quando não há resultados encontrados.	1. Acessar barra de busca e escrever "!@#\$\$%^" ou fazer uso de filtros que impossibilite resultados válidos para busca.	Tela exibe mensagem informando que não foram encontrados livros com as características especificadas.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
5: Sistema de busca (Extensão b)	Validar uso adequado do sistema de busca do aplicativo para perfis de usuários.	1. Acessar barra de busca e escrever o nome de um usuário cadastrado ("Teste").	Tela exibe lista de perfis de usuários com nomes relacionados ao termo digitado.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
5: Sistema de busca (Extensão c)	Validar uso adequado do sistema de busca do aplicativo apenas por filtros.	1. Acessar barra de busca e selecionar filtro de gênero literário "Romance" sem digitar nada.	Tela exibe lista de livros com resultados relacionados ao gênero escolhido.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
6: Ler e escrever resenhas	Validar uso adequado da leitura de resenhas.	1. Acessar página de informações de um livro	Tela exibe informações sobre o livro selecionado. A aba selecionada é a "Sobre".

		qualquer e deslizar para baixo na aba "Sobre".	
		2. Ler resenhas do livro escolhido.	Resenhas exibidas em uma lista.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
6: Ler e escrever resenhas (Extensão a)	Validar uso adequado da escrita de resenhas.	1. Selecionar botão "+" no menu principal e escolher "Escrever resenha". Usuário deve estar logado.	Exibe interface para escolha do livro para o qual deseja-se escrever a resenha.
		2. Escolher o livro para qual deseja-se escrever a resenha.	Exibe interface para escrever a resenha.
		3. Escrever "Teste de escrita da resenha." e selecionar publicar.	Confirmação de publicação da resenha é exibida e usuário é redirecionado ao menu principal.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
6: Ler e escrever resenhas (Extensão b)	Validar erro de acesso para usuários não logados tentando escrever resenhas.	1. Selecionar botão "+" no menu principal e escolher "Escrever resenha". Usuário não está logado.	Sistema exibe mensagem de erro alertando que o usuário deve estar logado para escrever resenhas.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
6: Ler e escrever	Validar situação em que usuário já escreveu uma	1. Selecionar botão "+" no menu principal e escolher	Exibe interface para escolha do livro para o qual deseja-se escrever a resenha.

resenhas (Extensão c)	resenha para aquele livro e deseja editá-la.	"Escrever resenha". Usuário deve estar logado.	
		2. Escolher o mesmo livro que foi selecionado previamente no caso de teste 6 extensão a, isto é, um livro com uma resenha já publicada pelo usuário.	Mensagem é exibida alertando o usuário que já escreveu uma resenha para aquele livro. Oferece opções de editar ou excluir a resenha previamente cadastrada.
		3. Selecionar a opção de edição de resenha.	Sistema exibe interface com a resenha previamente escrita de forma editável.
		4. Adicionar ao final da resenha já escrita "Teste de edição." e selecionar publicar.	Confirmação de publicação da resenha é exibida e usuário é redirecionado ao menu principal.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
6: Ler e escrever resenhas (Extensão c1)	Validar situação em que usuário já escreveu uma resenha para aquele livro e deseja apagá-la.	1. Selecionar botão "+" no menu principal e escolher "Escrever resenha". Usuário deve estar logado.	Exibe interface para escolha do livro para o qual deseja-se escrever a resenha.
		2. Escolher o mesmo livro que foi selecionado previamente no caso de teste 6 extensão a, isto é, um livro com uma resenha já publicada pelo usuário.	Mensagem é exibida alertando o usuário que já escreveu uma resenha para aquele livro. Oferece opções de editar ou excluir a resenha previamente cadastrada.

		3. Selecionar a opção de exclusão de resenha.	Sistema exibe mensagem de confirmação que a resenha será apagada.
		4. Concordar com a exclusão da resenha.	Sistema exibe mensagem confirmando a exclusão da resenha. Usuário é redirecionado ao menu principal.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
7: Marcar/Desmarcar livros que tem interesse ou já leu - Quero ler	Validar marcação de um livro como "Quero ler".	1. Acessar página de informações de um livro qualquer.	Tela exibe informações sobre o livro selecionado. A aba selecionada é a "Sobre".
		2. Selecionar campo indicando que deseja marcar o livro.	São exibidas as opções "Quero ler", "Já li" e "Cancelar".
		3. Selecionar opção "Quero ler".	Volta a ser exibida apenas a tela de "Sobre" do livro. Livro fica marcado na seção "Quero ler" do perfil do usuário.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
7: Marcar/Desmarcar livros que tem interesse ou já leu - Já li	Validar marcação de um livro como "Já li".	1. Acessar página de informações de um livro qualquer.	Tela exibe informações sobre o livro selecionado. A aba selecionada é a "Sobre".
		2. Selecionar campo indicando que deseja marcar o livro.	São exibidas as opções "Quero ler", "Já li" e "Cancelar".

		3. Selecionar opção "Já li".	Volta a ser exibida apenas a tela de "Sobre" do livro. Livro fica marcado na seção "Já li" do perfil do usuário.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
7: Marcar/Desmarcar livros que tem interesse ou já leu (Extensão a)	Validar cancelamento da opção de marcação antes de selecionar opção de como marcar.	1. Acessar página de informações de um livro qualquer.	Tela exibe informações sobre o livro selecionado. A aba selecionada é a "Sobre".
		2. Selecionar campo indicando que deseja marcar o livro.	São exibidas as opções "Quero ler", "Já li" e "Cancelar".
		3. Selecionar opção "Cancelar".	Volta a ser exibida apenas a tela de "Sobre" do livro. Livro não recebe marcação.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
7: Marcar/Desmarcar livros que tem interesse ou já leu (Extensão b)	Validar operação de desmarcar livro.	1. Acessar página de informações de um livro previamente marcado.	Tela exibe informações sobre o livro selecionado. A aba selecionada é a "Sobre".
		2. Selecionar campo indicando que deseja marcar o livro.	Livro é desmarcado e retirado da seção correspondente do perfil do usuário.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado

8: Visualizar feed de atividade de usuários seguidos	Validar acesso ao feed de resenhas.	1. Acessar submenu do feed do aplicativo.	Tela exibe em ordem de mais recente para mais antiga resenhas de usuários seguidos.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
9: Solicitar livros	Validar processo de solicitar doação de livro fazendo uso de mensagem no envio e oferta de troca.	1. Abrir tela de informações de um livro qualquer.	Tela exibida com informações do livro é exibida.
		2. Selecionar aba "Doadores"	Lista de doadores é exibida.
		3. Selecionar "Ver livro"	Sistema apresenta detalhes da oferta selecionada.
		4. Selecionar "Solicitar esse livro".	Tela exibe resumo da oferta e opções de adicionar mensagem e oferecer livro em troca.
		5. Selecionar "Adicionar mensagem" e escrever "Teste de solicitação 1".	Tela resumo volta a ser exibida, mas agora com <i>preview</i> da mensagem escrita.
		6. Selecionar "Oferecer livro em troca" e escolher um livro qualquer previamente cadastrado.	Tela resumo volta a ser exibida, mas agora também com <i>preview</i> da capa do livro oferecido em troca.
		7. Selecionar "Enviar solicitação".	Tela é exibida confirmando o envio. Doador recebeu solicitação.

		8. Selecionar "Continuar procurando livros".	Tela do menu principal é exibida.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
9: Solicitar livros (Extensão a)	Validar filtros para escolher oferta.	1. Abrir tela de informações de um livro qualquer.	Tela exibida com informações do livro é exibida.
		2. Selecionar aba "Doadores"	Lista de doadores é exibida.
		3. Aplicar um filtro qualquer na lista.	Lista é atualizada de acordo com os parâmetros selecionados.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
9: Solicitar livros (Extensão b)	Validar processo de solicitar doação de livro sem fazer uso de mensagem no envio e oferta de troca.	1. Abrir tela de informações de um livro qualquer.	Tela exibida com informações do livro é exibida.
		2. Selecionar aba "Doadores"	Lista de doadores é exibida.
		3. Selecionar "Ver livro"	Sistema apresenta detalhes da oferta selecionada.
		4. Selecionar "Solicitar esse livro".	Tela exibe resumo da oferta e opções de adicionar mensagem e oferecer livro em troca.
		5. Selecionar "Enviar solicitação".	Tela é exibida confirmando o envio. Doador recebeu solicitação.

		6. Selecionar "Continuar procurando livros".	Tela do menu principal é exibida.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
10: Acessar link patrocinado	Validar acesso correto de links patrocinados.	1. Abrir tela de informações de um livro qualquer.	Tela exibida com informações do livro é exibida.
		2. Selecionar aba "Compre".	Lista de lojas parceiras é exibida.
		3. Selecionar o primeiro link disponível.	Link é aberto no navegador padrão do dispositivo.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
11: Adicionar livros na plataforma	Validar processo de cadastro de um livro na plataforma com o uso de câmera para scan do código de barras e preenchendo todos os campos de cadastro.	1. Selecionar "+" e opção de adicionar livro.	Exibida mensagem pedindo permissão para acessar a câmera do dispositivo.
		2. Aceitar permissão.	Câmera do dispositivo é aberta dentro do aplicativo para escanear código de barras. Há também uma opção de adicionar livro manualmente.
		3. Realizar scan do código de barras do livro apontando e aproximando a câmera do código de barras.	Tela é exibida com informações do livro.
		4. Confirmar que este é o livro correto.	Tela é exibida para cadastro do livro na plataforma.

		5. Selecionar quaisquer opções. Exemplo de preenchimento: livro em estado "bom" e intenção de "doar". Preencher a seção de comentários com "Teste 11. Livro em bom estado." e registrar uma foto do livro. Finalmente, selecionar "Registrar".	Tela exibe confirmação de cadastro do livro. Obra agora está vinculada ao perfil do usuário e disponível para solicitação.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
11: Adicionar livros na plataforma (Extensão a)	Validar processo de cadastro de um livro na plataforma sem o uso de câmera para scan do código de barras e não preenchendo todos os campos de cadastro.	1. Selecionar "+" e opção de adicionar livro.	Exibida mensagem pedindo permissão para acessar a câmera do dispositivo.
		2. Negar permissão.	Câmera do dispositivo não é aberta dentro do aplicativo para escanear código de barras. É exibida a opção de adicionar livro manualmente.
		3. Preencher manualmente o campo do isbn do livro.	Tela é exibida com informações do livro.
		4. Confirmar que este é o livro correto.	Tela é exibida para cadastro do livro na plataforma.
		5. Selecionar quaisquer opções. Exemplo de preenchimento: livro em estado "bom" e intenção de "doar". Não preencher a seção de comentários e não registrar uma foto do livro.	Tela exibe confirmação de cadastro do livro. Obra agora está vinculada ao perfil do usuário e disponível para solicitação.

		Finalmente, selecionar "Registrar".	
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
11: Adicionar livros na plataforma (Extensão b)	Validar situação em que sistema não consegue identificar livro no banco de dados atual.	1. Selecionar "+" e opção de adicionar livro.	Exibida mensagem pedindo permissão para acessar a câmera do dispositivo.
		2. Aceitar permissão.	Câmera do dispositivo é aberta dentro do aplicativo para escanear código de barras. Há também uma opção de adicionar livro manualmente.
		3. Realizar scan do código de barras do livro apontando e aproximando a câmera do código de barras.	Mensagem é exibida alertando que a edição não foi encontrada. Um e-mail de contato é mostrado para que o usuário peça por fora do aplicativo a adição do livro na plataforma.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
12: Responder solicitação	Validar processo de aceitar transação proposta por outro usuário.	1. Acionar seção de notificações no menu de navegação.	Tela é exibida com todas as notificações do usuário.
		2. Selecionar "Ver solicitação" para a primeira oferta recebida ainda aberta.	Tela é exibida com informações resumindo aquela proposta de transação e botões para aceitar ou negar a solicitação.

		3. Aceitar transação.	Tela exibe mensagem de confirmação e botão para enviar mensagens para o usuário correspondente.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
12: Responder solicitação (Extensão a)	Validar processo de rejeitar transação proposta por outro usuário.	1. Acionar seção de notificações no menu de navegação.	Tela é exibida com todas as notificações do usuário.
		2. Selecionar "Ver solicitação" para a primeira oferta recebida ainda aberta.	Tela é exibida com informações resumindo aquela proposta de transação e botões para aceitar ou negar a solicitação.
		3. Rejeitar transação.	Tela exibe mensagem de confirmação e campo de avaliação de experiência.
		4. Avaliar experiência.	Avaliação é vinculada ao perfil do usuário correspondente.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
12: Responder solicitação (Extensão b)	Validar processo de rejeitar transação proposta por outro usuário.	1. Acionar seção de notificações no menu de navegação.	Tela é exibida com todas as notificações do usuário.
		2. Selecionar "Ver solicitação" para a primeira oferta recebida ainda aberta.	Tela é exibida com informações resumindo aquela proposta de

			transação e botões para aceitar ou negar a solicitação.
		3. Rejeitar transação.	Tela exibe mensagem de confirmação e campo de avaliação de experiência.
		4. Não avaliar experiência.	Nenhuma avaliação recebida.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
13: Enviar ou ler mensagens	Validar processo de enviar mensagens. Mensagem de segurança é exibida no topo da conversa.	1. Acionar seção de notificações no menu de navegação.	Tela é exibida com todas as notificações do usuário.
		2. Acionar submenu de mensagens.	Tela com histórico de mensagens é exibida, separada por destinatário.
		3. Selecionar a primeira conversa da lista.	Tela com conversas passadas é exibida. É possível enviar e ler novas mensagens. Ao topo da conversa, mensagem de segurança é exibida.
		4. Enviar mensagem "Teste 13 Envio"	Mensagem é enviada.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
13: Enviar ou ler mensagens	Validar processo de receber mensagens.	1. Acionar seção de notificações no menu de navegação.	Tela é exibida com todas as notificações do usuário.

- Recebimento Mensagem de

	segurança é exibida no topo da conversa.	2. Acionar submenu de mensagens.	Tela com histórico de mensagens é exibida, separada por destinatário.
		3. Selecionar a primeira conversa da lista.	Tela com conversas passadas é exibida. É possível enviar e ler novas mensagens. Ao topo da conversa, mensagem de segurança é exibida.
		4. Verificar recebimento da mensagem enviada no caso de teste anterior.	Mensagem é exibida adequadamente na tela.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
13: Enviar ou ler mensagens (Extensão a)	Validar processo de chegar à tela de conversa por meio de acesso direto a partir de notificações. Mensagem de segurança é exibida no topo da conversa.	1. Acionar seção de notificações no menu de navegação.	Tela é exibida com todas as notificações do usuário.
		2. Selecionar envio de mensagem na primeira notificação de transação recebida ainda em aberto.	Tela com conversas passadas é exibida. É possível enviar e ler novas mensagens. Ao topo da conversa, mensagem de segurança é exibida.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
14: Avaliar doador após transação	Validar avaliação de usuário.	1. Encerrar transação.	Tela exibe mensagem de confirmação e campo de avaliação de experiência.

		2. Avaliar experiência com nota 3 e comentar "Teste 14".	Avaliação é vinculada ao perfil do usuário correspondente.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
14: Avaliar doador após transação (Extensão a)	Validar opção de pular avaliação de usuário.	1. Encerrar transação.	Tela exibe mensagem de confirmação e campo de avaliação de experiência.
		2. Não avaliar a experiência.	Nenhuma avaliação é vinculada ao perfil do usuário correspondente.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
15: Visualizar perfil	Validar acesso a perfil próprio e de outros usuários.	1. Selecionar opção de "Perfil" no menu de navegação.	Tela exibe perfil próprio do usuário.
		2. Selecionar opção de "Início" no menu de navegação.	Tela exibe menu principal.
		3. Realizar busca na barra de busca pelo usuário "Teste 1c" e selecionar usuário de mesmo nome.	Tela exibe perfil do usuário Teste 1c.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
16: Editar perfil próprio	Validar edição do perfil próprio do usuário.	1. Selecionar opção de "Perfil" no menu de navegação.	Tela exibe perfil próprio do usuário.

		2. Abrir aba lateral de configurações.	Aba lateral aparece com opções "Edição de perfil", "Configurações", "Sobre" e "Logout".
		3. Acionar opção de edição de perfil.	Tela é exibida com campos com informações do usuário e opções para alterá-las.
		4. Editar campo idade para 50 e confirmar edições.	Tela do perfil do usuário volta a ser exibida, agora mostrando informações atualizadas.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
16: Editar perfil próprio (Extensão a)	Validar acesso às configurações do aplicativo.	1. Selecionar opção de "Perfil" no menu de navegação.	Tela exibe perfil próprio do usuário.
		2. Abrir aba lateral de configurações.	Aba lateral aparece com opções "Edição de perfil", "Configurações", "Sobre" e "Logout".
		3. Acionar opção de configurações.	Tela é exibida com configurações diversas e opções para alterá-las.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
16: Editar perfil próprio (Extensão b)	Validar logout do aplicativo.	1. Selecionar opção de "Perfil" no menu de navegação.	Tela exibe perfil próprio do usuário.

		2. Abrir aba lateral de configurações.	Aba lateral aparece com opções "Edição de perfil", "Configurações", "Sobre" e "Logout".
		3. Acionar opção de Logout.	Mensagem é exibida pedindo confirmação de logout para o usuário.
		4. Confirmar Logout.	Usuário é deslogado da plataforma. Tela de onboarding é exibida.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
16: Editar perfil próprio (Extensão c)	Validar cancelamento do logout do aplicativo.	1. Selecionar opção de "Perfil" no menu de navegação.	Tela exibe perfil próprio do usuário.
		2. Abrir aba lateral de configurações.	Aba lateral aparece com opções "Edição de perfil", "Configurações", "Sobre" e "Logout".
		3. Acionar opção de Logout.	Mensagem é exibida pedindo confirmação de logout para o usuário.
		4. Selecionar "Cancelar".	Mensagem de confirmação é fechada e usuário continuar logado.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado

16: Editar perfil próprio (Extensão d)	Validar acesso à seção "Sobre".	1. Selecionar opção de "Perfil" no menu de navegação.	Tela exibe perfil próprio do usuário.
		2. Abrir aba lateral de configurações.	Aba lateral aparece com opções "Edição de perfil", "Configurações", "Sobre" e "Logout".
		3. Acionar opção "Sobre".	Tela com informações relevantes sobre o aplicativo, como versão e licenças, é exibida.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
17: Realizar denúncia	Validar processo de denúncia contra um usuário.	1. Em algum conteúdo publicado por um usuário, como a foto de um livro cadastrado, acionar o botão de denúncia.	Mensagem de confirmação é exibida.
		2. Confirmar denúncia.	Mensagem é fechada e moderadores são notificados da denúncia.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
17: Realizar denúncia (Extensão a)	Validar cancelamento do processo de denúncia contra um usuário.	1. Em algum conteúdo publicado por um usuário, como a foto de um livro cadastrado, acionar o botão de denúncia.	Mensagem de confirmação é exibida.

		2. Cancelar denúncia.	Mensagem é fechada e moderadores não são notificados da denúncia.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
18: Moderar banco de dados	Validar moderação do banco de dados para excluir itens.	1. Buscar por denúncias realizadas.	Lista de resultados é exibida.
		2. Apagar item denunciado do banco de dados.	Item é removido.
Caso de uso	Descrição	Passos (Entradas)	Resultado esperado
18: Moderar banco de dados (Extensão a)	Validar moderação do banco de dados para aplicar flag a um usuário.	1. Buscar por usuário "Teste 1d"	Lista de resultados é exibida.
		2. Aplicar flag ao usuário.	Usuário é banido por tempo determinado e tem acesso proibido ao aplicativo.

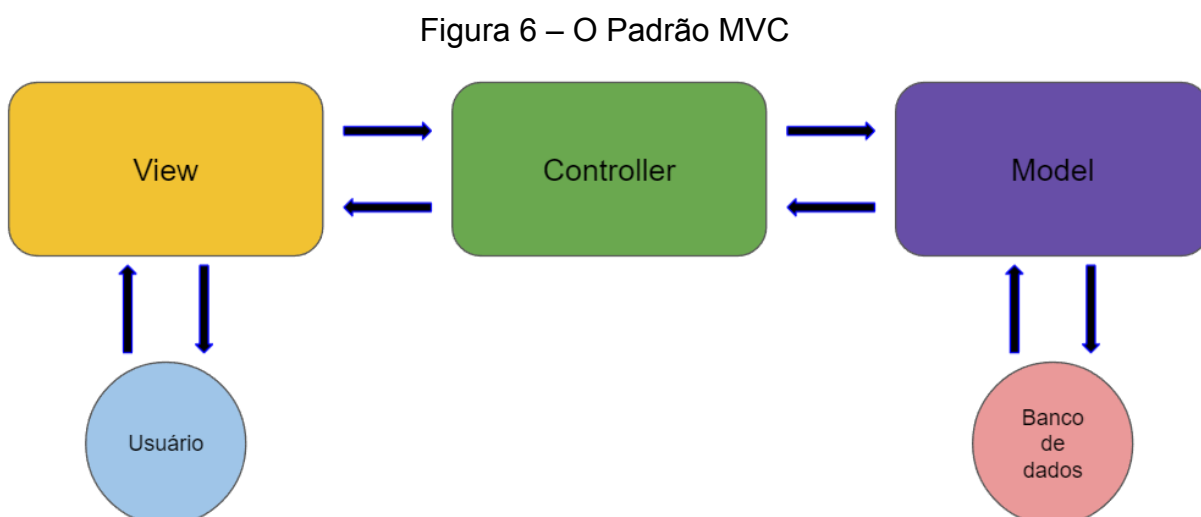
Fonte: autores.

4. PROJETO

Neste capítulo são apresentadas uma descrição dos padrões e APIs definidos para o projeto e a arquitetura do sistema baseado no detalhamento do Diagrama de Classes definido anteriormente para implementação do projeto com o objetivo de melhor compreender a estrutura do sistema. Os Diagramas de Classes detalhados podem ser conferidos no Apêndice B, completos e também em partes menores para melhor visualização.

4.1 Descrição de Padrões e APIs do Projeto

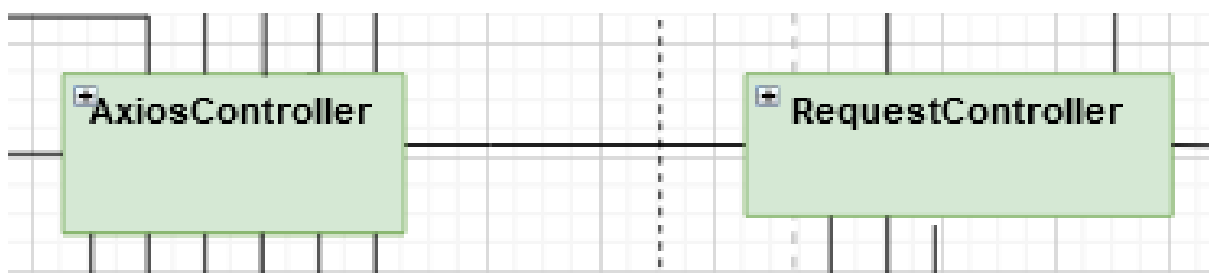
Para estruturar o sistema é usado o padrão MVC (*Model-View-Controller*), conforme apresentado na Figura 6. As classes de *Model* são responsáveis por tratar as regras de negócio e comportamento dos dados e foram marcadas pela cor roxa no diagrama de classes. As classes de *View*, por sua vez, cuidam de questões relacionadas à visualização e interface de usuário e foram marcadas pela cor amarela. Por fim os *Controllers* cuidam da intermediação entre *Models* e *Views*, e são sinalizadas pela cor verde.



Fonte: autores

O diagrama de classes foi especificado de forma a salientar a separação entre a aplicação no lado do cliente (aplicativo executado no celular do usuário) e do lado do servidor (servidor hospedado na Nuvem). Essas duas partes se conectam através de uma API (*Application Programming Interface*) que permite o cliente realizar requisições através da classe *AxiosController* para o servidor. O servidor recebe as requisições que são tratadas pelo *RequestController* e encaminhadas para as classes responsáveis pelo serviço solicitado. A conexão entre esses controladores pode ser vista na Figura 7, onde é feito um zoom no centro do diagrama de classes.

Figura 7 - Diagrama de classes - API de conexão entre cliente e servidor



Fonte: autores.

4.2 Especificação das Classes

Esta seção apresenta as descrições funcionais das classes do sistema separadas em Cliente e Servidor. As descrições baseiam-se nos diagramas de classes detalhados no Apêndice B.

4.2.1 Cliente

O lado esquerdo do diagrama de classes faz referência à parte do cliente da aplicação, isto é, a parte *mobile* do projeto. Não se trata apenas de partes meramente visuais, mas também incorpora o funcionamento e lógica da aplicação *mobile*.

Partindo da Figura B.2 pode-se ver o *AxiosController*, uma classe responsável por tratar das requisições da *API*. Ele implementa métodos que permitem realizar

chamadas *http* através da ferramenta *Axios*. Além disso, também é esta classe a responsável por implementar o tratamento de inclusão e *refresh* do *token* de autenticação em todas as requisições realizadas. Por fim, ele também inclui todo o tratamento de erro nas chamadas *http*.

O componente *OnboardingScreen* é responsável por apresentar o primeiro contato de um usuário novo com o aplicativo. O componente engloba algumas telas com mensagens de boas-vindas que são apresentadas ao usuário, que pode avançar uma a uma, ou pular diretamente para a tela de cadastro. Sendo assim, o controlador desse componente é responsável por controlar a passagem entre *slides*, além dos botões para navegação com outras telas, neste caso, telas de *login*, cadastro e o menu principal.

O *AccountScreen* é o componente que contém as telas de *login* e cadastro. Além dos botões para navegar entre as duas telas, há também botões de confirmação de entrada e entradas de texto para inserir os dados do usuário. O controlador deste componente tem importância particular de iniciar o contato com o servidor, dando início aos procedimentos de autenticação do usuário. Feita a autenticação, o controlador pode permitir o acesso do usuário apenas quando forem inseridas informações corretas.

PreferencesScreen é o componente onde ficam presentes as escolhas iniciais de gêneros e livros favoritos do usuário que está se cadastrando. Essas escolhas, caso o usuário decida fazê-las, alimentará o sistema de recomendações. O componente, portanto, armazena três telas, uma com a lista de gêneros literários, outra com uma pequena lista de livros, e finalmente uma tela de confirmação. O controlador neste caso precisa apenas receber as escolhas feitas pelo usuário e repassar as informações para o servidor.

BookInfoScreen é o componente que gerencia a tela de exibição de detalhes de um livro selecionado pelo usuário. É composto por uma mesma tela que contém três *submenus* distintos: *Sobre*, *Doadores* e *À Venda*. Há também um modal responsável

por identificar a escolha do usuário de marcar o livro como *Quero ler* ou *Já li*. O controlador tem como função receber as informações do livro vindas do banco de dados, bem como informações dos possíveis doadores do livro e lojas parceiras.

Seguindo para a Figura B.3, o componente *TransactionScreen* é responsável por exibir a interface de realização de pedidos de doação ou troca de um livro sendo oferecido. Composto por três telas e dois modais de edição este componente é um formulário que deve ser preenchido para realizar o pedido de doação. Os modais permitem a adição de uma mensagem a ser enviada junto com a solicitação, e a escolha de um livro a ser oferecido em troca. O controlador é responsável por pedir ao servidor que envie os dados da oferta de doação sendo usados, os livros que o usuário possui para oferecer em troca, além de iniciar o processo de envio da solicitação para o servidor, a ser encaminhada para o usuário que está doando o livro em questão.

O *NotificationsScreen* é o componente onde são armazenadas duas telas, uma que lista todas as notificações do usuário, e outra com a lista de conversas com outros usuários. O controlador é responsável por receber notificações e suas propriedades vindas do servidor, bem como as diferentes conversas entre usuários.

O componente *SolicitationsScreen* é quem gerencia a interação do usuário com cada notificação. Nele é possível ver os detalhes da notificação, aceitar e negar ofertas, e responder um formulário sobre como foi a transação. Também são exibidas telas de confirmação e conclusão de operação. O controlador, portanto, é responsável por gerenciar essas operações envolvendo ofertas de doação e trocas entre usuários, além da avaliação da transação.

O *UserProfileScreen* é um componente de tela única que mostra a tela de perfil de um usuário qualquer selecionado. Vale ressaltar que este componente contempla apenas usuários diferentes do perfil próprio, que é contemplado pelo *SelfUserProfileScreen*, visto na Figura B.5.

Na sequência, na Figura B.4, o componente *HomeScreen* é responsável por exibir o menu principal da interface do cliente. Neste componente são agrupadas duas telas, uma com o menu principal efetivamente, onde são apresentados livros recomendados, populares e recentes, e uma barra de buscas, além de outra tela contendo opções de filtro para busca. O controlador é responsável por dar início ao processo de busca, além de iniciar pedidos ao servidor para receber informações para preencher os carrosséis dos livros exibidos na tela.

O *ResearchResultScreen* é o componente onde são exibidos os resultados das buscas realizadas e os filtros aplicados. Vale ressaltar que além da tela de resultados, há novamente a tela para edição de filtros de busca, a mesma contida no componente anterior *HomeScreen*. Essa repetição foi feita por motivos de performance, de forma que a aplicação de filtros seja fluida em quaisquer uma das telas da qual parte, seja do menu principal, seja da tela com os resultados de uma busca previamente realizada. O controlador é responsável por receber as respostas vindas do servidor das buscas realizadas, bem como iniciar processos adicionais de buscas subsequentes.

O componente *FeedScreen* é responsável por exibir ao usuário as resenhas publicadas mais recentes dos usuários seguidos. O controlador recebe do banco de dados informações sobre as resenhas que devem ser exibidas, justamente ao número de pessoas que marcaram a resenha como “Gostei”. O controlador deve também enviar ao servidor uma possível nova marcação de “Gostei” vinda do usuário.

O componente *TransactionCreationScreen* diz respeito à interface pela qual o usuário passa para adicionar uma oferta de livro à plataforma. Composto por quatro telas e dois modais, este componente possui um controlador responsável por realizar a leitura do código de barras do livro, e enviar os dados adequados ao banco de dados para identificar o livro, além de posteriormente publicar a oferta de doação ou troca.

Por fim, na Figura B.5, o *ChatScreen* é o componente responsável pela tela de mensagens em si e onde a conversa efetivamente acontece. O controlador precisa gerenciar o recebimento e envio de mensagens, fazendo a conexão com o servidor.

O componente *ProfileEditScreen* abriga a tela de edição de perfil do usuário. Seu controlador é responsável por enviar as alterações realizadas para o banco de dados.

Já o componente *ConfigScreen* aloja as telas de configurações e “Sobre” do cliente da aplicação. Seu controlador deve aplicar as devidas configurações personalizadas na interface do usuário, de acordo com o selecionado.

Finalmente, o componente *ReviewScreen* é o responsável por receber a resenha escrita pelo usuário e a nota que deu ao livro. Essas informações são enviadas ao servidor e vinculadas tanto ao livro quanto ao perfil do autor da resenha.

4.2.2 Servidor

Com relação à especificação das classes do servidor, também se faz referência ao diagrama de classes detalhado, encontrado no Apêndice B.

Começando com a Figura B.6, nela tem-se representado o *RequestController*, classe responsável pela API que recebe as requisições do cliente e repassa para o *endpoint* específico. Depois temos a *ResearchController* que trata dos mecanismos de busca tanto de livros quanto de usuários. Há também o *AuthenticationController*, que cuida do login, gerencia os tokens dos usuários e faz esse tratamento de sessões autenticadas.

Na sequência, o *ProfileController* cuida das informações do perfil do usuário. Associado a ele há o *UserModel* que é a classe que modela a tabela do banco de dados com as informações do usuário. Associados têm-se também o *model UserEvaluation* que registra dados referentes ao feedback de outras pessoas em

relação ao usuário e também o *model Address* que registra dados sobre o endereço do usuário.

O *BookCatalogController* é a classe que cuida de fornecer os dados dos livros que estão no acervo, seja através do *BookCatalogApiController* que faz a requisição dos dados na API da *OpenLibrary* ou do banco de dados caso já esteja lá, e caso os dados não estejam no acervo, é possível adicionar um novo através do *BookCatalogSubmissionController*. Para finalizar, o *BookReviewController* é usado para recuperar as resenhas dos livros. Vale lembrar que todos esses *controllers* tem um *model* associado, como pode ser visto na imagem.

Partindo para a Figura B.7, grande parte das classes representadas nela já foram comentadas anteriormente, as que ainda não foram são: *RecommendationController*, classe que vai lidar com a atualização dos livros que estão na listagem dos populares ou novos, além de registrar informações sobre os livros que o usuário já leu ou quer ler e por fim gerar uma lista personalizada de livros recomendados. Para tal classe, é necessário o *model UserBooksLists* que também se relaciona com *BooksLists*, uma classe intermediária que agrupa os livros que estão no catálogo.

Por fim, na Figura B.8 tem-se a *TransactionsController*, responsável por criar e gerenciar uma transação de livro, troca ou doação, associada a uma unidade de livro, que representa o estado daquele livro em específico. Há também a *SolicitationsController* que cuida de implementar a lógica de negócio das solicitações de transações, ou seja, a resposta/proposta a uma transação. Vale lembrar que ambos os *controllers* possuem um *model* para auxiliar a implementação.

4.3 Estrutura de Segurança

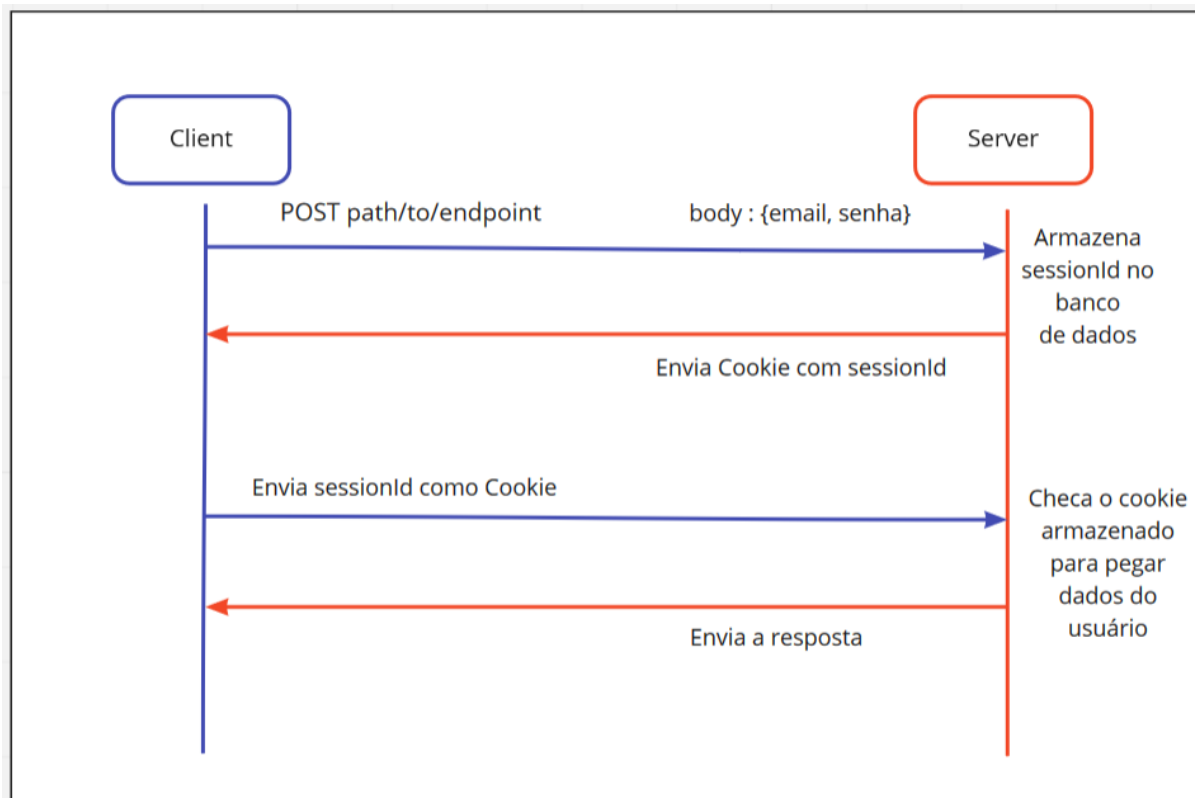
Ainda que a aplicação *mobile* permita acesso a certas funções sem que o usuário tenha realizado *login*, a maior parte do projeto só faz sentido para usuários cadastrados e autenticados. Desta forma, é necessário possuir uma forma de

garantir que este usuário possa criar uma conta e autenticá-la de forma segura.

Atualmente existe uma imensa quantidade de alternativas para realizar este processo. Dentre estas alternativas, foram consideradas 3 possibilidades: autenticação por sessões, por *token* e por JWT (*JSON Web Token*).

A autenticação por meio de sessões é uma das formas mais tradicionais na *Web* e consiste em armazenar um identificador de sessão assim que o usuário realiza o *login* na plataforma. A partir deste momento, o usuário deverá enviar o identificador desta sessão através dos *Cookies* em todas as requisições. O servidor, por sua vez, receberá esse identificador e realizará a verificação da sessão no banco de dados. Se tudo estiver correto, o servidor captura os dados do usuário através deste *Id*. A Figura 8 representa o fluxo deste modelo de autenticação.

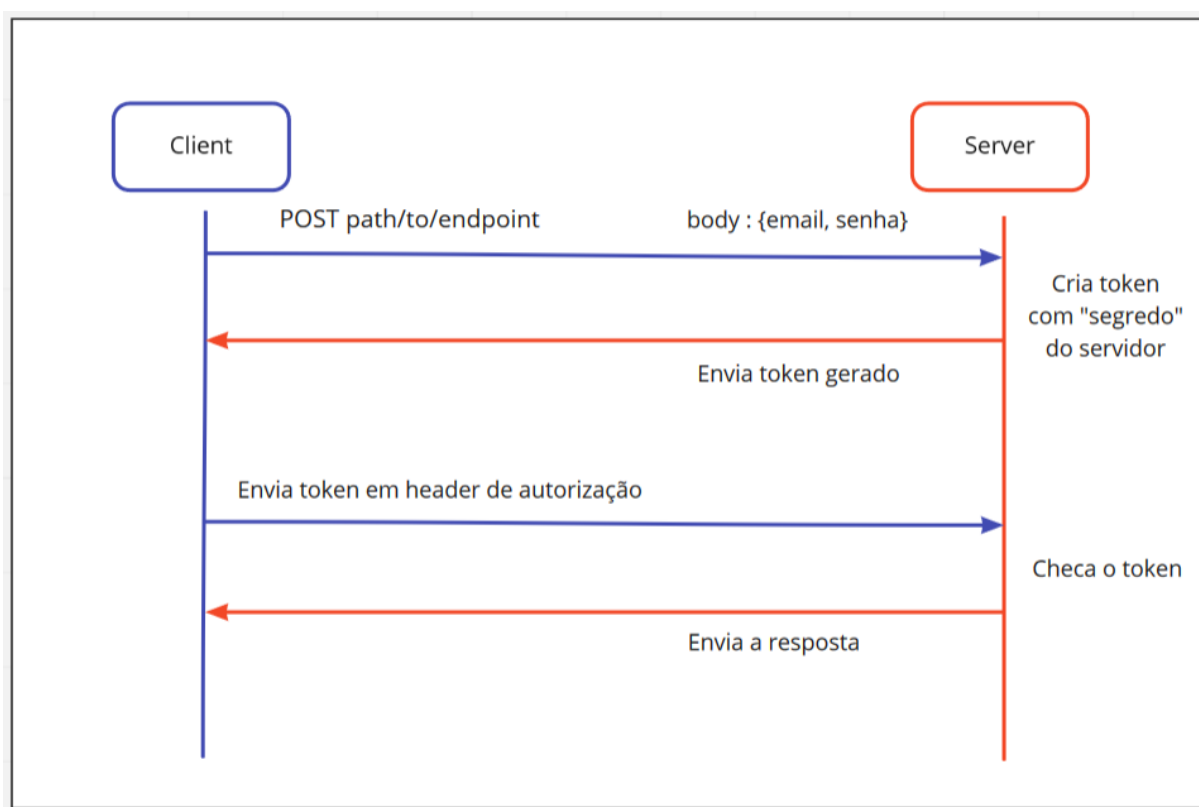
Figura 8 - Autenticação por *Cookie*



Fonte: autores.

Uma alternativa ao modelo de sessões é a utilização de *tokens*. Em vez de criar identificadores para sessão, é gerado um *token* com uma chave secreta conhecida apenas pelo servidor (sendo possível utilizar modelos mais complexos, por exemplo com a utilização de chaves pública e privada). Este *token* geralmente será armazenado no banco de dados e, em toda requisição, o usuário incluirá o *token* no cabeçalho de autenticação. O servidor, por sua vez, irá comparar o token recebido com o *token* armazenado no banco de dados para autorizar a requisição. A Figura 9 mostra um exemplo desse funcionamento.

Figura 9 - Autenticação por *Token*



Fonte: autores.

Por fim, tem-se o método JWT (JSON Web Token). Este método é um tipo de autenticação de *token* com uma diferença na forma de tratar a validação do *token* pelo lado do servidor. O *token* possui um formato específico e é assinado a partir de um algoritmo HMAC (Hash-based Message Authentication Code) como o HSA256 e

da chave secreta. Esse *token* pode ser decodificado e sua assinatura só pode ser validada por aquele que conhecer a chave secreta (ou seja, apenas o servidor). Desta forma, é possível realizar a validação do *token* enviado pelo usuário sem a necessidade de armazená-lo no banco de dados.

O modelo de autenticação utilizado neste projeto é o JWT pela facilidade de tratamento em ambientes *Mobile* e pela possibilidade de utilização sem precisar armazenar os *tokens* no banco de dados.

5. IMPLEMENTAÇÃO

Nesta seção é discutida a implementação do código, sua estrutura e organização, de forma que seja possível entender a forma na qual a aplicação foi desenvolvida na prática. Na seção 5.1 é possível ler como a interface mobile e o servidor foram organizados, respectivamente nas seções 5.1.1 e 5.1.2. Já na seção 5.2 pode-se conferir o link para o *download* do cliente da aplicação para instalação local.

5.1 Organização do código

5.1.1 Mobile

A interface *mobile* da aplicação foi estruturada de tal forma que o programa pudesse ser organizado de forma clara e modularizada, evitando repetição desnecessária de código e reaproveitando trechos sempre que possível.

A estrutura do programa em *React Native* é baseada em componentes. Isto é, a maior parte dos elementos presentes na interface são “peças” que podem receber atributos e serem chamados repetidas vezes conforme a necessidade. O funcionamento é similar ao de classes e objetos, muito embora essa nomenclatura não seja utilizada. Botões, estilos de texto, estruturas mais complexas e até mesmo telas inteiras são considerados componentes.

Para manter a organização do programa, uma estrutura de arquivos foi estabelecida desde o início da codificação e pode ser vista a seguir:

```
src/  
  Assets/  
  modules/  
  components/  
    atom/  
    mol/  
    org/  
  services/  
  store/  
  actions/
```

```
constants/  
reducers/  
store.js  
utils/  
consts.js  
App.js
```

Há outros arquivos e pastas para funcionamento interno do *framework* e alguns arquivos de configuração, mas a principal é a pasta *src*. Nela fica concentrada a maior parte dos arquivos responsáveis pelas interfaces e lógica de funcionamento.

Seu primeiro integrante é a pasta *Assets*, onde ficam armazenados arquivos como imagens e fontes de texto personalizadas a serem importadas nas diversas telas da aplicação.

Há também a pasta *modules*, responsável por armazenar os arquivos de cada tela a ser exibida no cliente. Esses componentes muitas vezes não representam cada um apenas uma tela, mas sim um pequeno conjunto delas, quando por exemplo for comum sua visualização em rápida sucessão, com o objetivo de melhorar a organização de navegação entre telas.

A pasta *components* guarda os arquivos de todos os outros componentes utilizados para construir as telas e é dividida em três outras pastas: *atom*, *mol* e *org*. Essa divisão ocorre de acordo com o grau de complexidade do componente, onde *atom* representa componentes mais simples, como por exemplo botões, enquanto *org* representa componentes mais complexos, como conjuntos de *atoms* e *mols*. A pasta *mol* é para componentes intermediários, nem tão simples, nem tão complexos.

A pasta *services* é a responsável por serviços de acesso à APIs e armazenamento, isto é, onde é feito de fato a conexão com o *back-end* da aplicação.

Em *store*, por sua vez, são armazenados os arquivos relacionados ao gerenciamento de estados globais das telas da interface mobile. Da maneira como o

framework funciona, estados não são salvos ou passados de um componente para o outro, sendo de responsabilidade desses arquivos, gerenciados pela biblioteca importada *Redux*, salvar e informar estados globais do cliente. Isso é feito utilizando as pastas *actions*, *constants*, *reducers* e o arquivo *store*. *Actions* definem quais mudanças serão feitas nos estados globais. *Reducers* recebem essas definições e aplicam as mudanças de fato. O arquivo *store* guarda os estados, e *constants* são constantes de funcionamento interno para referenciar estados.

A pasta *utils* contém utilitários gerais que não se encaixam em nenhuma das outras divisões, mas de uso geral em várias partes do código. Já o arquivo *consts.js* registra constantes gerais para uso de todo o código, como por exemplo, dimensões da tela do dispositivo e cores padronizadas do layout, de forma que sua definição fique centralizada.

5.1.2 Servidor

A estrutura do servidor foi planejada seguindo os modelos de controladores e *models*, a fim de organizar as interfaces de requisições do mobile e requisições para API externas.

Com relação ao código em si, ele foi desenvolvido em *Django Python*, seguindo o *pattern* de MVC (*model-view-controller*), como já explicado anteriormente.

Com relação a estrutura de arquivos, tem-se:

```
src/  
  backend-env/  
  core/  
    domain/  
    exceptions/  
    migrations/  
    tests/  
    views/  
  admin.py  
  apps.py
```

```
models.py
serializers.py
urls.py
Desapega/
  asgi.py
  settings.py
  urls.py
  wsgi.py
Docs/
manage.py
```

Na pasta `core/domain/` são localizados os controladores da lógica de negócio, já o tratamento de exceções e os retornos são definidos na `exceptions/`.

O diretório `migrations/` é uma interface obrigatória do framework Django e nela consiste as mudanças relacionadas ao banco de dados, além de um log de mudanças passadas. Os testes unitários estão na `test/` e as `views` definem a interface entre os controladores e as `urls`, ou `endpoints`.

O arquivo de `admin.py` define as configurações do *Django Admin* parte do framework relacionada a questões de segurança e autenticidade, além de possuir de fato uma interface de super usuário para visualização dos dados.

Já o `models.py`, como o nome sugere, atua como ORM, mapeamento objeto-relacional, para as tabelas do banco de dados, em conjunto temos os `endpoints` definidos na `urls.py`.

Dentro da pasta `Desapega/` é onde está definido todas as questões relacionadas a inicialização do projeto *django*, configurações relacionadas ao projeto em si e também ao ambiente de hospedagem.

Sendo `settings.py` o arquivo principal onde as configurações são definidas, diferentemente do outro `urls.py`, essa atua somente para definir a página do *Django Admin*.

5.2 Repositórios

Por uma questão de segurança na eventual publicação futura do aplicativo, os repositórios não são divulgados de forma pública. Contudo, pode ser realizado o *download* do cliente da aplicação no link https://drive.google.com/drive/folders/1m91blvy3fAgC7WFim-q_cEMQkT3b2JFO?usp=sharing.

As instruções para a instalação são descritas na seção 7.1.

6. TESTES

Nesta seção é descrita a execução dos testes de validação, bem como seus resultados, de acordo com o Plano de Testes de Validação do sistema. É feita uma análise dos testes bem e mal sucedidos, além de discussões sobre o que foi feito nos casos incorretos e quais foram as lições aprendidas durante o processo, seja em relação aos testes em si, seja no que diz respeito ao projeto como um todo.

6.1 Relatório de Testes de Validação

Como preparação para a realização dos testes, a aplicação *mobile* foi carregada em um *smartphone* de sistema operacional *Android* (versão 11) e modelo *Samsung Galaxy A70* (SM-A705MN) e em um simulador, baseado no *software Android Studio*, do dispositivo *Nexus 5* também utilizando o sistema operacional *Android* (versão 11). A preferência pela utilização, durante os testes, foi com o dispositivo físico, com o uso do simulador para testes envolvendo mais de um usuário. Além disso, foi assegurada uma conexão estável com a Internet e o pleno funcionamento do servidor antes do início do processo.

A execução dos testes ocorreu seguindo os passos estabelecidos na Tabela 3 e comparando os resultados esperados com os obtidos. Os resultados então foram anotados na Tabela 4, indicando o sucesso ou não de cada teste e eventuais observações adicionais observadas.

É possível ver na Tabela 4 os resultados obtidos no processo de validação especificado na seção 3.3. A tabela é organizada separando os testes para cada caso de uso, mostrando quantos foram bem sucedidos.

Tabela 4 - Resultados dos testes de validação

Caso de Uso	Teste		Observações
1	1	Ok	Mudança de layout: usuário sempre é redirecionado para a tela de login

			primeiro
	1a	Ok	Mudança de layout decorrente da anterior: usuário é redirecionado diretamente para a tela de login
	1b	X	Não é mais possível usar o aplicativo sem cadastro
2	2	Ok	Mudança de design: inclui campo de telefone para contato. Foto e região de moradias redirecionados para edição de perfil e não são obrigatórios. Lista de livros exibida não é relacionada aos gêneros selecionados, mas sim usa os livros mais populares do momento. Ao final, usuário é redirecionado para cadastro/login, onde deve inserir suas credenciais para entrar.
	2a	X	Cadastro por meio de redes sociais não foi implementado.
	2a1	X	Cadastro por meio de redes sociais não foi implementado.
	2b	Ok	
	2c	Ok	Testes 2c e 2d foram unidos, isto é, ao pular em qualquer uma das telas, o usuário é redirecionado diretamente para a tela de confirmação.
	2d	Ok	
	2e	X	Ainda não há termos de uso disponíveis.
	2f	Ok	Mensagem de erro não é específica para indicar que o email já foi cadastrado, mas uma mensagem informando de um erro é exibida.
	2g	X	Região de moradia não é mais obrigatória, e foi deslocada para edição de perfil ao invés do cadastro.
3	3	Ok	

	3a	X	Login por meio de redes sociais não foi implementado
	3b	Ok	
4	4	X	O sistema de recomendações personalizadas foi retirado do escopo do projeto.
5	5-livro	Ok	
	5-autor	Ok	
	5a	Ok	
	5b	Ok	
	5c	X	Filtros de gêneros literários foi retirado temporariamente do escopo por conta da falta de suporte do banco de dados usado como base em primeiro momento.
6	6	Ok	
	6a	Ok	Fluxo de criação de resenha foi alterado, agora o botão fica na própria página de informações do livro.
	6b	X	Não é mais possível usar o aplicativo sem cadastro
	6c	X	Usuário não pode editar sua resenha, mas pode publicar outras.
	6c1	X	Usuário não pode apagar sua resenha.
7	7-Quero ler	Ok	
	7-Já li	Ok	
	7a	Ok	Pequena mudança de layout: basta tocar fora do modal para cancelar, botão próprio para isso era desnecessário.
	7b	Ok	

8	8	Ok	
9	9	Ok	Pequenas mudanças de layout das telas.
	9a	X	Filtro de doadores foi descartado.
	9b	Ok	
10	10	X	Links patrocinados removidos do escopo do projeto.
11	11	Ok	
	11a	Ok	
	11b	Ok	Foi acrescentado um fluxo ao escopo do projeto, onde o usuário pode preencher um formulário interno ao aplicativo onde preenche o ISBN, título, autor(a), categoria, foto da capa e sinopse. Dessa forma o banco de dados crescerá gradativamente com o uso. O administrador ainda deve aprovar a inclusão antes de passar a valer.
12	12	Ok	Com a remoção do sistema de chat do escopo do projeto, o telefone do usuário é exibido e uma mensagem solicita que entrem em contato para combinar os detalhes da transação.
	12a	Ok	Os testes 12a e 12b foram unidos, visto que não é possível avaliar transações que foram rejeitadas. A avaliação foi alterada para quando uma transação é finalizada.
	12b	Ok	
13	13	X	Sistema de chat removido do escopo do projeto.
	13-Recebimento	X	

	13a	X	
14	14	Parcial	Depoimentos são exibidos apenas no perfil próprio do usuário como feedback pessoal.
	14a	X	Agora é obrigatório que a transação seja avaliada.
15	15	Ok	
16	16	Ok	A idade do usuário foi descartada. Pequena alteração de <i>layout</i> : botão de edição de perfil agora fica no meio da tela de perfil.
	16a	X	Não há configurações a serem alteradas no app.
	16b	Parcial	Para fazer logout do aplicativo basta fechá-lo, tornando o botão desnecessário.
	16c	X	Idem teste 16b
	16d	X	Não há uma seção de "Sobre" para ser exibida no protótipo.
17	17	X	Sistema de denúncia removido do escopo do protótipo.
	17a	X	
18	18	X	Sistema de denúncia removido do escopo do projeto.
	18a	X	Não é possível banir o usuário por um tempo, apenas excluí-lo. Contudo, há opções adicionais de administração, como a de aprovação de livros cadastrados novos.

Fonte: autores.

Como pode ser visto, obteve-se um bom grau de sucesso ao longo dos testes executados. Há casos em que não houve sucesso por mudanças no escopo da aplicação e também por pequenas alterações de *layout* que ocorreram entre o período de especificação do Plano de Testes de Validação e da conclusão do projeto.

6.2 Discussões sobre o Processo de Validação

Nesta seção são discutidos os resultados finais e conclusões obtidas por meio dos resultados dos testes de validação mostrados na seção 6.1.

Um primeiro ponto que vale ser destacado é que os testes de validação foram elencados em um ponto do desenvolvimento do projeto onde a especificação estava sendo concluída e a implementação ainda não havia sido iniciada. Isso fez com que após o processo de elaboração dos testes ainda houvesse diversas mudanças no escopo do projeto, seja alterando ou incluindo partes específicas. Durante o desenvolvimento e após a especificação inicial aconteceram mudanças na aplicação por meio de processos iterativos com o cliente, o que inclui alterações no escopo do projeto em funcionalidade e layout.

Contudo, foi importante para o grupo manter a consistência do que foi escrito anteriormente e não apagar mudanças. Readequar os testes de acordo com os resultados faria com que o todo o processo perdesse o propósito.

Diversos aprendizados decorreram do processo de testes como um todo. Em primeiro lugar, chegou-se à conclusão que definir aspectos visuais e de *layout* na especificação de testes não tem grande utilidade, salvo quando a interface é relevante para a funcionalidade exigida. Não é frutífero para o projeto tentar especificar, por exemplo, o posicionamento de botões na tela em um momento do desenvolvimento em que a definição de funcionalidades é mais relevante que a de visuais, principalmente quando tais visuais ainda não estão bem sedimentados e poderão sofrer alterações.

Em segundo lugar, elencar os testes de validação serve não apenas como base para testes futuros ao final do projeto, mas também para um melhor entendimento e solidificação de ideias em seu início. Isso facilita a compreensão de fluxos mais complexos da aplicação e deixa o desenvolvimento mais fluido.

Os testes de validação também ajudaram na identificação de pequenos erros dos quais não se tinha conhecimento, mas que foram prontamente corrigidos. Um exemplo foi um *bug* que fazia com que todas as fotos de usuário fossem iguais por uma inconsistência na forma como o nome das imagens era gerado e enviado para o servidor. O problema foi corrigido e o teste refeito com sucesso.

Por fim, ressalvados esses pontos, os resultados obtidos foram muito satisfatórios e atendem à Especificação de Requisitos Funcionais e Não-funcionais definida no início do projeto junto ao cliente.

7. INSTALAÇÃO

O objetivo desta seção é descrever um procedimento de como instalar e executar a aplicação desenvolvida na perspectiva de usuário, tanto para o cliente *mobile* quanto para o servidor. O público-alvo engloba possíveis interessados em testar a aplicação para fins de contribuição ou avaliação deste projeto, bem como usuários em geral e integrantes do Enactus USP e associados.

7.1 Cliente

7.1.1 Pré-requisitos do Dispositivo

É necessário um *smartphone* com sistema operacional Android de versão 11 ou superior. Também é necessária uma conexão com a Internet. Para ter acesso a todas as funções do sistema, é necessário que o *smartphone* tenha câmera integrada.

7.1.2 Instalação do Software

A instalação se dá por meio de um arquivo do tipo *.apk* que deve ser instalado no *smartphone*. O *download* do arquivo pode ser encontrado no link exibido na seção 5.2. Como cada celular possui suas peculiaridades em termos de arquitetura, o android produz arquivos *.apk* para diversas arquiteturas. Ao publicar um aplicativo na *Google Play Store*, essas questões são tratadas automaticamente. Como não é o nosso caso, será necessário detectar qual a arquitetura do seu processador. Por exemplo: se utilizamos um celular Motorola Moto G7, precisaremos baixar o arquivo *app-arm64-v8a-release.apk* (arquitetura baseada em ARM). Também é possível tentar utilizar o *app-universal-release.apk*, mas em alguns casos ele pode não funcionar.

Após o *download* do arquivo terminar, deve-se selecionar o arquivo diretamente no *smartphone* que será usado. O dispositivo poderá pedir permissão de instalação antes de continuar, visto que não é um aplicativo reconhecido pela loja oficial do

smartphone. Essa permissão deve ser aceita, caso contrário a instalação será cancelada.

Após o término da instalação o aplicativo se encontrará disponível para uso.

7.2 Servidor

7.2.1 Pré-requisitos do Servidor

O sistema do servidor precisa suportar várias requisições ao mesmo tempo pensando no futuro do projeto quando a base de usuários estiver bem alta, além de necessitar de um armazenamento da ordem de 10 *terabytes* (TB) para os dados estáticos trafegados e também para o banco de dados.

O grupo recomenda utilizar um sistema de nuvem para hospedar a aplicação, seja da Amazon, Google, Microsoft ou Heroku, pois assim pode-se escalar os requisitos à medida que a base de usuários cresça evitando custos exorbitantes logo de início.

7.2.2 Instalação do Software

A instalação do lado do servidor se dá ao configurar o banco de dados, o sistema de armazenamento de arquivos e, por fim, realizar o *deploy* da aplicação “*backend-desapega*”.

O banco de dados sugerido é o PostgreSQL, que foi escolhido por já ser integrado ao Heroku, utilizado para desenvolvimento. Na eventual tentativa de subir o projeto para outros serviços em nuvem, a alteração do banco de dados pode ser proveitosa de acordo com a compatibilidade de cada um. Contudo, outros tipos de banco de dados não foram testados neste projeto.

Para realizar o *deploy* da aplicação, deve-se usar como referência o arquivo *Procfile*, disponível na *branch* “heroku” do repositório.

8. CONSIDERAÇÕES FINAIS

Nesta seção são apresentadas as conclusões obtidas pelo desenvolvimento do projeto após sua finalização. Também são descritas contribuições do trabalho, isto é, desafios e inovações de partes relevantes do desenvolvimento. Finalmente, são listadas sugestões para trabalhos futuros que podem ajudar a guiar melhorias na aplicação por parte do Enactus USP e seus associados dado o interesse.

8.1 Conclusões

A aplicação final, em meio a alterações em seu escopo ao longo do desenvolvimento, se provou um sucesso em mais de um aspecto.

No que diz respeito à aplicação em si, se encontra funcional em seu núcleo principal. Com ela, o Enactus USP tem um protótipo com o qual pode realizar testes de usabilidade internos com o objetivo de descobrir decisões de *design* e funções que surtem os efeitos desejados ou que precisam de ajustes para atender seus objetivos finais.

Além disso, o cliente possui uma base sólida na qual se apoiar para incremento de funções adicionais e na transformação do projeto em produto, reduzindo custos e tornando o projeto economicamente vantajoso.

No entanto, o uso comercial do produto não foi escopo do projeto. Portanto, ainda é necessário que seja feita uma passagem legal da aplicação como um todo para o Enactus USP, avaliando quais cuidados devem ser tomados perante as leis vigentes antes da publicação final do produto. Por esse motivo, o grupo não recomenda e não se responsabiliza pela publicação da aplicação, na versão atual de protótipo, para o público em geral, mas incentiva o uso interno para desenvolvimento com usuários que estejam cientes e de acordo, para fins de testes e melhorias, desde que dentro das normas legais.

Do ponto de vista do projeto enquanto Trabalho de Conclusão de Curso, os objetivos do grupo foram alcançados e superados em expectativas. A maior meta neste sentido era o de aprendizado, tanto técnico quanto de gerenciamento de projeto e envolvimento com um cliente real. Todo o processo ensinou diversas lições e complementou de forma riquíssima o aprendizado da graduação, aprofundando diversos conceitos estudados durante o curso, ao mesmo tempo em que preparou o grupo para o mercado de trabalho.

No Apêndice C é possível visualizar algumas das telas da aplicação em seu estado final para este projeto, com a finalidade de exemplificação e ilustração.

8.2 Contribuições do Trabalho

Um dos maiores desafios para o frontend foi desenvolver uma interface que fosse responsiva com proporções variadas de tela. Por experiência própria, o grupo conheceu diversos aplicativos de celular com interface não adequada para dispositivos com proporções menores, causando até mesmo problemas de utilização, pois botões poderiam ficar fora da tela, por exemplo. Trazer essa compatibilidade e evitar esses problemas eram de extrema importância para o grupo, de forma que as telas sempre eram testadas em dois sistemas de proporções diferentes ao longo de todo o desenvolvimento. Uma das inovações que surgiram disso foi um algoritmo que ajusta a escala do tamanho das fontes de texto de acordo com o tamanho da tela por meio de interpolação linear, que cumpriu muito bem seu propósito.

Outro grande desafio para o *frontend* foi o tratamento adequado de requisições ao servidor por meio do Axios e do Redux. Esses dois pacotes oferecem uma estrutura robusta para gerenciamento de requisições, mas são complexas devido à sua alta versatilidade. O seu uso na aplicação permite que requisições sejam realizadas de forma controlada e segmentada, tornando o código melhor organizado. Seu uso

pode servir como fonte de estudo de ambos os pacotes e como integrá-los a um programa desenvolvido em React Native.

Além disso, um dos pontos mais relevantes foi o processo de estruturação da aplicação em si. Construir documentações que permitissem organizar desde a fundação do projeto até o seu desenvolvimento possibilitou analisar com calma os diversos aspectos positivos e negativos das soluções possíveis para cada bloco da aplicação. Tópicos como segurança, autenticação e hospedagem, por exemplo, são alguns exemplos que representaram grandes desafios técnicos e exigiram muito estudo e planejamento para definir a melhor forma de implementação.

8.3 Trabalhos Futuros

Ao longo do processo de desenvolvimento diversas ideias de melhoria surgiram e acabaram não sendo implementadas, seja porque fugiam do escopo do projeto, seja porque foram removidas durante as fases iniciais de planejamento antes do fechamento do escopo. Caso o Enactus USP deseje dar continuidade ao projeto, algumas sugestões de melhoria para a aplicação são descritas nesta seção.

Uma primeira sugestão de uma função adicional fora do escopo do projeto seria um sistema de compras interno. Nas conversas iniciais de definição de projeto, essa era uma demanda para a aplicação que já na época foi decidido que não seria incluída, visto sua grande complexidade para um funcionamento confiável e seguro. É uma *feature* interessante de ser implementada para melhoria do sistema e que poderia servir de base para um projeto inteiro à parte.

Outra melhoria possível de ser implementada seria a compatibilidade do sistema com dispositivos baseados no sistema operacional *iOS*. Muito embora o foco do projeto tenha sido fazer o desenvolvimento para sistemas *Android*, o uso do *framework* do *React Native* deve facilitar a transição e adaptação para a nova plataforma, o que incluiria uma parcela relevante de usuários.

No que diz respeito ao banco de dados dos livros, muito embora tenha sido feito um esforço para possibilitar sua expansão e melhoria, ainda não há um sistema que

permita edição dos livros já existentes no banco de dados usado como base, que eventualmente traz informações em inglês ou faltantes. Um sistema de edição desses campos incorretos, ou pelo menos de aviso à administração melhoraria consideravelmente a experiência do usuário.

Além disso, também é possível citar algumas funções que foram removidas do escopo do projeto e que estão descritas nas seções anteriores ou que foram sugeridas ao longo do desenvolvimento mas acabaram não sendo viáveis de serem acrescentadas ao escopo, mas que agregariam positivamente a usabilidade do usuário. Alguns exemplos são o acesso por pessoas não cadastradas; novas funções no *feed* de resenhas, como rastreamento de outras atividades, botões de curtir e comentar; inclusão de geolocalização do usuário para listar ofertas por proximidade; *login* e cadastro por meio de redes sociais; e a implementação de um sistema de *chat* interno à aplicação.

Um sistema de recomendação robusto também poderia ser implementado, focando no uso de técnicas de inteligência artificial para melhores recomendações. Esta é outra ideia capaz de sustentar todo um projeto focado apenas nisso.

Do ponto de vista do *front-end* da aplicação, uma melhoria específica que pode ser citada é a implementação de animações para deixar a interface mais agradável e responsiva ao usuário. Além disso, é importante realizar testes de uso da aplicação em mais aparelhos de forma a assegurar compatibilidade com ainda mais tamanhos diferentes de tela.

Do ponto de vista do servidor, a primeira e grande melhoria que tornaria o projeto mais completo seria uma interface dedicada para uso interno do administrador de sistema, juntamente a um sistema de denúncias, remoção de fotos e banimento de usuários. Essa interface desde o início não era parte principal do escopo do projeto, que é focado na aplicação *mobile*. Contudo é uma adição grata ao projeto. Muito embora haja um sistema administrativo, ele não é tão robusto justamente por não ser foco do escopo do projeto.

Além disso, é possível adicionar um sistema robusto de *logs* no sistema e fazer a persistência dele no banco de dados, a fim de facilitar a identificação de erros e facilitar a manutenção futura.

REFERÊNCIAS

DJANGO. The web framework for perfectionists with deadlines, 2005. Disponível em <<https://www.djangoproject.com/>>. Acesso em: 27 de jul. de 2021.

EXPRESS. **NODE.JS web application framework**, 2017. Disponível em <<https://expressjs.com/>>. Acesso em: 27 de jul. de 2021.

FLASK. **Flask documentation (2.0.x)**, 2010. Disponível em <<https://flask.palletsprojects.com/en/2.0.x/>>. Acesso em: 27 de jul. de 2021.

FLUTTER. **Beautiful native apps in record time**, 2021. Disponível em <<https://flutter.dev/>>. Acesso em: 27 de jul. de 2021.

INSTITUTO PRÓ-LIVRO. **Retratos da leitura no Brasil**, 5ª edição, 2020. Disponível em:
<https://prolivro.org.br/wp-content/uploads/2020/09/5a_edicao_Retratos_da_Leitura_no_Brasil_IPL-compactado.pdf>. Acesso em: 11 de nov. de 2021.

REACT NATIVE. Learn once, write anywhere, 2021. Disponível em <<https://reactnative.dev/>>. Acesso em: 27 de jul. de 2021.

APÊNDICE A - Tabela de relação entre casos de uso e requisitos funcionais

Pode-se ver na Tabela A uma relação entre os casos de uso e os requisitos funcionais definidos, seguindo o que foi especificado na Seção 2.

Tabela A - Relação entre casos de uso e requisitos funcionais

Caso de Uso	Requisitos funcionais relacionados
Caso de Uso 1: Acesso ao aplicativo	3, 4
Caso de Uso 2: Cadastrar usuários	1
Caso de Uso 3: Permitir acesso de usuário	1, 2
Caso de Uso 4: Visualizar recomendações personalizadas de livros	3, 9, 10
Caso de Uso 5: Sistema de busca	3, 8, 23, 24
Caso de Uso 6: Ler e escrever resenhas	2 (para escrever), 3, 5, 22, 26
Caso de Uso 7: Marcar/Desmarcar livros que tem interesse ou já leu	2, 6, 7
Caso de Uso 8: Visualizar feed de atividade de usuários seguidos	2, 24
Caso de Uso 9: Solicitar livros	2, 11, 14, 26
Caso de Uso 10: Acessar link patrocinado	3, 15
Caso de Uso 11: Adicionar livros na plataforma	2, 5, 11, 12, 13
Caso de Uso 12: Responder solicitação	2, 16, 20, 21
Caso de Uso 13: Enviar ou ler mensagens	2, 18, 19, 20, 26
Caso de Uso 14: Avaliar doador após transação	2, 16
Caso de Uso 15: Visualizar perfil	2, 17, 23, 26
Caso de Uso 16: Editar perfil próprio	2, 17
Caso de Uso 17: Realizar denúncia	2, 26
Caso de Uso 18: Moderar banco de dados	2(superusuário), 25

Fonte: autores.

APÊNDICE B - Diagrama de classes detalhado

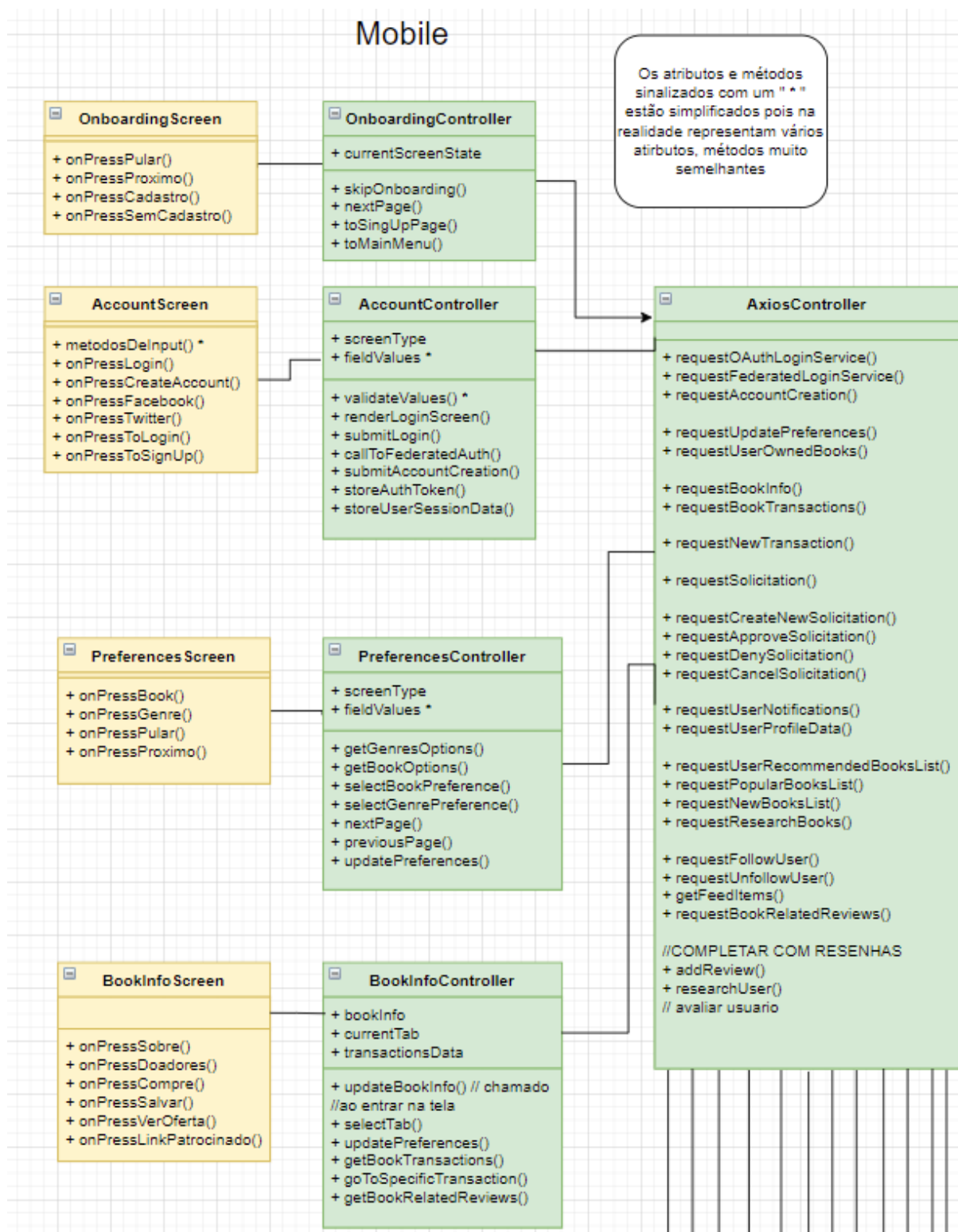
Pode-se ver na Figura B.1 a visão completa do diagrama de classes detalhado. A parte referente ao *mobile* se encontra à esquerda, e a parte referente ao servidor se encontra à direita. Por questões de legibilidade, partes do diagrama são copiadas separadamente nas Figuras B.2 a B.7.

Figura B.1 - Diagrama de classes completo detalhado



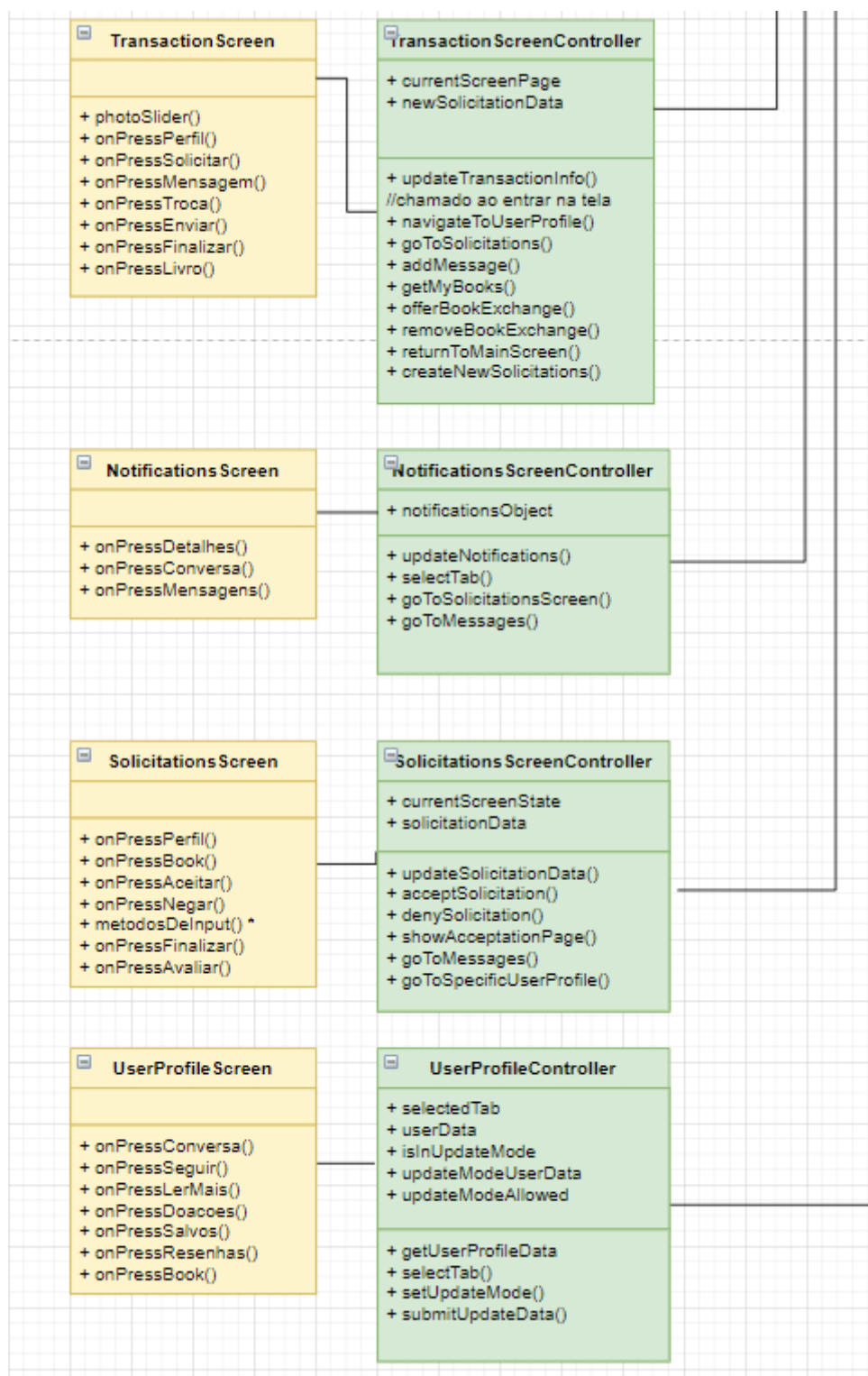
Fonte: autores.

Figura B.2 - Diagrama de classes completo detalhado - Parte 1 Cliente



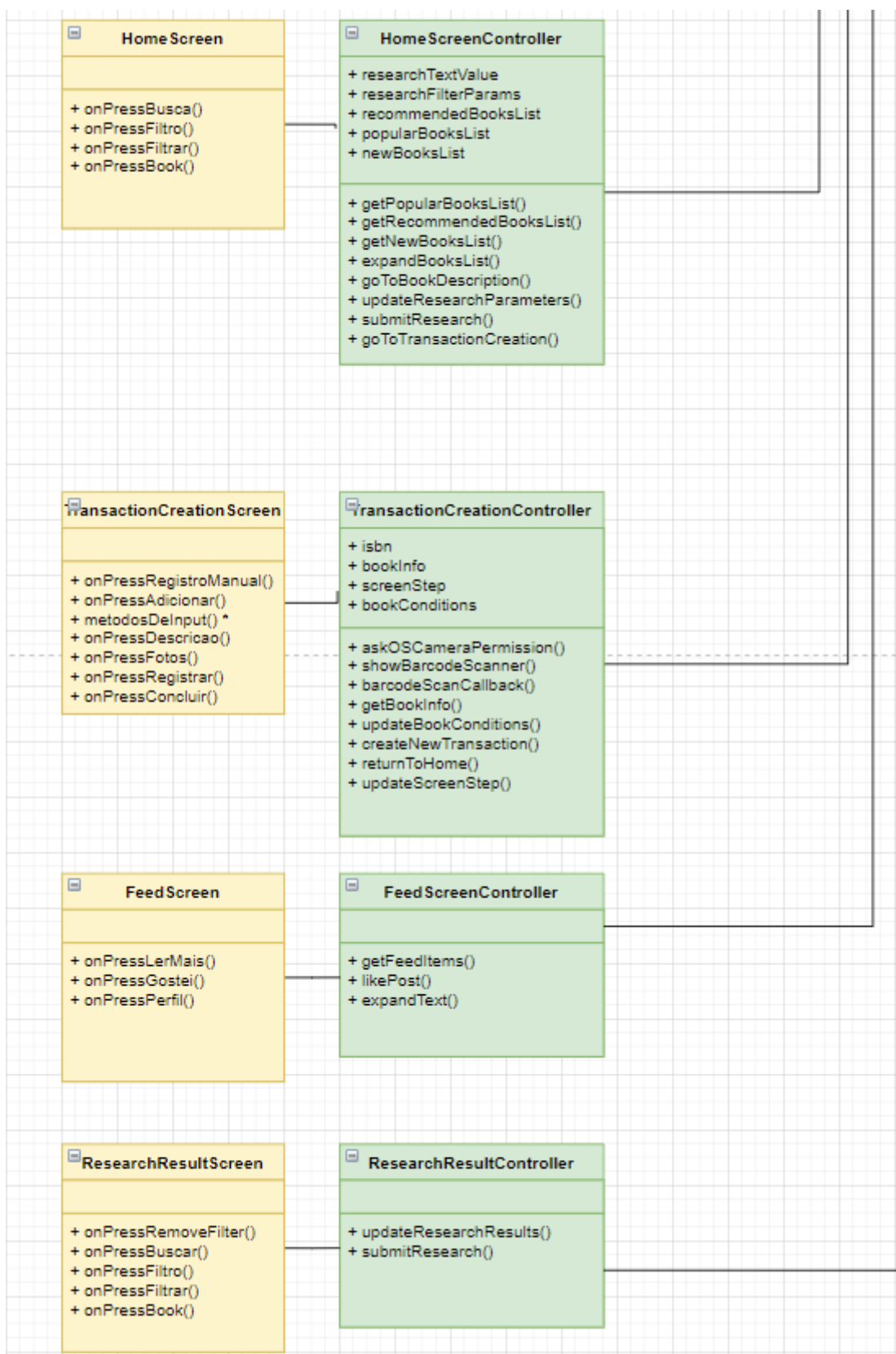
Fonte: autores.

Figura B.3 - Diagrama de classes completo detalhado - Parte 2 Cliente



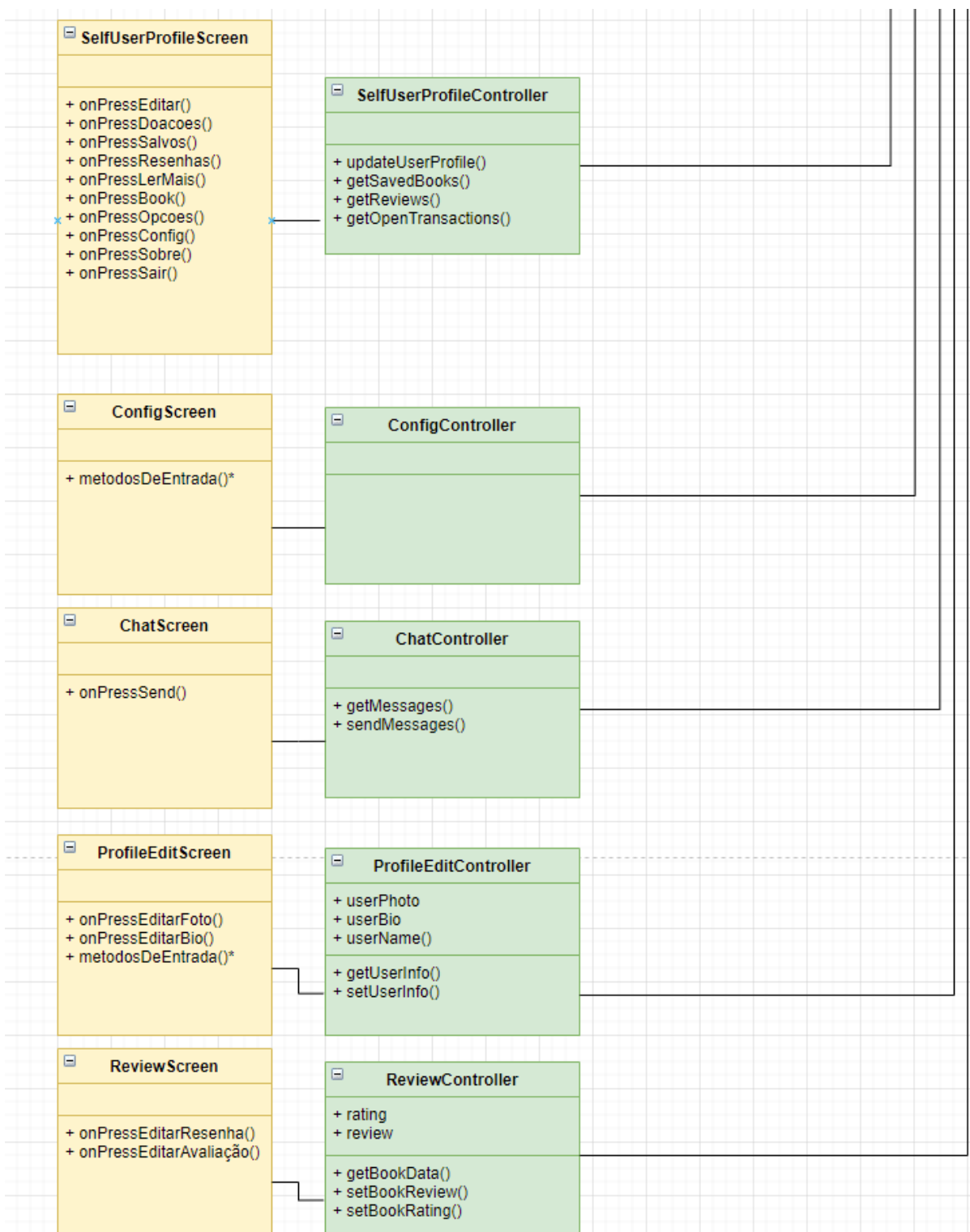
Fonte: autores.

Figura B.4 - Diagrama de classes completo detalhado - Parte 3 Cliente



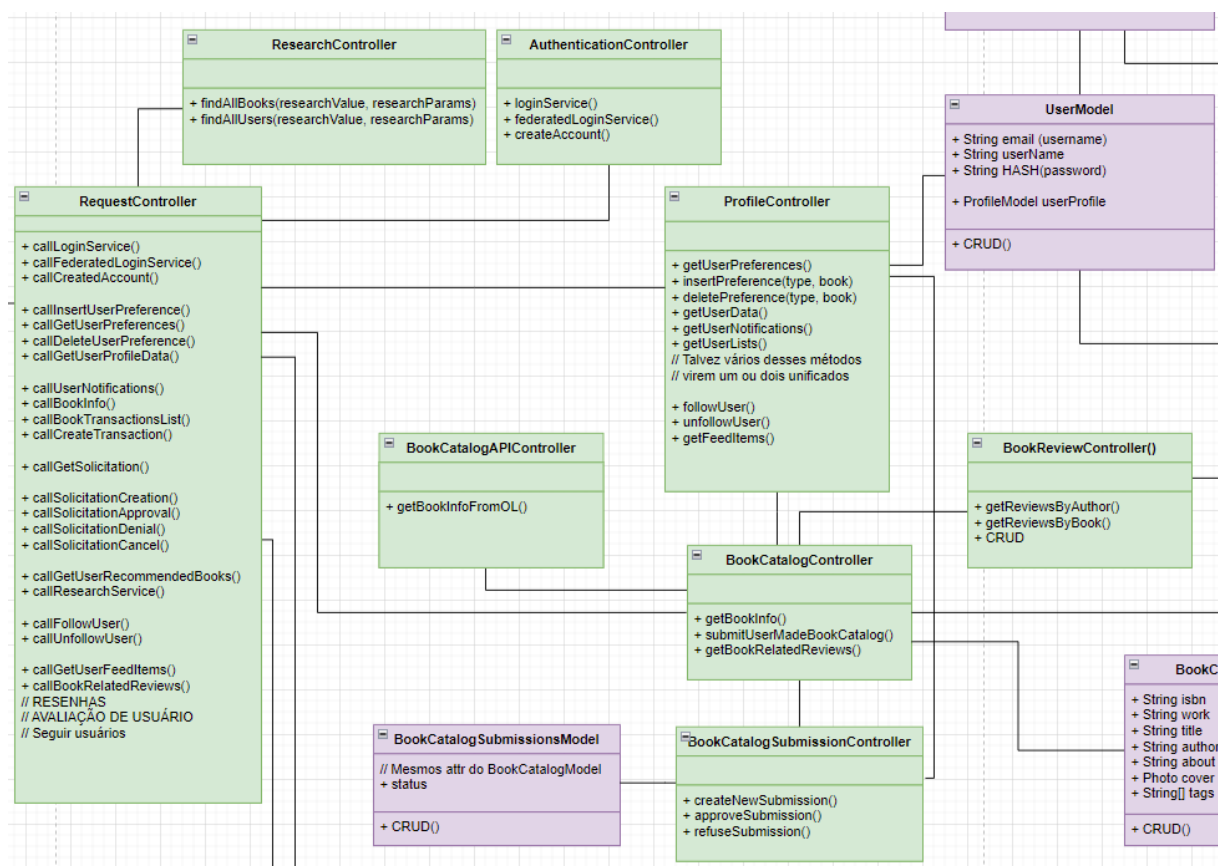
Fonte: autores.

Figura B.5 - Diagrama de classes completo detalhado - Parte 4 Cliente



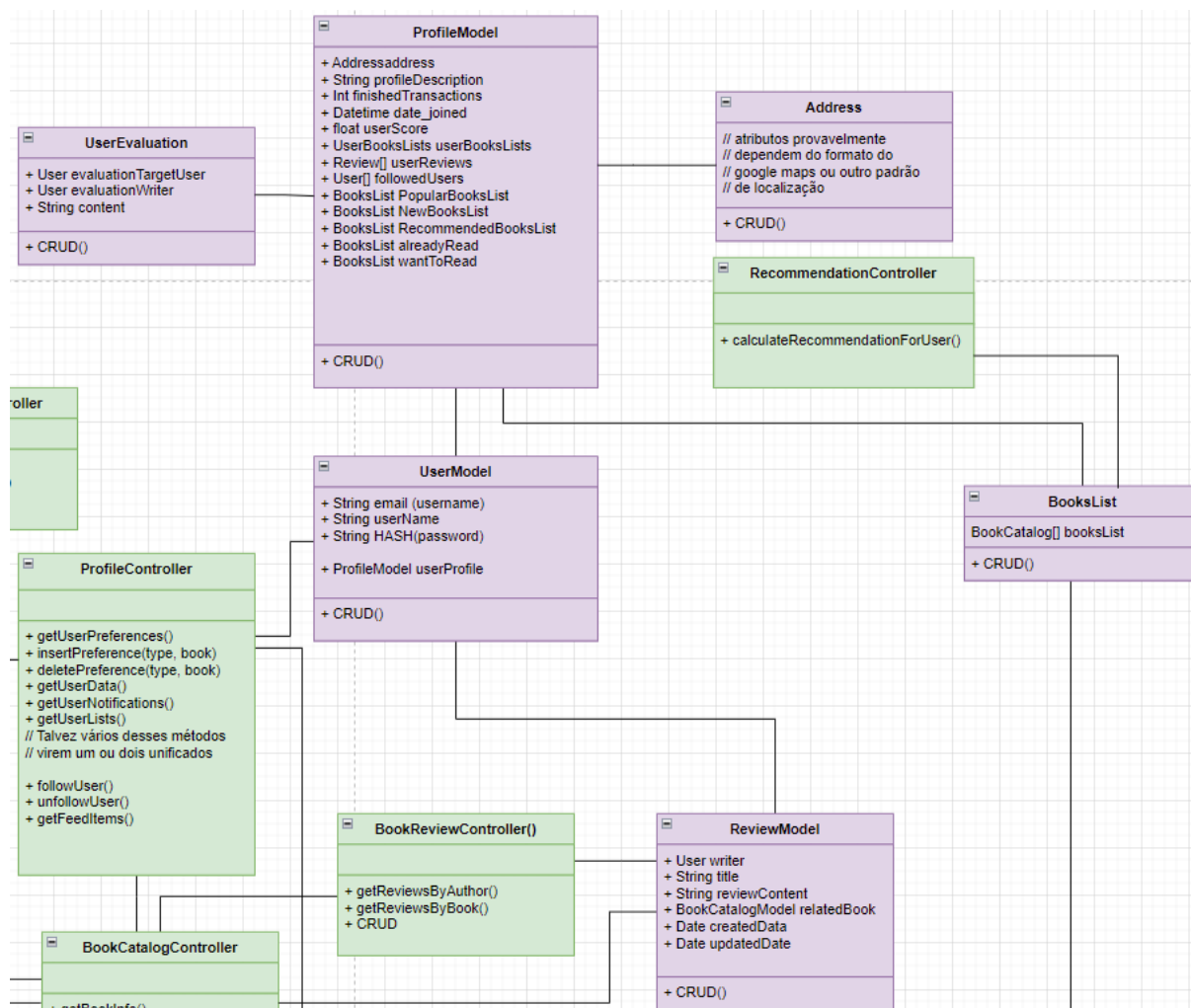
Fonte: autores.

Figura B.6 - Diagrama de classes completo detalhado - Parte 1 Servidor



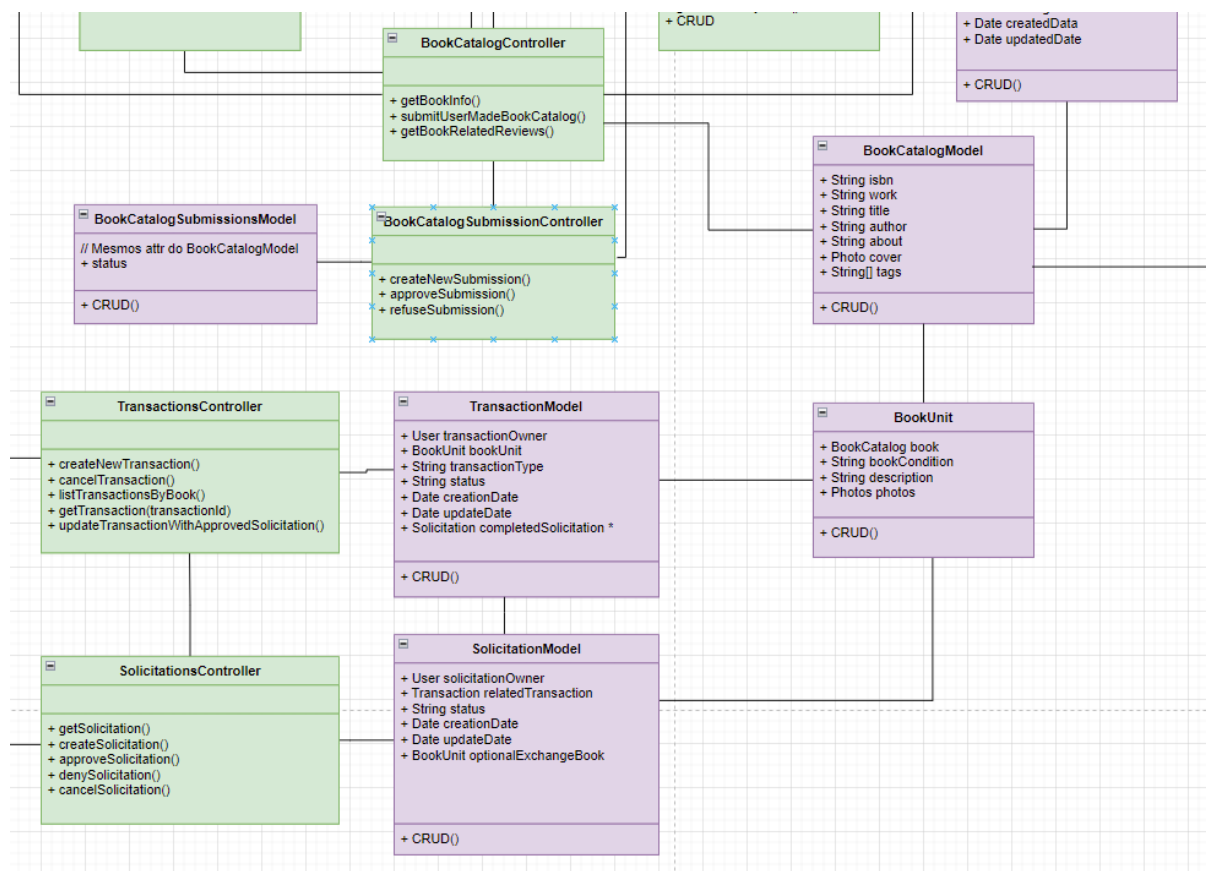
Fonte: autores.

Figura B.7 - Diagrama de classes completo detalhado - Parte 2 Servidor



Fonte: autores.

Figura B.8 - Diagrama de classes completo detalhado - Parte 3 Servidor

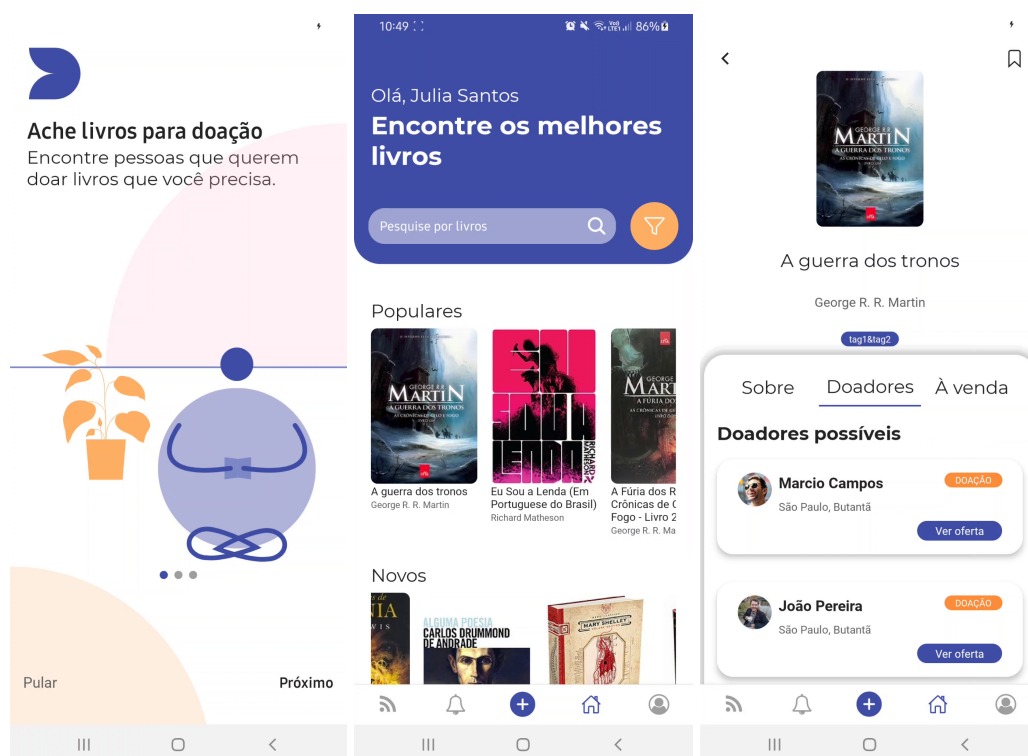


Fonte: autores.

APÊNDICE C - Seleção de telas da aplicação final

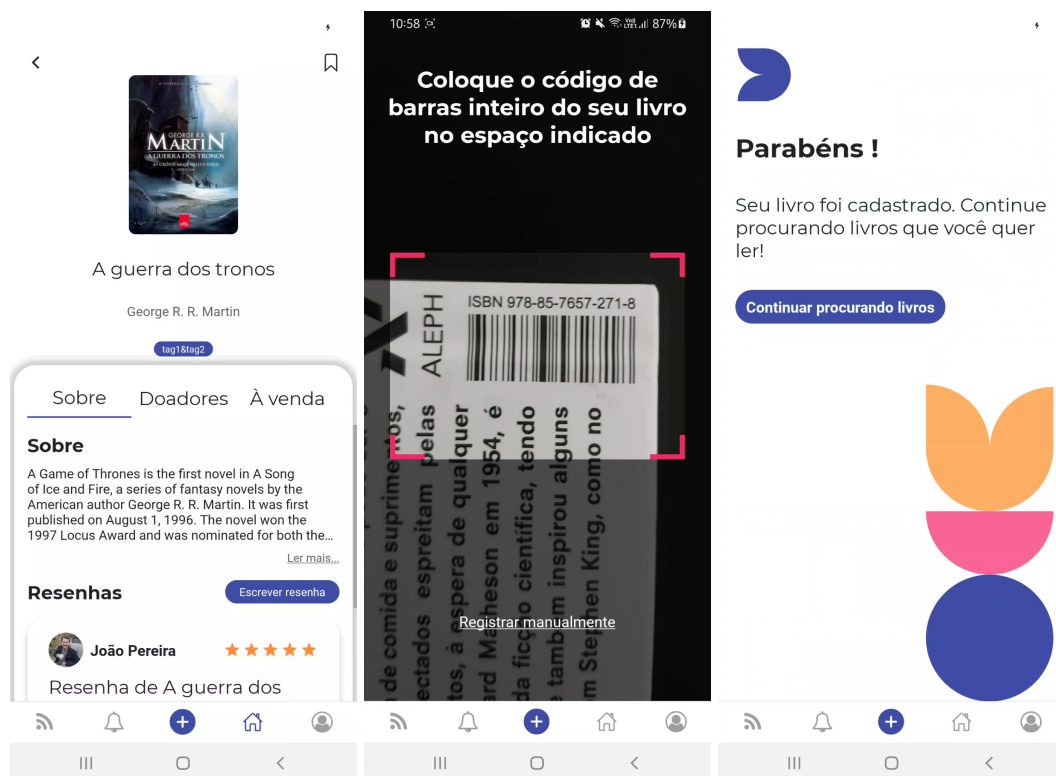
Neste apêndice são ilustradas nas Figuras C.1 à C.10 algumas das telas finais da aplicação a fim de ilustrar o resultado final obtido.

Figuras C.1 à C.3 - Imagens do *onboarding*, menu principal e informações de livro



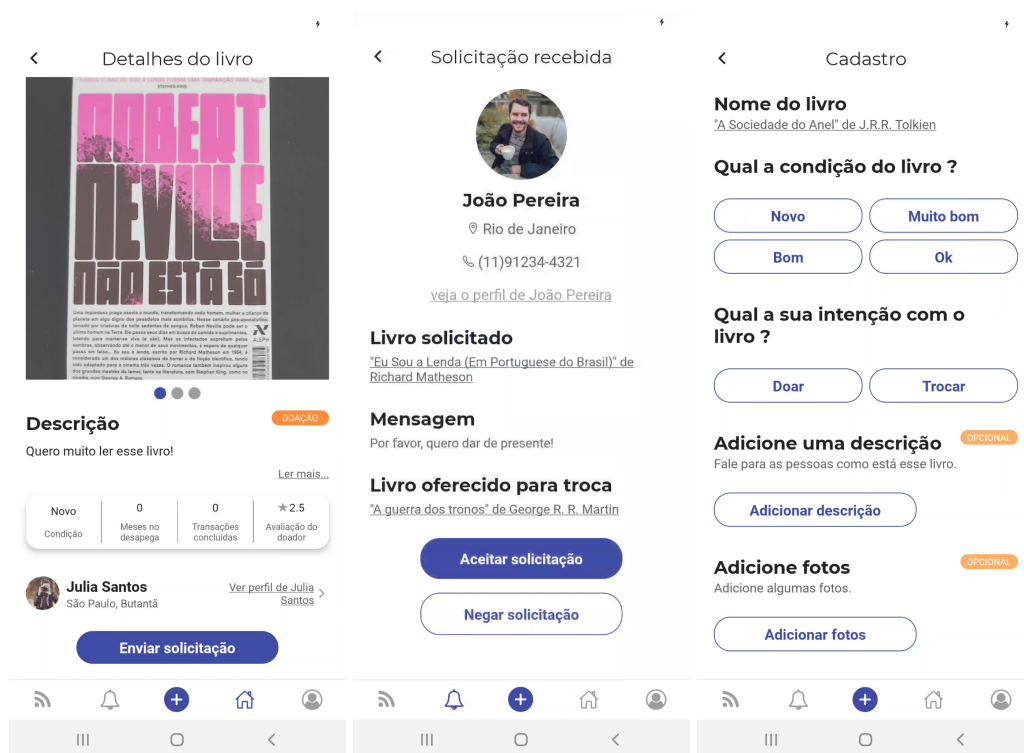
Fonte: autores.

Figuras C.4 à C.6 - Telas de informações de livro, escaneamento e confirmação



Fonte: autores.

Figuras C.7 à C.9 - Telas de oferta, solicitação e cadastro de livro



Fonte: autores.

Figuras C.10 - Tela da interface do administrador

The screenshot shows the Django administration interface for the 'bookcatalogsubmissionmodel' app. The main content area is titled 'Select book catalog submission model to change'. It features a table with the following data:

TITLE	AUTHOR	BOOKCODE	COVER	AÇÕES DE SUBMISSÕES DE LIVROS
<input type="checkbox"/> A Sociedade do Anel	J.R.R. Tolkien	9788595084759	/app/Desapega/media/book/catalog/submissions/bookcatalogsubmission3.jpg	Aprovar Negar

Below the table, it indicates '1 book catalog submission model'. The left sidebar contains a navigation menu with categories like 'AUTHENTICATION AND AUTHORIZATION', 'CORE', and 'Users'. The top navigation bar shows 'Django administration' and 'WELCOME, ADMIN@ADMIN.COM'. A 'FILTER' sidebar on the right allows filtering by status, with options for 'All' and 'Analisar'.

Fonte: autores.