UNIVERSIDADE FEDERAL DE JUIZ DE FORA

TRABALHO FINAL

DCC025 - Orientação a Objetos

João Pedro Martins Cruz 202365552C

Júlia Zoffoli Caçador 202365520B

Robert Gonçalves Vieira de Souza 202365505B

Juiz de Fora, 2024

Sumário

1 Introdução

Nesse projeto, foi desenvolvido um aplicativo chamado "BookSelf", uma ferramenta que possibilita a criação de uma biblioteca virtual personalizada. Através dele, os usuários podem adicionar registros detalhados das suas últimas leituras, bem como acrescentar suas resenhas e notas a fim de registrar sua experiência e conclusões sobre cada obra. Com essa funcionalidade, é possível organizar suas leituras e gerenciar projetos literários de forma prática e eficiente, garantindo que você tenha uma visão clara das obras já lidas, das que estão em andamento e das que pretende explorar. Seu principal objetivo é oferecer uma maneira fácil, intuitiva e personalizável de acompanhar sua jornada literária e, acima de tudo, possibilitar que o sentimento despertado em você durante cada leitura nunca seja esquecido.

Link para o repositório:
https://github.com/martins-joaopedro/DCC025-TrabalhoFinal.git

2 Instruções para Compilação e Execução

Para seguir os passos seguintes, assume-se que o ambiente já possui o JDK 21 e Apache Maven 3.6.3 ou superiores devidamente instalados e configurados.

2.1 Compilação

Para compilar o projeto basta seguir a sequência de passos abaixo.

- Passo 1: Abrir o terminal na pasta raiz do projeto, para verificar que está na pasta raiz execute o comando ls no Linux ou dir no Windows. O resultado deve conter a pasta src e o arquivo pom.xml.
- Passo 2: Executar o comando mvn clean install ou mvn install. Isso irá gerar a pasta target que possui os arquivos e pastas que são resultados do processo de compilação e empacotamento, dentre eles o arquivo bookself-1-jar-with-dependencies.jar que será utilizado para executar o projeto.

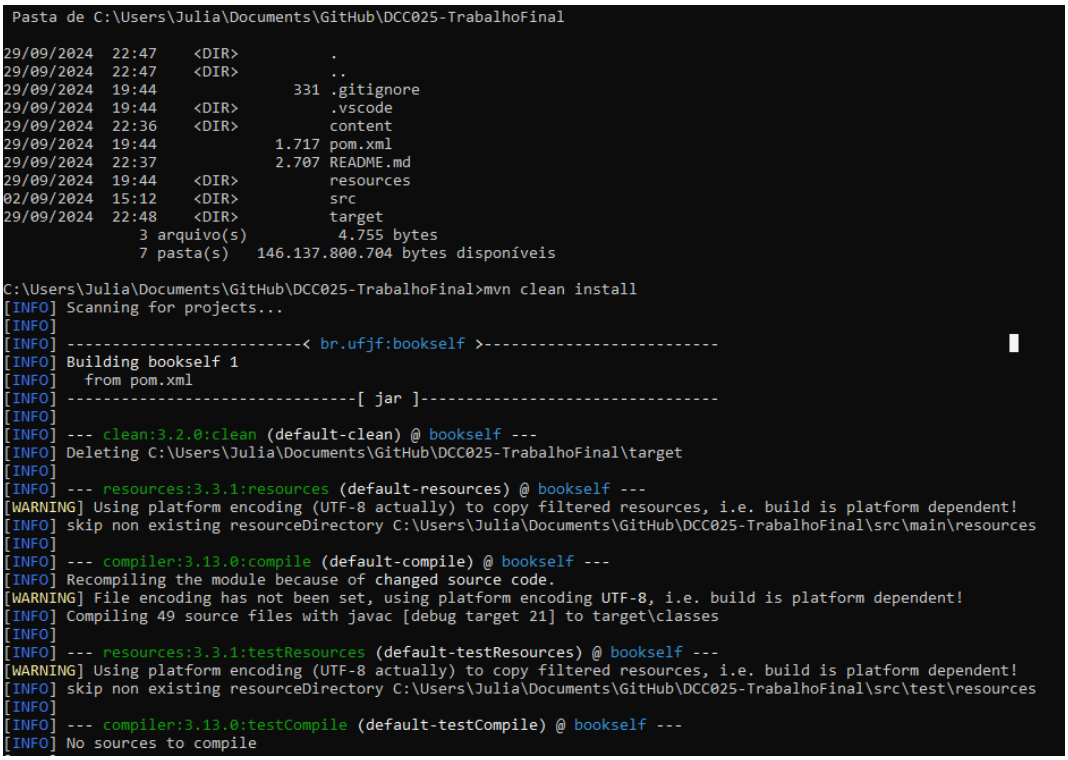

Figura 1: Utilizando o comando *"mvn clean install"*.

2.2 Execução

Para executar o projeto basta abrir a pasta que está com arquivo pom.xml pelo terminal e digitar o comando java -jar target/bookself-1-jar-with-dependencies.jar

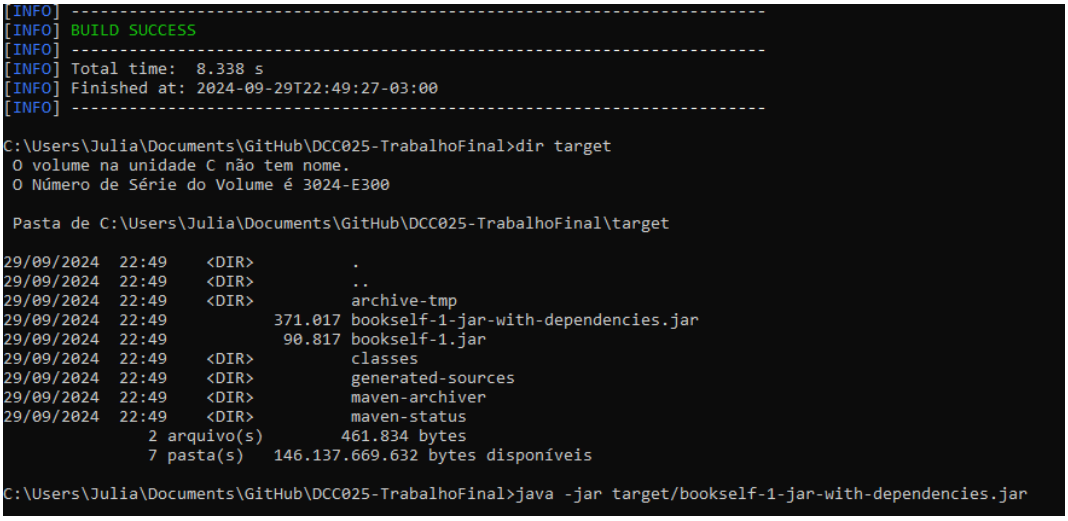

Figura 2: Executando o projeto.

3 Diagrama de Classes

A fim de fornecer uma visão clara e organizada de como os componentes do sistema interagem e se comportam, foi desenvolvido o Diagrama de Classes do projeto para representar visualmente o código do sistema. Através dessa diagrama, foram representadas as classes com seus devidos métodos e atributos e os relacionamentos entre elas. Dessa forma, o grupo conseguiu se organizar a respeito de qual caminho tomar e quais classes deveriam implementar primeiro.

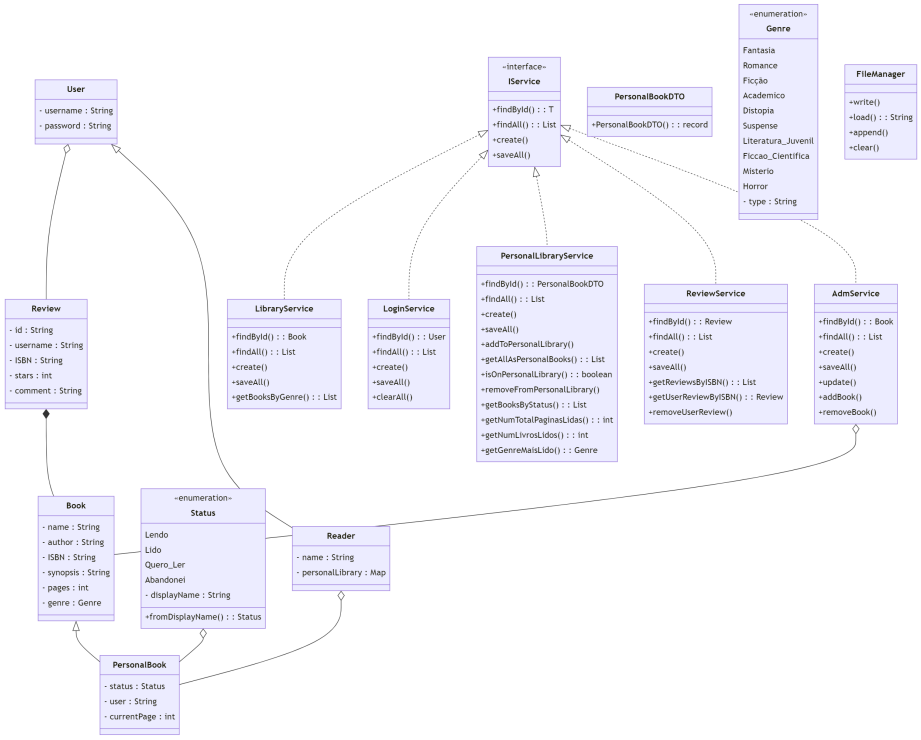

Figura 3: Diagrama de Classes

4 Desenvolvimento

Nessa seção, serão aprofundados as classes implementadas durante o projeto e suas funcionalidades particulares.

4.1 Classes De Excessão

A fim de lidar com erros e situações inesperadas que podem ocorrer durante a execução do programa, foram implementados Tratamentos de Excessão ao decorrer do código. Assim, foi possível garantir que o programa não falhe quando encontra um erro ou uma situação inesperada.

Para tratar excessões específicas, foram criadas as classes CouldNotConvertJsonException, ExceptionsController, LibraryException, ParserExceptions e ReviewsException que extendem de Exception ou Runtime Exception e tem como função tratar essas excessões.

4.2 Enumerations

4.2.1 Classe Genre

A classe Genre é uma enumeration dos gêneros que os livros podem têr. Através do método getType() é possível acessar o gênero do livro escolhido para manipulações futuras.

4.2.2 Classe Status

Assim como a classe Genre, a classe Status também é uma enumeration. Entretanto, o que difere as classes é que, em Status, são enumerados os status em que os livros podem se encontrar, entre eles "Lido", "Lendo", "Pretendo Ler"e "Abandonei". Assim, será possível usar esses dados para melhorar a experiência do usuário e separar os livros do seu acervo pessoal por status. Vale ressaltar que a classe usa o tratamento de excessão para os casos em que nenhum Status com o nome fornecido foi encontrado.

4.3 Classe Book

A classe Book é responsável por representar a base fundamental dessa aplicação, os livros. Ela possui como atributos as características pelas quais um livro pode ser identificado, como seu nome, autor, sinopse, número de paginas e seu gênero. Vale ressaltar que no atributo gênero é utilizada a enumeration Genre, já citada na subseção 4.3.

4.4 Classe PersonalBook

A classe PersonalBook é uma subclasse de Book, ou seja, herda seus métodos e características principais. Entretando, nessa classe são adicionadas informações extras, que possuem a função de levar um dos pilares do projeto que é a personalização da experiência do usuário. Para isso, são utilizados os atributos user, status e currentPage que representam, respectivamente, o usuário, o status atual do livro (utilizando a enumeração Status) e a página em que o leitor parou a leitura da última vez que leu o livro em questão.

4.5 Classe Reader

Essa classe corresponde a quem toda a plataforma desenvolvida é designada, o leitor. Ela possui como atributos seu nome e a lista de livros (PersonalBook) que ele guarda em seu histórico.

4.6 Classe Review

A classe Review, em português, Avaliação, possui como função armazenar as avaliações de cada livro separadas pelo usuário que a fez. Assim, será possível desenvolver uma tela com todas as avaliações dos variados usuários da plataforma e proporcionar uma melhor experiência para o leitor.

4.7 Classe User

É responsável por armazenar o nome de usuário e senha do usuário para, através disso, possibilitar a entrada na plataforma.

4.8 Classe FileManager

A classe FileManager possui a função de manipular o arquivo passado a ela. Por meio de seus métodos write(), load(), append() e clear() é possível carregar, escrever e até mesmo limpar os dados deste arquivo.

4.9 Interface IService

Para essa parte do projeto, foi utilizado o conceito de interface, ou seja, um tipo especial de classe que não tem implementação, definindo apenas um protocolo que será assinado por classes subsequentes. Os métodos findById(), findAll(), create() e saveAll() são definidos por essa interface e utilizam tipos genéricos para fazer suas manipulações. Essa decisão foi tomada pois, como será visto nas seções seguintes, as subclasses que herdarão tais métodos possuem tipos diferentes de parâmetros e retornos, dessa forma, é possível englobar todos eles na mesma assinatura da interface.

4.9.1 Classes Services

As classes AdmService, LibraryService, LoginService, PersonalLibraryService e ReviewService implementam a interface exposta nessa seção. Elas utilizam os métodos já definidos anteriormente na classe FileManager - seção 3.8 - para acessar e manipular o arquivo necessário e, através dos métodos findById() e findAll(), buscar nele os dados que interessam a cada classe.

Na classe LibraryService, por exemplo, o método findById() é utilizado para buscar um livro na lista de livros utilizando seu ISBN.

5 Interface Gráfica e Design

Para o desenvolvimento da Interface Gráfica e do Design, foi utilizada a biblioteca Swing. Algumas das telas da aplicação se encontram nas figuras abaixo.

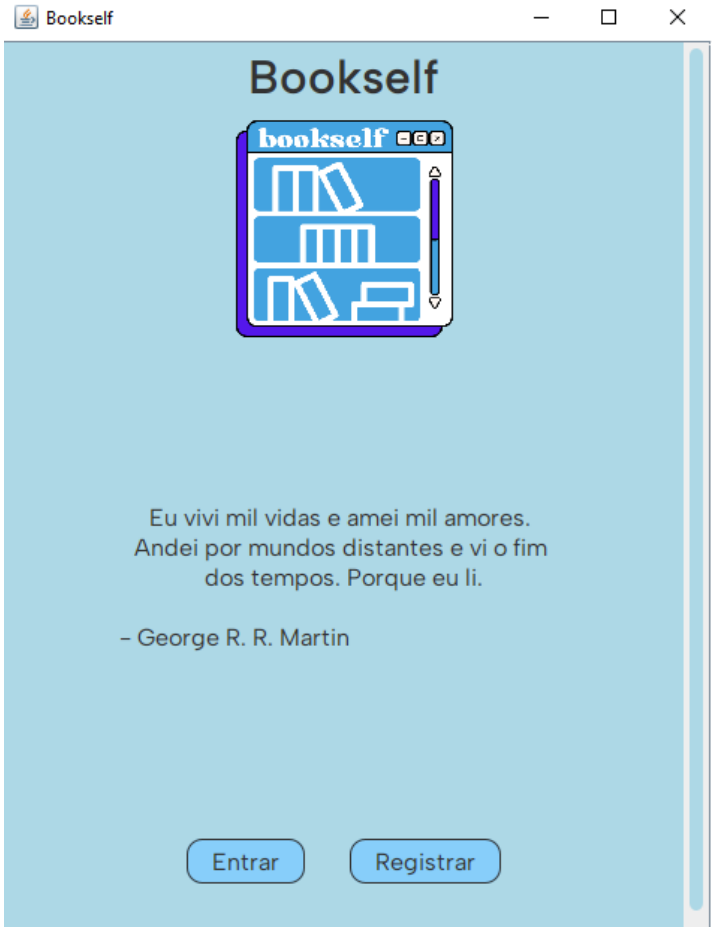

Figura 4: Tela Inicial

Ao iniciar a aplicação, o usuário terá acesso à primeira tela: a tela Inicial. Nela, serão oferecidas duas opções, a de cadastro -para novos usuários - ou a de login para usuários já cadastrados.

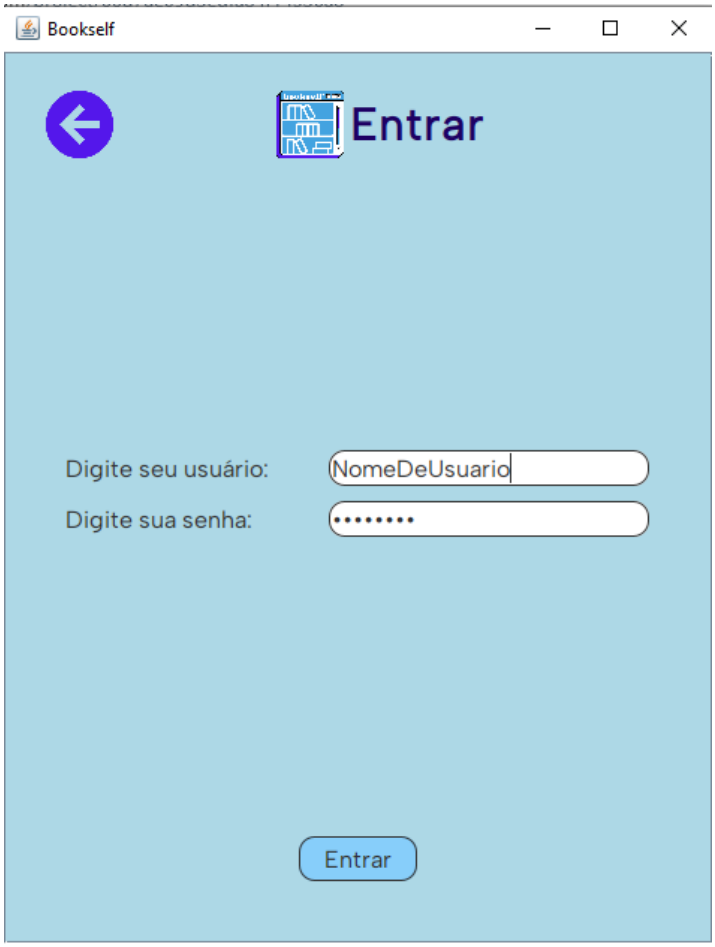

Figura 5: Tela de Login

Nessa tela, os usuários poderão fazer seu login e acessar os registros armazenados em sua conta.

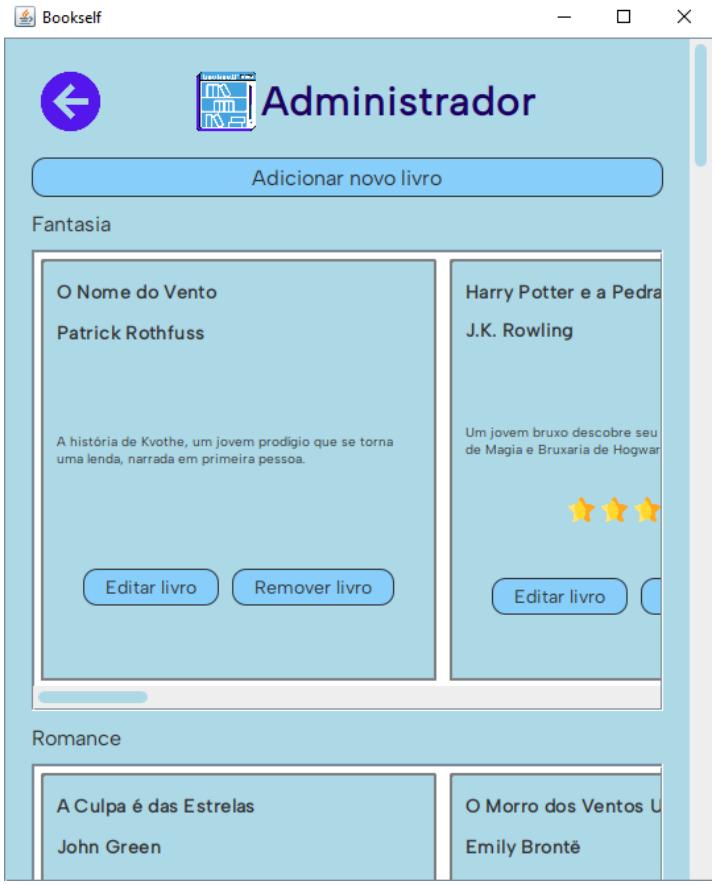

Figura 6: Tela do Administrador

Vale ressaltar que existe um login especial para o Administrador da aplicação. Logando-se com o usuário "admin"e a senha "admin", é permitido o acesso a algumas funcionalidades especiais para a manipulação da biblioteca, como a edição e adição de novas obras.

Figura 7: Tela de Registro

Na tela de registro, novos usuários poderão se registrar, ou seja, criar uma nova conta.

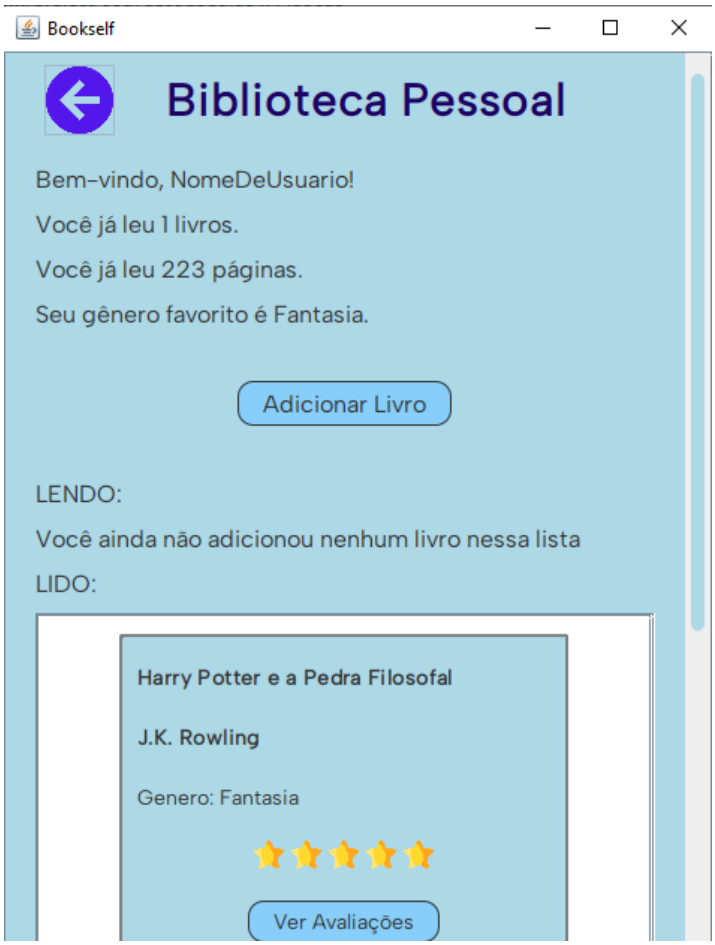

Figura 8: Biblioteca Pessoal

A Biblioteca pessoal é onde o usuário encontrará todo o seu histórico de leitura como o número de livros lidos, o total de páginas e o gênero favorito. Além disso, os livros adicionados a essa biblioteca, encontram-se separados por status -lidos, lendo, abandonei e quero ler.

* Outras telas podem ser encontradas ao executar a aplicação final.

5.1 Classes de Interface Gráfica

Nessa subseção, serão listadas algumas das classes utilizadas na implementação da Interface Gráfica que devem ser ressaltadas.

5.1.1 Classe BasicScreen

A classe BasicScreen é responsável por representar uma tela padrão da interface gráfica do usuário, utilizando um layout de painel que permite a inclusão de componentes como

títulos, botões e outras interfaces interativas. Além disso, a classe também centraliza o estilo visual, isso permite que alterações visuais sejam feitas de forma centralizada e sem a necessidade de modificar cada componente individualmente

5.1.2 Classe BookCard

A classe BookCard é uma classe abstrata que tem como função principal representar os *"cards"* na aplicação. A classe configura a aparência visual e fornece um método para adicionar botões personalizados, permitindo a inclusão de interações adicionais, como visualizar avaliações. Por ser uma classe abstrata, a criação de diferentes tipos de cartões de livro se dá de forma mais prática, garantindo consistência no design e comportamento através do uso de estilos centralizados.

5.1.3 Classes do Pacote Widget

As classes encontradas no pacote Widget herdam de componentes da biblioteca Swing e utilizam o polimorfismo para fazer algumas modificações e personalizar o design. Um exemplo dessa implementação está presente na classe Button visto que ela herda de JButton e faz as alterações necessárias para arredondar as bordas de um botão, por exemplo.

6 Expectativas

O grupo tinha como expectativa implementar de forma clara e coesa todos os conceitos abordados na disciplina de Orientação a Objetos (DCC025), visando estruturar e consolidar os conhecimentos adquiridos ao longo do período. Para isso, surgiu a proposta de desenvolver a plataforma "BookSelf", que tem como objetivo atender as expectativas dos usuários, oferecendo uma ferramenta totalmente personalizável e pessoal. Um dos principais motivadores desse desenvolvimento foi a afinidade dos autores com o universo literário. Como também são usuários, eles trouxeram suas próprias expectativas em relação à aplicação, o que enriqueceu o processo criativo e gerou novas ideias e incentivos.

7 Conclusão

Em conclusão, o desenvolvimento da plataforma "BookSelf"reflete a aplicação prática dos conceitos aprendidos na disciplina de Orientação a Objetos e também se beneficia da rica interação entre autores e suas expectativas. Essa abordagem colaborativa enriquece a experiência do usuário e também fortalece a base teórica e prática adquirida pelo grupo, resultando em uma ferramenta altamente pessoal de acordo com as preferências do usuário. Com isso, espera-se que o BookSelf se torne uma plataforma na qual ideias possam florescer, memórias literárias e as melhores experiências sejam guardadas e

eternizadas. Afinal, como diz George R. R. Martin, "Eu vivi mil vidas e amei mil amores. Andei por mundos distantes e vi o fim dos tempos. Porque eu li.".